

Location-Based Activity Recognition using Relational Markov Networks

Lin Liao and Dieter Fox and Henry Kautz
Department of Computer Science & Engineering
University of Washington
Seattle, WA 98195

Abstract

In this paper we define a general framework for activity recognition by building upon and extending Relational Markov Networks. Using the example of activity recognition from location data, we show that our model can represent a variety of features including temporal information such as time of day, spatial information extracted from geographic databases, and global constraints such as the number of homes or workplaces of a person. We develop an efficient inference and learning technique based on MCMC. Using GPS location data collected by multiple people we show that the technique can accurately label a person's activity locations. Furthermore, we show that it is possible to learn good models from less data by using priors extracted from other people's data.

1 Introduction

Activity recognition and context-aware computing are gaining increasing interest in the AI and ubiquitous computing communities. Most existing systems have been focused on relatively low level activities within small environments or during short periods of time. In this paper, we describe a system that can recognize high level activities (*e.g.*, working, shopping, and dining out) over many weeks. Our system uses data from a wearable GPS location sensor, and is able to identify a user's significant places, and learn to discriminate between the activities performed at these locations — including novel locations. Such activity information can be used in many applications. For example, it could be used to automatically instruct a user's cell phone not to ring when dining at a restaurant, or it could support home rehabilitation of people suffering from traumatic brain injuries [Salazar *et al.*, 2000] by providing automatic activity monitoring. Beyond estimating high-level activity categories, our system can be expanded to incorporate additional sensor information, thereby recognizing fine-grained indoor household tasks, such as those described in [Philipose *et al.*, 2004].

Because behavior patterns can be highly variable, a reliable discrimination between activities must take several sources of evidence into account. Our system considers (1) temporal information such as time of day; (2) spatial information

extracted from geographic databases, including information about the kinds of businesses in various locations; (3) sequential information such as which activity follows which activity; and (4) global constraints such as the number of different homes or workplaces. Additionally, it uses data collected by other users so as to improve the classification of a specific user's activities. All these constraints are *soft*: for example, on some days a person may do something unusual, such as go to a movie in the middle of a workday. Furthermore, the norm for some individuals may be different than our general commonsense prior: for example, some people work two jobs, or shuttle between two different homes. It is necessary to use a rich and flexible language to robustly integrate such a wide variety of both local and global probabilistic constraints.

Our system builds upon previous work on extracting places from traces of users' movements, gathered by GPS or other localization technologies [Ashbrook and Starner, 2003; Hariharan and Toyama, 2004; Liao *et al.*, 2004]. Our work goes beyond theirs in that our system also recognizes the activities associated with the places. Moreover, previous approaches to modeling personal movements and place patterns require a large amount of training data from each user, and cannot be generalized to new places or new users. By contrast, our relational approach requires less individual training data by leveraging data collected by others. In summary, the contributions of this paper are:

1. A general framework for sensor-based activity recognition based on Relational Markov Networks (RMNs) [Taskar *et al.*, 2002], which are both highly expressive and well-suited for discriminative learning;
2. An extension of RMNs to incorporate complex, global features using *aggregations* and *label-specific cliques*;
3. Efficient Markov-chain Monte-Carlo (MCMC) algorithms for inference and learning in extended RMNs, and in particular, an MCMC algorithm to *simultaneously* evaluate a likelihood function and its gradient;
4. Positive experimental results on real data from multiple subjects, including evidence that we can improve accuracy by extracting priors from others' data.

This paper is organized as follows. We will introduce our relational activity model in Section 2. Inference and learning will be discussed in Section 3, followed by experimental evaluations. Conclusions and future work are given in Section 5.

2 The Relational Activity Model

In this section we first discuss RMNs and our extensions. Then we show how to use them for modeling activities.

2.1 Relational Markov Networks

RMNs are extensions of Conditional Random Fields (CRFs), which are undirected graphical models that were developed for labeling sequence data [Lafferty *et al.*, 2001]. CRFs are discriminative models that have been shown to out-perform generative approaches such as HMMs and Markov random fields in areas such as natural language processing [Lafferty *et al.*, 2001] and computer vision [Kumar and Hebert, 2003]. RMNs extend CRFs by providing a relational language for describing clique structures and enforcing parameter sharing at the template level. Thereby RMNs are an extremely flexible and concise framework for defining features that can be used in the activity recognition context.

An RMN consists of three parts: a *schema* $\mathcal{E}$ for the domain, a set of *relational clique templates* $\mathcal{C}$, and corresponding *potentials* Φ . The schema $\mathcal{E}$ specifies the set of *classes* (i.e., entity types) and attributes in each class. An attribute could be a content attribute, a label attribute, or a reference attribute that specifies reference relation among the classes. An *instantiation* $\mathcal{I}$ of a schema specifies the set of entities for each class and the values of all attributes for each entity. In our context an instantiation consists of the sequence of all significant locations visited by a user along with the temporal and spatial attributes.

A *relational clique template* $C \in \mathcal{C}$ is similar to a relational database query (e.g., SQL) in that it selects tuples from an instantiation $\mathcal{I}$; the query result is denoted as $C(\mathcal{I})$. We extend the definition of such templates in two ways. First, we allow a template to select *aggregations* of tuples. For example, we can group tuples and define potentials over counts or other statistics of the groups. Second, we introduce *label-specific cliques*, whose structures depend on values of the labels. For example, our model can construct a clique over all activities labeled as “AtHome.” Because labels are hidden during inference, such cliques potentially involve all the labels. Label-specific cliques can be specified by allowing label attributes to be used in the “Where” clause of an SQL query.

Each clique template C is associated with a potential function $\phi_C(\mathbf{v}_C)$ that maps a tuple (values of variables or aggregations) to a non-negative real number. Using a log-linear combination of feature functions, we get the following representation: $\phi_C(\mathbf{v}_C) = \exp\{\mathbf{w}_C^T \cdot \mathbf{f}_C(\mathbf{v}_C)\}$, where $\mathbf{f}_C()$ defines a feature vector for C and $\mathbf{w}_C^T$ is the transpose of the corresponding weight vector. For instance, a feature could be the number of different homes defined using aggregations.

For a specific instantiation $\mathcal{I}$, an RMN defines a conditional distribution $p(\mathbf{y}|\mathbf{x})$ over labels $\mathbf{y}$ given the observed attributes $\mathbf{x}$. To compute such a conditional distribution, the RMN generates an *unrolled* Markov network, in which the nodes correspond to the content attributes and the label attributes. The cliques of the unrolled network are built by applying each clique template $C \in \mathcal{C}$ to the instantiation, which can result in several cliques per template (see Fig. 1(b) for an example). All cliques that originate from the same tem-

plate must share the same weights $\mathbf{w}_C$. The resulting cliques factorize the conditional distribution as

$$p(\mathbf{y} | \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{C \in \mathcal{C}} \prod_{\mathbf{v}_C \in C} \phi_C(\mathbf{v}_C) \quad (1)$$

$$= \frac{1}{Z(\mathbf{x})} \prod_{C \in \mathcal{C}} \prod_{\mathbf{v}_C \in C} \exp\{\mathbf{w}_C^T \cdot \mathbf{f}_C(\mathbf{v}_C)\} \quad (2)$$

$$= \frac{1}{Z(\mathbf{x})} \exp\{\mathbf{w}^T \cdot \mathbf{f}\}, \quad (3)$$

where the normalizing partition function $Z(\mathbf{x}) = \sum_{\mathbf{y}} \prod_{C \in \mathcal{C}} \prod_{\mathbf{v}_C \in C} \phi_C(\mathbf{v}_C)$. (3) follows by moving the products into the exponent and combining all summations into $\mathbf{w}$ and $\mathbf{f}$.

2.2 Relational Activity Models

We will now describe our relational activity models. Even though we illustrate the concepts using the example of location-based activity recognition, our model is very flexible and can be applied to a variety of activity recognition tasks.

The schema for activity recognition based on temporal and spatial patterns is shown in Fig. 1(a). It includes three classes: *Activity*, *Place*, and *Transition*.

Activity: Activity is the central class in the domain. Its attribute *Label* is the only hidden variable. The set of possible labels in our experiments is {'AtHome', 'AtWork', 'Shopping', 'DiningOut', 'Visiting', 'Others'}. Attribute *Id* serves as the primary key. The class also contains temporal information associated with an activity, such as *TimeOfDay*, *DayOfWeek*, and *Duration*, whose values are discretized when necessary. Finally, *Place* is a reference attribute that points to a Place entity where the activity has been performed.

Place: The class Place includes two boolean attributes: *NearRestaurant* and *NearStore*, which indicate whether there are restaurants or stores nearby.

Transition: Transition captures temporal succession relationship among activities. The reference attributes *From* and *To* refer to a pair of consecutive activities.

Based on the schema, we define the following relational clique templates. Each of them takes into account a number of discriminative features.

1. *Temporal patterns:* Different activities often have different temporal patterns, such as their duration or time of day. Such local patterns are modeled by clique templates that connect each attribute with the activity label.
2. *Geographic evidence:* Information about the types of businesses close to a location can be extremely useful to determine a user's activity. Such information can be extracted from geographic databases, such as Microsoft MapPoint [Hariharan *et al.*, 2005] used in our experiments. Since location information in such databases is not accurate enough, we consider such information by checking whether, for example, a restaurant is within a certain range from the location.
3. *Transition relations:* The first-order transitions between activities can also be informative. For example, staying at home followed by being at work is very common

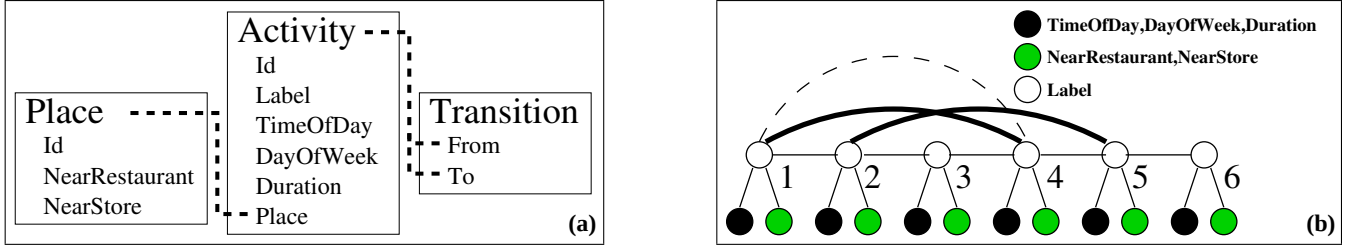


Figure 1: (a) The schema of the relational activity model. Dashed lines indicate reference relations among classes. (b) An example of an unrolled Markov network with six activity locations. Solid straight lines indicate cliques generated by the templates of temporal, geographic, and transition features; bold solid curves represent spatial constraints (activity 1 and 4 are associated with the same place and so are 2 and 5); dashed curves stand for global features, which generate label-specific cliques (e.g., activity 1 and 4 are both labeled 'AtHome').

while dining out immediately followed by another dining out is rare. The SQL query for this clique template is:

```
SELECT a1.Label, a2.Label
FROM Activity a1, Activity a2, Transition t
WHERE t.From=a1.Id AND t.To=a2.Id
```

4. *Spatial constraints*: Activities at the same place are often similar. In other words, the number of different types of activities in a place is often limited. We can express such a constraint using an aggregation function *Count()*:

```
SELECT COUNT(DISTINCT Label)
FROM Activity
GROUP BY Place
```

5. *Global features*: Such features model global, soft constraints on activities of a person. The number of different home locations is an example of global constraints. Such a constraint is modeled by a clique template that selects all places labeled as home and returns how many of them are different:

```
SELECT COUNT(DISTINCT Place)
FROM Activity
WHERE Label='AtHome'
```

Note that the label variable appears in the “Where” clause, so this is an example of label-specific clique. In a different activity recognition context, global features could also model information such as “the number of times a person has lunch per day.”

In the first three templates, the feature functions $f_C()$ are just indicator functions that return binary values. They can also return numbers, such as in the last two templates.

3 Inference and Learning

3.1 Labeling Activities

In our application, the task of inference is to estimate the labels of activities given a sequence of locations visited by a person. To do so, our RMN converts a location sequence into *unrolled* Markov networks, as illustrated in Fig. 1(b). Inference in our relational activity model is complicated by the fact that the structure of the unrolled Markov network can change during inference because of the label-specific cliques. Using standard belief propagation in such networks would require the construction of cliques over all labels, which is obviously inefficient [Taskar *et al.*, 2002]. We overcome this problem

by using MCMC for inference [Gilks *et al.*, 1996]. In a nutshell, whenever the label of an object is changed during sampling, we determine all cliques that could be affected by this change and re-compute their potentials.

We first implemented MCMC using basic Gibbs sampling. Unfortunately, this technique performs poorly in our model because of the strong dependencies among labels. To make MCMC mix faster, we first make an additional spatial constraint that all activities occurring in the same place must have the same label (the relaxation of this constraint will be addressed in future work). This hard constraint allows us to put all activities occurring in the same place into a so-called *block*. We then develop a mixture of two transition kernels that converges to the correct posterior.

The first kernel is a block Gibbs sampler. At each step we update the labels in a block simultaneously by sampling from the full conditional distribution

$$P(y_k | \mathbf{y}_{-k}, \mathbf{x}, \mathbf{w}) \propto \exp\{\mathbf{w}^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}_{-k} \cup y_k)\} \quad (4)$$

where k is the index of the block, y_k is the label of block k , $\mathbf{y}_{-k}$ are the labels for blocks other than k . The second kernel is a Metropolis-Hasting (MH) sampler. To update the label for block k , the MH sampler randomly picks a block j and proposes to exchange label y_k and y_j . The acceptance rate of the proposal follows as

$$a(\mathbf{y}, \mathbf{y}') = \min\left(1, \frac{\exp\{\mathbf{w}^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}')\}}{\exp\{\mathbf{w}^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{y})\}}\right) \quad (5)$$

where $\mathbf{y}$ and $\mathbf{y}'$ are the labels before and after the exchange, respectively.

The numbers of different homes and workplaces are stored in the chains as global variables. This allows us to compute the *global* features *locally* in both kernels: in the Gibbs kernel we increase or decrease the numbers depending on the labels of the given block and in the MH kernel the numbers remain intact. At each time step, we choose the Gibbs sampler with probability γ , and the MH sampler with probability $1 - \gamma$.

3.2 Supervised Learning

We show how to learn *generic* activity models from labeled activity sequences of N different users. Learning a *customized* model for an individual user is a special case when $N = 1$. The parameters to be learned are the feature weights $\mathbf{w}$ that define clique potentials in (3). To avoid overfitting, we perform maximum a posterior (MAP) parameter estimation and impose an independent Gaussian prior with constant

variance for each component of $\mathbf{w}$, i.e., $p(\mathbf{w}) \propto \exp\{-(\mathbf{w} - \mu)^T \cdot (\mathbf{w} - \mu) / 2\sigma^2\}$, where μ is the mean and σ^2 is the variance. We define the MAP objective function as the *negative* log-likelihood of training data from N subjects plus the prior:

$$L(\mathbf{w}) \equiv \sum_{j=1}^N \{-\log P(\mathbf{y}_j | \mathbf{x}_j, \mathbf{w})\} - \log p(\mathbf{w})$$

$$= \sum_{j=1}^N \{-\mathbf{w}^T \cdot \mathbf{f}(\mathbf{x}_j, \mathbf{y}_j) + \log Z(\mathbf{x}_j, \mathbf{w})\} + \frac{(\mathbf{w} - \mu)^T \cdot (\mathbf{w} - \mu)}{2\sigma^2} \quad (6)$$

where j ranges over different users and $\mathbf{y}_j$ are the activity labels for each user. Since (6) is convex, the global minimum can be found using standard optimization algorithms [Taskar *et al.*, 2002]. We apply the quasi-Newton technique to find the optimal weights [Sha and Pereira, 2003]. Each iteration of this technique requires the value and gradient of (6) computed at the weights returned in the previous iteration.

Evaluating the objective function

It can be intractable to compute exact objective values in (6) for all but the simplest cases. This is due to the fact that, for a specific $\mathbf{w}$, it is necessary to evaluate the partition function $Z(\mathbf{x}_j, \mathbf{w})$, which requires summation over all possible label configurations. We approximate the objective value using Monte-Carlo methods [Geyer and Thompson, 1992]. Suppose we already know the value of $L(\tilde{\mathbf{w}})$ for a weight vector $\tilde{\mathbf{w}}$. Then for each subject j , we use our MCMC inference to get M random samples, $\tilde{\mathbf{y}}_j^{(i)} (1 \leq i \leq M)$, from the distribution $P(\mathbf{y} | \mathbf{x}_j, \tilde{\mathbf{w}})$. Then $L(\mathbf{w})$ can be approximated as:

$$L(\mathbf{w}) \approx L(\tilde{\mathbf{w}}) + \sum_{j=1}^N \left\{ \log \left(\frac{1}{M} \sum_{i=1}^M \exp\{(\mathbf{w} - \tilde{\mathbf{w}})^T \cdot \Delta \tilde{\mathbf{f}}_j^{(i)}\} \right) \right. \\ \left. + \frac{(\mathbf{w} - \mu)^T \cdot (\mathbf{w} - \mu) - (\tilde{\mathbf{w}} - \mu)^T \cdot (\tilde{\mathbf{w}} - \mu)}{2\sigma^2} \right\} \quad (7)$$

where $\Delta \tilde{\mathbf{f}}_j^{(i)} = \mathbf{f}(\mathbf{x}_j, \tilde{\mathbf{y}}_j^{(i)}) - \mathbf{f}(\mathbf{x}_j, \mathbf{y}_j)$ is the difference between sampled feature counts using $\tilde{\mathbf{w}}$ and the empirical feature counts in the labeled data.

Eq. (7) can only be used to estimate values of $L(\mathbf{w})$ relative to $L(\tilde{\mathbf{w}})$. Fortunately, such relative values are sufficient for the purpose of optimization. It can be shown that the best approximation in (7) is obtained when $\tilde{\mathbf{w}}$ is close to the optimal $\mathbf{w}$. Therefore, during optimization, our algorithm updates $\tilde{\mathbf{w}}$ with better weight estimates whenever possible.

Evaluating the gradient

The gradient of the objective function, $\nabla L(\mathbf{w})$, equals to the difference between the sampled feature counts and the empirical feature counts, plus a prior term. To generate the sampled feature counts under $\mathbf{w}$, we again run MCMC inference. Suppose we have obtained M random samples, $\mathbf{y}_j^{(i)} (1 \leq i \leq M)$, from the distribution $P(\mathbf{y} | \mathbf{x}_j, \mathbf{w})$. We can compute the gradient as:

$$\nabla L(\mathbf{w}) = \sum_{j=1}^N \{ \mathbf{E}_{\mathbf{w}}[\mathbf{f}(\mathbf{x}_j, \mathbf{y})] - \mathbf{f}(\mathbf{x}_j, \mathbf{y}_j) \} + \frac{\mathbf{w} - \mu}{\sigma^2}$$

$$\approx \sum_{j=1}^N \left\{ \frac{1}{M} \sum_{i=1}^M \Delta \mathbf{f}_j^{(i)} \right\} + \frac{\mathbf{w} - \mu}{\sigma^2} \quad (8)$$

```

input : the weights  $\mathbf{w}$  provided by the optimizer
output:  $L(\mathbf{w})$  and  $\nabla L(\mathbf{w})$ 

//Evaluate the gradient  $\nabla L(\mathbf{w})$ 
foreach subject  $j$  do
    Run MCMC with  $\mathbf{w}$  and get  $M$  samples;
    Get feature count difference  $\Delta \mathbf{f}_j^{(i)} (1 \leq i \leq M)$ ;
end
Compute the gradient  $\nabla L(\mathbf{w})$  using Eq. (8);

//Evaluate the objective value  $L(\mathbf{w})$ 
if First time calling this function then
     $L(\tilde{\mathbf{w}}) = L(\mathbf{w}) = 0$ ;  $\tilde{\mathbf{w}} = \mathbf{w}$ ;
     $\Delta \tilde{\mathbf{f}}_j^{(i)} = \Delta \mathbf{f}_j^{(i)}$  for  $1 \leq j \leq N, 1 \leq i \leq M$ ;
else
    Compute  $L(\mathbf{w})$  using Eq. (7);
    if  $L(\mathbf{w}) < L(\tilde{\mathbf{w}})$  then
         $L(\tilde{\mathbf{w}}) = L(\mathbf{w})$ ;  $\tilde{\mathbf{w}} = \mathbf{w}$ ;
         $\Delta \tilde{\mathbf{f}}_j^{(i)} = \Delta \mathbf{f}_j^{(i)}$  for  $1 \leq j \leq N, 1 \leq i \leq M$ ;
    end
end

```

Algorithm 1: MCMC-based algorithm for simultaneously evaluating objective function and its gradient.

where $\Delta \mathbf{f}_j^{(i)} = \mathbf{f}(\mathbf{x}_j, \mathbf{y}_j^{(i)}) - \mathbf{f}(\mathbf{x}_j, \mathbf{y}_j)$ is the difference between the sampled and the empirical feature counts.

Algorithm

If we compare (7) and (8), we see both require the difference between the sampled and the empirical feature counts. While samples in (7) are based on the weights $\tilde{\mathbf{w}}$, those in (8) are based on $\mathbf{w}$. Therefore, if we always keep the best weight estimate as $\tilde{\mathbf{w}}$, we can reuse the sampled feature counts from gradient estimation, thereby making objective value evaluation very efficient.

Our algorithm *simultaneously* estimates at each iteration the value and the gradient of the negative log-likelihood (6) for given weights $\mathbf{w}$. These estimates are used by the quasi-Newton approach to compute new weights, and then the estimation is repeated. As shown in Alg. 1, both $L(\tilde{\mathbf{w}})$ and $L(\mathbf{w})$ are initialized as 0 and thus all the objective values are evaluated relative to the objective value of initial weights. In later iterations, when we find a better weight estimate that makes $L(\mathbf{w})$ less than $L(\tilde{\mathbf{w}})$, we update $\tilde{\mathbf{w}}$ with the new $\mathbf{w}$ and also keep the new $L(\tilde{\mathbf{w}})$ and $\Delta \tilde{\mathbf{f}}_j^{(i)} (1 \leq j \leq N, 1 \leq i \leq M)$. By doing that, we not only evaluate objective values very efficiently, but are also able to get more accurate approximations as $\tilde{\mathbf{w}}$ approaches closer to the optimal weights.

4 Experiments

To evaluate our location-based activity recognition technique, we collected two sets of location data using wearable GPS units. The first data set (called “single”) contains location traces from a single person over a time period of four months (see Fig. 2). It includes about 400 visits to 50 different places. The second data set (called “multiple”) was collected by five different people, about one week for each. Each person’s data include 25 to 35 visits and 10 to 15 different places. We extracted places / visits from the GPS logs by detecting locations at which a person spends more than 10 minutes [Hariha-

ran and Toyama, 2004]. Each instance corresponds to an activity. We then clustered nearby activity locations into places. For training and evaluation, we let the subjects manually label the types of activities. Then, we trained the models and tested their accuracy. Accuracy was determined by the activities for which the most likely labeling was correct.

Applying learned models to other people

In practice, it is of great value to learn a generic activity model that can be immediately applied to new users without additional training. In this experiment, we used the “multiple” data set and performed leave-one-subject-out cross-validation: we trained using data from four subjects, and tested on the remaining one. The average error rates are indicated by the white bars in Fig. 3(a). By using all the features, the generic models achieved an average error rate of 18%. It can be seen that global features and spatial constraints significantly improve classification. To gage the impact of different habits on the results, we also performed the same evaluation using the “single” data set. In this case, we used one-month data for training and the other three-month data for test, and we repeated the validation process for each month. The results are shown by the gray bars in Fig. 3(a). In this case, the models achieved an error rate of only 7% by using all the features. This experiment shows that it is possible to learn good activity models from groups of people. It also demonstrates that models learned from more “similar” people can achieve higher accuracy. This indicates that models can be improved by grouping people based on their activity patterns.

Table 1 shows the confusion matrix of one experiment on generic models (rightmost white bar in Fig. 3(a)). As can be seen, our approach is able to perfectly label homes and workplaces. The technique performs surprisingly well on the other activities, given that they are extremely difficult to distinguish based on location information alone. The confusion matrix also shows that simply labeling places by the most frequent activity (home) would result in an error rate of 62%.

Truth	Inferred labels					
	Home	Work	Shop	Dining	Visit	Other
Home	57	0	0	0	0	0
Work	0	34	0	0	0	0
Shop	0	0	8	2	0	4
Dining	0	0	3	6	0	4
Visit	0	0	1	0	4	3
Other	0	0	6	1	2	15

Table 1: Confusion matrix of cross-validation on generic models with all features.

To evaluate the impact of number of people available for model learning, we trained our model using data from different numbers of subjects and tested on the remaining one (all features were used). The average error rates of the cross-validation are shown in Fig. 3(b). When trained using only one subject, the system does not perform well (error rate of 35%), mainly because many patterns specific to that person are applied onto others. When more subjects are used for training, the patterns being learned are more generic and the models achieve significantly higher accuracy.

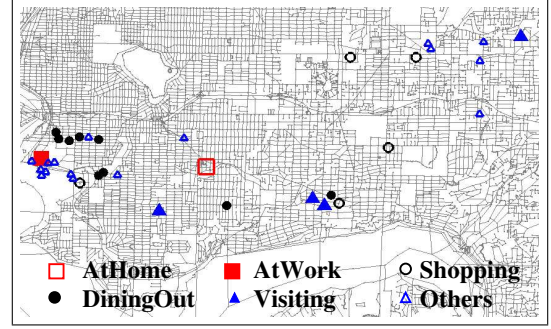


Figure 2: Part of the locations contained in the “single” data set, collected over a period of four months (x -axis is 8 miles long).

Improved learning through priors extracted from others

When estimating the weights of RMNs, a prior is imposed in order to avoid overfitting. Without additional information, a zero mean Gaussian is typically used as the prior [Taskar *et al.*, 2002]. [Peng and McCallum, 2004] demonstrated that better accuracy can be achieved if feature-dependent variances are used. Our experiment shows that performance can also be improved by estimating the prior means of the weights (μ in Eq. (6)) using data collected from other people.

In this experiment, we compared the models of a specific person trained using a zero-mean prior with the models trained using an estimated prior. In the latter case, we first learned the feature weights from other people and used those as the mean of the Gaussian prior. We evaluated the performance for different amounts of training data available for the test person. The results are shown in Fig. 3(c), in which the error rates are counted only on the *novel* places, *i.e.*, places that were not visited in the training data and thus often very irregular. We can see that using data from others to generate a prior boosts the accuracy significantly, especially when only small amounts of training data are available.

The Bayesian prior allows the model to smoothly shift from generic to customized: On one end, when no data from the given subject are available, the approach returns the generic (prior) model; on the other end, as more labeled data become available, the model adjusts more and more to the specific patterns of the user.

Additional experiments

For comparison, we also built basic HMMs in which the hidden states are the labels and all the observations are independent given the states. Parameter estimation in HMMs with labeled data is done via frequency counting. The most likely labels can be found using the Viterbi algorithm. In the one-month-training cross-validation on the “single” data set, the HMM produced an average error rate of 21.1% by using the temporal, geographic, and transition features.¹ Because of the advantages of discriminative learning, even using the same features, RMNs performed better than HMMs and reduced the relative error rate by about 10%.

In a separate set of experiments, we tested the performance of our MCMC sampler. By visualizing the standard Gelman-Rubin statistics [Gilks *et al.*, 1996] generated from parallel

¹Spatial constraints and global features do not satisfy the first-order Markov assumption and thus are difficult to model as HMMs.

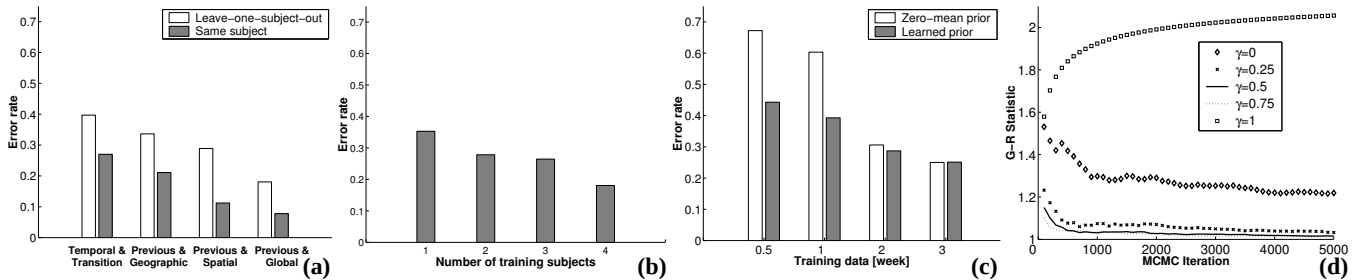


Figure 3: (a) Error rates of models using different features: White bars represent errors of models learned from data collected by other people, and gray bars are for models learned and tested using data collected by the same person (“Previous” means all previous features are also used). (b) Error rates of generic models with respect to different numbers of training subjects. (c) Error rates of zero-mean prior vs. priors learned from other people. (d) Convergence comparison of MCMC using different γ ’s: G-R statistics approaching 1 indicates good convergence ($\gamma = 0$ corresponds to using only MH sampler and $\gamma = 1$ corresponds to the block Gibbs sampler).

chains, we observed that by combining the Gibbs and the MH kernels, MCMC converged much faster than using only one of them (see Fig. 3(d)). All results reported here were achieved with a mixing parameter $\gamma = 0.5$ (see Section 3).

5 Conclusions and Future Work

In this paper, we presented a discriminative relational approach for activity recognition based on the framework of RMNs, which are well-suited to model constraints for activity recognition. We showed how to perform efficient inference and learning using MCMC with a mixture of kernels.

Using our relational approach, we developed and tested a specific model for location-based activity recognition. The results are very promising: the system is able to learn models that can accurately label human activities solely based on GPS data. We demonstrated that spatial and global features are very important to achieve good recognition rates. We also showed how to obtain good priors using data from other people so as to learn an improved model for a specific person that requires less labeled data.

We plan to extend our model in a number of ways. First, by collecting data from more subjects, we can learn a set of generic models by clustering the subjects based on their similarities; then we can use a mixture of these models to better recognize activities of a new person. Second, we will relax the hard spatial constraint of one activity per location and thus recognize different activities performed at the same place. Third, we will integrate information from other wearable sensors (e.g., microphones or accelerometers) into our general framework, thereby enabling much finer-grained activity recognition. We will also apply the model to the problem of estimating a person’s indoor activities from RFID sensor data. By incorporating constraints such as “a person typically has lunch once per day,” we expect strong improvements over the results reported in [Philipose et al., 2004].

Acknowledgments

We thank Anthony LaMarca, Don J. Patterson, and Xiaolin Sun for collecting data used in our experiments. We would also like to thank John Krumm for suggesting the use of geographic databases such as Microsoft MapPoint. This research is based on the work that has been supported by DARPA’s CALO project and NSF.

References

- [Ashbrook and Starner, 2003] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. In *Personal and Ubiquitous Computing*, 2003.
- [Geyer and Thompson, 1992] C. J. Geyer and E. A. Thompson. Constrained Monte Carlo Maximum Likelihood for dependent data. *Journal of Royal Statistical Society*, 1992.
- [Gilks et al., 1996] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall/CRC, 1996.
- [Hariharan and Toyama, 2004] R. Hariharan and K. Toyama. Project Lachesis: parsing and modeling location histories. In *Geographic Information Science*, 2004.
- [Hariharan et al., 2005] R. Hariharan, J. Krumm, and E. Horvitz. Web-enhanced GPS. In *International Workshop on Location- and Context-Awareness*, 2005.
- [Kumar and Hebert, 2003] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Proc. of the International Conference on Computer Vision (ICCV)*, 2003.
- [Lafferty et al., 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of the International Conference on Machine Learning (ICML)*, 2001.
- [Liao et al., 2004] L. Liao, D. Fox, and H. Kautz. Learning and inferring transportation routines. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2004.
- [Peng and McCallum, 2004] F. Peng and A. McCallum. Accurate information extraction from research papers using conditional random fields. In *HLT-NAACL*, 2004.
- [Philipose et al., 2004] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, D. Hhnel, D. Fox, and H. Kautz. Inferring ADLs from interactions with objects. *IEEE Pervasive Computing*, 3(4), 2004.
- [Salazar et al., 2000] A. Salazar, D. Warden, K. Schwab, J. Specator, S. Braverman, J. Walter, R. Cole, M. Rosner, E. Martin, and R. Ellenbogen. Cognitive rehabilitation for traumatic brain injury. *Journal of American Medical Association*, 283(23), 2000.
- [Sha and Pereira, 2003] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of Human Language Technology-NAACL*, 2003.
- [Taskar et al., 2002] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

Concurrent Hierarchical Reinforcement Learning

Bhaskara Marthi, Stuart Russell, David Latham

Computer Science Division

University of California

Berkeley, CA 94720

{bhaskara,russell,latham}@cs.berkeley.edu

Carlos Guestrin

School of Computer Science

Carnegie Mellon University

Pittsburgh, PA 15213

guestrin@cs.cmu.edu

Abstract

We consider applying hierarchical reinforcement learning techniques to problems in which an agent has several effectors to control simultaneously. We argue that the kind of prior knowledge one typically has about such problems is best expressed using a *multithreaded partial program*, and present concurrent ALisp, a language for specifying such partial programs. We describe algorithms for learning and acting with concurrent ALisp that can be efficient even when there are exponentially many joint choices at each decision point. Finally, we show results of applying these methods to a complex computer game domain.

1 Introduction

Hierarchical reinforcement learning (HRL) is an emerging subdiscipline in which reinforcement learning methods are augmented with prior knowledge about the high-level structure of behaviour. Various formalisms for expressing this prior knowledge exist, including HAMs [Parr and Russell, 1997], MAXQ [Dietterich, 2000], options [Precup and Sutton, 1998], and ALisp [Andre and Russell, 2002]. The idea is that the prior knowledge should enormously accelerate the search for good policies. By representing the hierarchical structure of behaviour, these methods may also expose additive structure in the value function [Dietterich, 2000].

All of these HRL formalisms can be viewed as expressing constraints on the behavior of a single agent with a single thread of control; for example, ALisp extends the Lisp language with nondeterministic constructs to create a single-threaded partial programming language (see Section 2). Experience with programming languages in robotics suggests, however, that the behaviour of complex physical agents in real domains is best described in terms of *concurrent* activities. Such agents often have several *effectors*, so that the total action space available to the agent is the Cartesian product of the action spaces for each effector.

Consider, for example, the *Resource domain* shown in Figure 1. This is a subproblem of the much larger Stratagus domain (see stratagus.sourceforge.net). Each peasant in the Resource domain can be viewed as an effector, part of a multi-body agent. Controlling all the peasants with a single control

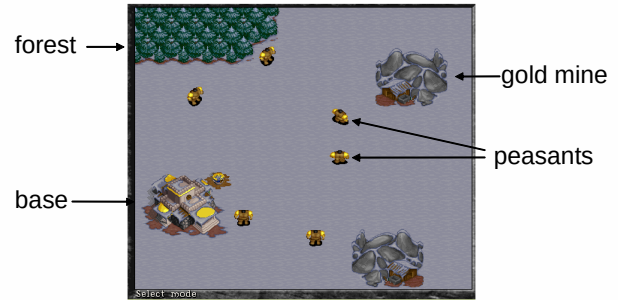


Figure 1: A resource-gathering subgame within Stratagus. Peasants can move one step in the N, S, E, W directions, or remain stationary (all transitions are deterministic). They can pick up wood or gold if they are at a forest or goldmine and drop these resources at the base. A reward of 1 is received whenever a unit of a resource is brought back to the base. A cost of 1 is paid per timestep, and an additional cost of 5 is paid when peasants collide. The game ends when the player’s reserves of gold and wood reach a predefined threshold.

thread is difficult. First, different peasants may be involved in different activities and may be at different stages within those activities; thus, the thread has to do complex bookkeeping that essentially implements multiple control stacks. Second, with N peasants, there are 8^N possible joint actions, so action selection is combinatorially nontrivial even when exact value functions are given. Finally, there are interactions among all the peasants’ actions—they may collide with each other and may compete for access to the mines and forests.

To handle all of these issues, we introduce *concurrent ALisp*, a language for HRL in multi-effector problems. The language extends ALisp to allow *multithreaded partial programs*, where each thread corresponds to a “task”. Threads can be created and destroyed over time as new tasks are initiated and terminated. At each point in time, each effector is controlled by some thread, but this mapping can change over time. Prior knowledge about coordination between threads can be included in the partial program, but even if it is not, the threads will coordinate their choices at runtime to maximize the global utility function.

We begin by briefly describing ALisp (Section 2) and its theoretical foundations in semi-Markov decision processes (SMDPs). Section 3 defines the syntax of concurrent ALisp and its semantics—i.e., how the combination of a concur-

rent ALisp program and a physical environment defines an SMDP. We obtain a result analogous to that for ALisp: the optimal solution to the SMDP is the optimal policy for the original environment that is consistent with the partial program. Section 4 describes the methods used to handle the very large SMDPs that we face: we use linear function approximators combined with relational feature templates [Guestrin *et al.*, 2003], and for action selection, we use coordination graphs [Guestrin *et al.*, 2002]. We also define a suitable SMDP Q-learning method. In Section 5, we show experimentally that a suitable concurrent ALisp program allows the agent to control a large number of peasants in the resource-gathering domain; we also describe a much larger Stratagus subgame in which very complex concurrent, hierarchical activities are learned effectively. Finally, in Section 6, we show how the additive reward decomposition results of Russell and Zimdars [2003] may be used to recover the three-part Q-decomposition results for single-threaded ALisp within the more complex concurrent setting.

2 Background

ALisp [Andre and Russell, 2002] is a language for writing partial programs. It is a superset of common Lisp, and ALisp programs may therefore use all the standard Lisp constructs including loops, arrays, hash tables, etc. ALisp also adds the following new operations :

- **(choose choices)** picks one of the forms from the list choices and executes it. If choices is empty, it does nothing.
- **(action a)** performs action *a* in the environment.
- **(call f args)** calls subroutine *f* with argument list args.
- **(get-state)** returns the current environment state.

Figure 2 is a simple ALisp program for the Resource domain of Figure 1 in the single-peasant case. The agent executes this program as it acts in the environment. Let $\omega = (s, \theta)$ be the *joint state*, which consists of the *environment state* *s* and the *machine state* θ , which in turn consists of the program counter, call stack, and global memory of the partial program. When the partial program reaches a choice state, i.e., a joint state in which the program counter is at a **choose** statement, the agent must pick one of the choices to execute, and the learning task is to find the optimal choice at each choice point as a function of ω . More formally, an ALisp partial program coupled with an environment results in a semi-Markov decision process (a Markov decision process where actions take a random amount of time) over the joint choice states, and finding the optimal policy in this SMDP is equivalent to finding the optimal completion of the partial program in the original MDP [Andre, 2003].

3 Concurrent ALisp

Now consider the Resource domain with more than one peasant and suppose the prior knowledge we want to incorporate is that each individual peasant behaves as in the single-peasant case, i.e., it picks a resource and a location, navigates to that location, gets the resource, returns to the base and drops it off, and repeats the process.

```
(defun single-peasant-top ()
  (loop do
    (choose
      '((call get-gold) (call get-wood)))))

(defun get-wood ()
  (call nav (choose *forests*))
  (action 'get-wood)
  (call nav *home-base-loc*)
  (action 'dropoff))

(defun get-gold ()
  (call nav (choose *goldmines*))
  (action 'get-gold)
  (call nav *home-base-loc*)
  (action 'dropoff))

(defun nav (l)
  (loop until (at-pos l) do
    (action (choose '(N S E W Rest)))))
```

Figure 2: ALisp program for Resource domain with a single peasant.

```
(defun multi-peasant-top ()
  (loop do
    (loop until (my-effectors) do
      (choose '()))
    (setf p (first (my-effectors)))
    (choose
      (spawn p #'get-gold () (list p))
      (spawn p #'get-wood () (list p)))))
```

Figure 3: New top level of a concurrent ALisp program for Resource domain with any number of peasants. The rest of the program is identical to Figure 2.

To incorporate such prior knowledge, we have developed a concurrent extension of ALisp that allows multithreaded partial programs. For the multiple-peasant Resource domain, the concurrent ALisp partial program is identical to the one in Figure 2, except that the top-level function is replaced by the version in Figure 3. The new top level waits for an unassigned peasant and then chooses whether it should get gold or wood and spawns off a thread for the chosen task. The peasant will complete its assigned task, then its thread will die and it will be sent back to the top level for reassignment. The overall program is not much longer than the earlier ALisp program and doesn't mention coordination between peasants. Nevertheless, when executing this partial program, the peasants will automatically coordinate their decisions and the Q-function will refer to joint choices of all the peasants. So, for example, they will learn not to collide with each other while navigating. The joint decisions will be made efficiently despite the exponentially large number of joint choices. We now describe how all this happens.

3.1 Syntax

Concurrent ALisp includes all of standard Lisp. The **choose** and **call** statements have the same syntax as in ALisp.

Threads and effectors are referred to using unique IDs. At any point during execution, there is a set of threads and each thread is assigned some (possibly empty) subset of the effectors. So the **action** statement now looks like (**action** $e_1 a_1 \dots e_n a_n$) which means that each effector e_i must do a_i . In the special case of a thread with exactly one effector assigned to it, the e_i can be omitted. Thus, a legal program for single-threaded ALisp is also legal for concurrent ALisp.

The following operations deal with threads and effectors.

- (**spawn** thread-id fn args effector-list) creates a new thread with the given id which starts by calling fn with arguments args, with the given effectors.
- (**reassign** effector-list thread-id) reassigns the given effectors (which must be currently assigned to the calling thread) to thread-id.
- (**my-effectors**) returns the set of effectors assigned to the calling thread.

Interthread communication is done through condition variables, using the following operations :

- (**wait** cv-name) waits on the given condition variable (which is created if it doesn't exist).
- (**notify** cv-name) wakes up all threads waiting on this condition variable.

Every concurrent ALisp program should contain a designated top level function where execution will begin. In some domains, the set of effectors changes over time. For example, in Stratagus, existing units may be destroyed and new ones trained. A program for such a domain should include a function **assign-effectors** which will be called at the beginning of each timestep to assign new effectors to threads.

3.2 Semantics

We now describe the state space, initial state, and transition function that obtain when running a concurrent ALisp program in an environment. These will be used to construct an SMDP, analogous to the ALisp case; actions in the SMDP will correspond to joint choices in the partial program.

The joint state space is $\Omega = \{\omega = (s, \theta)\}$ where s is an environment state and θ is a machine state. The machine state consists of a global memory state μ , a list Ψ of threads, and for each $\psi \in \Psi$, a unique identifier ι , the set $\mathcal{E}$ of effectors assigned to it, its program counter ρ , and its call stack σ . We say a thread is *holding* if it is at a **choose**, **action**, or **wait** statement. A thread is called a *choice thread* if it is at a **choose** statement. Note that statements can be nested, so for example in the last line of Figure 2, a thread could either be about to execute the **choose**, in which case it is a choice thread, or it could have finished making the choice and be about to execute the **action**, in which case it is not.

In the initial machine state, global memory is empty and there is a single thread starting at the top level function with all the effectors assigned to it.

There are three cases for the transition distribution. In the first case, known as an *internal state*, at least one thread is not holding. We need to pick a nonholding thread to execute next, and assume there is an external *scheduler* that makes this choice. For example, our current implementation is built

on Allegro Lisp and uses its scheduler. However, our semantics will be seen below to be independent of the choice of scheduler, so long as the scheduler is fair (i.e., any nonholding thread will eventually be chosen). Let ψ be the thread chosen by the scheduler. If the next statement in ψ does not use any concurrent ALisp operations, its effect is the same as in standard Lisp. The **call** statement does a function call and updates the stack; the **spawn**, **reassign**, **my-effectors**, **wait**, and **notify** operations work as described in Section 3.1. After executing this statement, we increment ψ 's program counter and, if we have reached the end of a function call, pop the call stack repeatedly until this is not the case. If the stack becomes empty, the initial function of the thread has terminated, ψ is removed from Ψ , and its effectors are reassigned to the thread that spawned it.

For example, suppose the partial program in Figure 3 is being run with three peasants, with corresponding threads ψ_1 , ψ_2 , and ψ_3 , and let ψ_0 be the initial thread. Consider a situation where ψ_0 is at the dummy **choose** statement, ψ_1 is at the third line of **get-wood**, ψ_2 is at the **call** in the first line of **get-gold**, and ψ_3 is at the last line of **get-gold**. This is an internal state, and the set of nonholding threads is $\{\psi_1, \psi_2\}$. Suppose the scheduler picks ψ_1 . The **call** statement will then be executed, the stack appropriately updated, and ψ_1 will now be at the top of the **nav** subroutine.

The second case, known as a *choice state*, is when all threads are holding, and there exists a thread which has effectors assigned to it and is not at an **action** statement. The agent must then pick a joint choice for all the choice threads¹ given the environment and machine state. The program counters of the choice threads are then updated based on this joint choice. The choices made by the agent in this case can be viewed as a *completion* of the partial program. Formally, a completion is a mapping from choice states to joint choices. In the example, suppose ψ_0 is at the dummy **choose** statement as before, ψ_1 and ψ_3 are at the **choose** in **nav**, and ψ_2 is at the **dropoff** action in **get-wood**. This is a choice state with choice threads $\{\psi_0, \psi_1, \psi_3\}$. Suppose the agent has learnt a completion of the partial program which makes it choose $\{(), N, \text{Rest}\}$ here. The program counter of ψ_0 will then move to the top of the **loop**, and the program counters of ψ_1 and ψ_3 will move to the **action** statement in **nav**.

The third case, known as an *action state*, is when all threads are holding, and every thread that has effectors assigned to it is at an **action** statement. Thus, a full joint action is determined. This action is performed in the environment and the action threads are stepped forward. If any effectors have been added in the new environment state, the **assign-new-effectors** function is called to decide which threads to assign them to. Continuing where we left off in the example, suppose ψ_0 executes the **my-effectors** statement and gets back to the **choose** statement. We are now at an action state and the joint action $\{N, \text{dropoff}, \text{Rest}\}$ will be done in the environment. ψ_1 and ψ_3 will return to the **until** in the loop and ψ_2 will die, releasing its effector to **top**.

¹If there are no choice threads, the program has deadlocked. We leave it up to the programmer to write programs that avoid deadlock.

Let Γ be a partial program and S a scheduler, and consider the following random process. Given a choice state ω and joint choice u , repeatedly step the partial program forward as described above until reaching another choice state ω' , and let N be the number of joint actions that happen while doing this. ω' and N are random variables because the environment is stochastic. Let $P_{\Gamma,S}(\omega', N | \omega, u)$ be their joint distribution given ω and u .

We say Γ is *good* if $P_{\Gamma,S}$ is the same for any fair S .² In this case, we can define a corresponding SMDP over Ω_c , the set of choice states. The set of “actions” possible in the SMDP at ω is the set of joint choices at ω . The transition distribution is $P_{\Gamma,S}$ for any fair S . The reward function $R(\omega, u)$ is the expected discounted reward received until the next choice state. The next theorem shows that acting in this SMDP is equivalent to following the partial program in the original MDP.

Theorem 1 *Given a good partial program Γ , there is a bijective correspondence between completions of Γ and policies for the SMDP which preserves the value function. In particular, the optimal completion of Γ corresponds to the optimal policy for the SMDP.*

Here are some of the design decisions implicit in our semantics. First, threads correspond to tasks and may have multiple effectors assigned to them. This helps in incorporating prior knowledge about tasks that require multiple effectors to work together. It also allows for “coordination threads” that don’t directly control any effectors. Second, threads wait for each other at action statements, and all effectors act simultaneously. This prevents the joint behaviour from depending on the speed of execution of the different threads. Third, choices are made jointly, rather than sequentially with each thread’s choice depending on the threads that chose before it. This is based on the intuition that a Q-function for such a joint choice $Q(u_1, \dots, u_n)$ will be easier to represent and learn than a set of Q-functions $Q_1(u_1), Q_2(u_2|u_1), \dots, Q_n(u_n|u_1, \dots, u_{n-1})$. However, making joint choices presents computational difficulties, and we will address these in the following section.

4 Implementation

In this section, we describe our function approximation architecture and algorithms for making joint choices and learning the Q-function. The use of linear function approximation turns out to be crucial to the efficiency of the algorithms.

4.1 Linear function approximation

We represent the SMDP policy for a partial program implicitly, using a Q-function, where $Q(\omega, u)$ represents the total expected discounted reward of taking joint choice u in ω and acting optimally thereafter. We use the linear approximation $\hat{Q}(\omega, u; \vec{w}) = \sum_{k=1}^K w_k f_k(\omega, u)$, where each f_k is a *feature* that maps (ω, u) pairs to real numbers. In the resource domain, we might have a feature $f_{\text{gold}}(\omega, u)$ that returns the amount of gold reserves in state ω . We might also have a set of features $f_{\text{coll},i,j}(\omega, u)$ which returns 1 if the navigation

²This is analogous to the requirement that a standard multi-threaded program contain no race conditions.

choices in u will result in a collision between peasants i and j and 0 otherwise.

Now, with N peasants, there will be $O(N^2)$ collision features, each with a corresponding weight to learn. Intuitively, a collision between peasants i and j should have the same effect for any i and j . Guestrin et al. [2003] proposed using a *relational value-function approximation*, in which the weights for all these features are all tied together to have a single value w_{coll} . This is mathematically equivalent to having a single “feature template” which is the sum of the individual collision features, but keeping the features separate exposes more structure in the Q-function, which will be critical to efficient execution as shown in the next section.

4.2 Choice selection

Suppose we have a partial program and set of features, and have somehow found the optimal set of weights $\vec{w}$. We can now run the partial program, and whenever we reach a choice state ω , we need to pick the u maximizing $Q(\omega, u)$. In the multieffector case, this maximization is not straightforward. For example, in the Resource domain, if all the peasants are at navigation choices, there will be 5^N joint choices.

An advantage of using a linear approximation to the Q-function is that this maximization can often be done efficiently. When we reach a choice state ω , we form the *coordination graph* [Guestrin et al., 2002]. This is a graph containing a node for each choosing thread in ω . For every feature f , there is a clique in the graph among the choosing threads that f depends on. The maximizing joint choice can then be found in time exponential in the treewidth of this graph using *cost network dynamic programming* [Dechter, 1999].

In a naive implementation, the treewidth of the coordination graph might be too large. For example, in the Resource domain, there is a collision feature for each pair of peasants, so the treewidth can equal N . However, in a typical situation, most pairs of peasants will be too far apart to have any chance of colliding. To make use of this kind of “context-specificity”, we implement a feature template as a function that takes in a joint state ω and returns only the component features that are active in ω —the inactive features are equal to 0 regardless of the value of u . For example, the collision feature template $F_{\text{coll}}(\omega)$ would return one collision feature for each pair of peasants who are sufficiently close to each other to have some chance of colliding on the next step. This significantly reduces the treewidth.

4.3 Learning

Thanks to Theorem 1, learning the optimal completion of a partial program is equivalent to learning the Q-function in an SMDP. We use a Q-learning algorithm, in which we run the partial program in the environment, and when we reach a choice state, we pick a joint choice according to a GLIE exploration policy. We keep track of the accumulated reward and number of environment steps that take place between each pair of choice states. This results in a stream of samples of the form $(\omega, u, r, \omega', N)$. We maintain a running estimate of $\vec{w}$, and upon receiving a sample, we perform the update

$$\vec{w} \leftarrow \vec{w} + \alpha \left(r + \gamma^N \max_{u'} Q(\omega', u'; \vec{w}) - Q(\omega, u; \vec{w}) \right) \vec{f}(\omega, u)$$

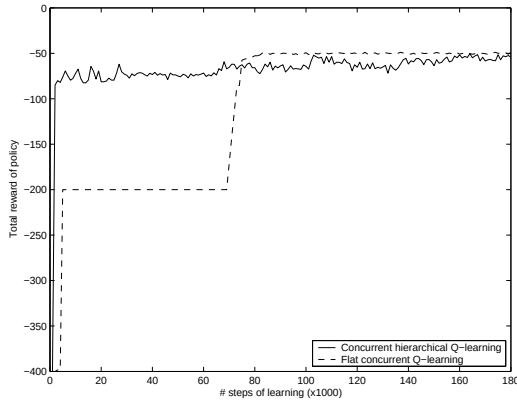


Figure 4: Learning curves for the 3x3 Resource domain with 4 peasants, in which the goal was to collect 20 resources. Curves averaged over 5 learning runs. Policies were evaluated by running until 200 steps or termination. No shaping was used.

4.4 Shaping

Potential-based shaping [Ng *et al.*, 1999] is a way of modifying the reward function R of an MDP to speed up learning without affecting the final learned policy. Given a *potential function* Φ on the state space, we use the new reward function $\tilde{R}(s, a, s') = R(s, a, s') + \gamma\Phi(s') - \Phi(s)$. As pointed out in [Andre and Russell, 2002], shaping extends naturally to hierarchical RL, and the potential function can now depend on the machine state of the partial program as well. In the Resource domain, for example, we could let $\Phi(\omega)$ be the sum of the distances from each peasant to his current navigation goal together with a term depending on how much gold and wood has already been gathered.

5 Experiments

We now describe learning experiments, first with the Resource domain, then with a more complex strategic domain. Full details of these domains and the partial programs and function approximation architectures used will be presented in a forthcoming technical report.

5.1 Running example

We begin with a 4-peasant, 3x3 Resource domain. Though this world seems quite small, it has over 10^6 states and 8^4 joint actions in each state, so tabular learning is infeasible. We compared flat coordinated Q-learning [Guestrin *et al.*, 2002] and the hierarchical Q-learning algorithm of Section 4.3. Figure 4 shows the learning curves. Within the first 100 steps, hierarchical learning usually reaches a “reasonable” policy (one that moves peasants towards their current goal while avoiding collisions). The remaining gains are due to improved allocation of peasants to mines and forests to minimize congestion. A “human expert” in this domain had an average total reward around -50 , so we believe the learnt hierarchical policy is near-optimal, despite the constraints in the partial program. After 100 steps, flat learning usually learns to avoid collisions, but still has not learnt that picking up resources is a good idea. After about 75000 steps, it

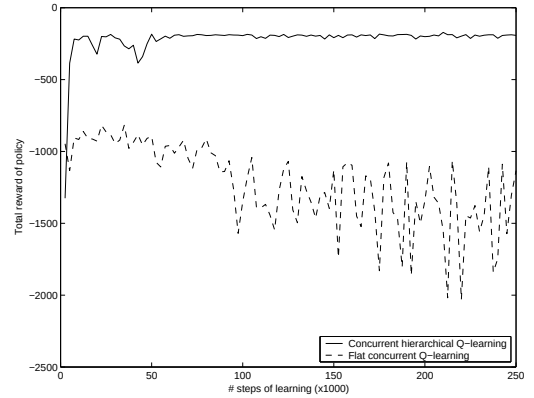


Figure 5: Learning curves for the 15x15 Resource domain with 20 peasants, in which the goal was to collect 100 resources. The shaping function from Section 4.4 was used.

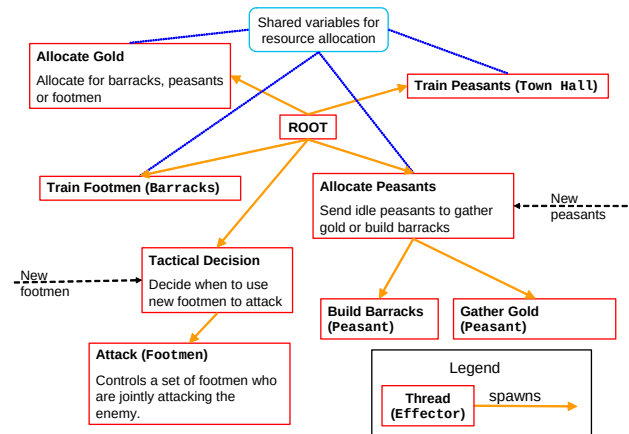


Figure 6: Structure of the partial program for the strategic domain

makes this discovery and reaches a near-optimal policy.

Our next test was on a 15x15 world with 20 peasants, with learning curves shown in Figure 5. Because of the large map size, we used the shaping function from Section 4.4 in the hierarchical learning. In flat learning, this shaping function cannot be used because the destination of the peasant is not part of the environment state. To write such a shaping function, the programmer would have to figure out in advance which destination each peasant should be allocated to as a function of the environment state. For similar reasons, it is difficult to find a good set of features for function approximation in the flat case, and we were not able to get flat learning to learn any kind of reasonable policy. Hierarchical learning learnt a reasonable policy in the sense defined earlier. The final learnt allocation of peasants to resource areas is not yet optimal as its average reward was about -200 whereas the human expert was able to get a total reward of -140 . We are working to implement the techniques of Section 6, which we believe will further speed convergence in situations with a large number of effectors.

5.2 Larger Stratagus domain

We next considered a larger Stratagus domain, in which the agent starts off with a single town hall, and must defeat a single powerful enemy as quickly as possible. An experienced human would first use the townhall to train a few peasants, and as the peasants are built, use them to gather resources. After enough resources are collected, she would then assign a peasant to construct barracks (and reassign the peasant to resource-gathering after the barracks are completed). The barracks can then be used to train footmen. Footmen are not very strong, so multiple footmen will be needed to defeat the enemy. Furthermore, the dynamics of the game are such that it is usually advantageous to attack with groups of footmen rather than send each footman to attack the enemy as soon as he is trained. The above activities happen in parallel. For example, a group of footmen might be attacking the enemy while a peasant is in the process of gathering resources to train more footmen.

Our partial program is based on the informal description above. Figure 6 shows the thread structure of the partial program. The choices to be made are about how to allocate resources, e.g., whether to train a peasant next or build a barracks, and when to launch the next attack. The initial policy performed quite poorly—it built only one peasant, which means it took a long time to accumulate resources, and sent footmen to attack as soon as they were trained. The final learned policy, on the other hand, built several peasants to ensure a constant stream of resources, sent footmen in groups so they did more damage, and pipelined all the construction, training, and fighting activities in intricate and balanced ways. We have put videos of the policies at various stages of learning on the web.³

6 Concurrent learning

Our results so far have focused on the use of concurrent partial programs to provide a concise description of multi-effector behaviour, and on associated mechanisms for efficient action selection. The learning mechanism, however, is not yet concurrent—learning does not take place at the level of individual threads. Furthermore, the Q -function representation does not take advantage of the *temporal Q -decomposition* introduced by Dietterich [2000] for MAXQ and extended by Andre and Russell [2002] for ALisp. Temporal Q -decomposition divides the complete sequence of rewards that defines a Q -value into subsequences, each of which is associated with actions taken within a given subroutine. This is very important for efficient learning because, typically, the sum of rewards associated with a particular subroutine usually depends on only a small subset of state variables—those that are specifically relevant to decisions within the subroutine. For example, the sum of rewards (step costs) obtained while a lone peasant navigates to a particular mine depends only on the peasant’s location, and not on the amount of gold and wood at the base.

We can see immediately why this idea cannot be applied directly to concurrent HRL methods: with multiple peasants,

there is no natural way to cut up the reward sequence, because rewards are not associated formally with particular peasants and because at any point in time different peasants may be at different stages of different activities. It is as if, every time some gold is delivered to base, all the peasants jump for joy and feel smug—even those who are wandering aimlessly towards a depleted forest square.

We sketch the solution to this problem in the resource domain example. Implementing it in the general case is ongoing work. The idea is to make use of the kind of *functional reward decomposition* proposed by Russell and Zimdars [2003], in which the global reward function is written as $R(s, a) = R_1(s, a) + \dots + R_N(s, a)$. Each sub-reward function R_j is associated with a different functional sub-agent—for example, in the Resource domain, R_j might reflect deliveries of gold and wood by peasant j , steps taken by j , and collisions involving peasant j . Since there is one thread per peasant, we can write $R(s, a) = \sum_{\psi \in \Psi} R_\psi(s, a)$ where Ψ is a fixed set of threads. This gives a thread-based decomposition of the Q -function $Q(\omega, u) = \sum_{\psi \in \Psi} Q_\psi(\omega, u)$. That is, Q_ψ refers to the total discounted reward gained in thread ψ .

Russell and Zimdars [2003] derived a decomposed SARSA algorithm for additively decomposed Q -functions of this form that converges to globally optimal behavior. The same algorithm can be applied to concurrent ALisp: each thread receives its own reward stream and learns its component of the Q -function, and global decisions are taken so as to maximize the sum of the component Q -values. Having localized rewards to threads, it becomes possible to restore the temporal Q -decomposition used in MAXQ and ALisp. Thus, we obtain fully concurrent reinforcement learning within the concurrent HRL framework, with both functional and temporal decomposition of the Q -function representation.

7 Related work

There are several HRL formalisms as mentioned above, but most are single-threaded. The closest related work is by Mahadevan’s group [Makar *et al.*, 2001], who have extended the MAXQ approach to concurrent activities (for many of the same reasons we have given). Their approach uses a fixed set of identical single-agent MAXQ programs, rather than a flexible, state-dependent concurrent decomposition of the overall task. However, they do not provide a semantics for the interleaved execution of multiple coordinating threads. Furthermore, the designer must stipulate a fixed level in the MAXQ hierarchy, above which coordination occurs and below which agents act independently and greedily. The intent is to avoid frequent and expensive coordinated decisions for low-level actions that seldom conflict, but the result must be that agents have to choose completely nonoverlapping high-level tasks since they cannot coexist safely at the low level. In our approach, the state-dependent coordination graph (Section 4.2) reduces the cost of coordination and applies it only when needed, regardless of the task level.

The issue of concurrently executing several high-level actions of varying duration is discussed in [Rohanimanesh and Mahadevan, 2003]. They discuss several schemes for when to make decisions. Our semantics is similar to their “continue”

³<http://www.cs.berkeley.edu/~bhaskara/ijcai05-videos>

scheme, in which a thread that has completed a high-level action picks a new one immediately and the other threads continue executing.

8 Conclusions and Further Work

We have described a concurrent partial programming language for hierarchical reinforcement learning and suggested, by example, that it provides a natural way to specify behavior for multieffector systems. Because of concurrent ALisp's built-in mechanisms for reinforcement learning and optimal, yet distributed, action selection, such specifications need not descend to the level of managing interactions among effectors—these are learned automatically. Our experimental results illustrate the potential for scaling to large numbers of concurrent threads. We showed effective learning in a Stratagus domain that seems to be beyond the scope of single-threaded and nonhierarchical methods; the learned behaviors exhibit an impressive level of coordination and complexity.

To develop a better understanding of how temporal and functional hierarchies interact to yield concise distributed value function representations, we will need to investigate a much wider variety of domains than that considered here. To achieve faster learning, we will need to explore model-based methods that allow for lookahead to improve decision quality and extract much more juice from each experience.

References

- [Andre and Russell, 2002] D. Andre and S. Russell. State abstraction for programmable reinforcement learning agents. In *AAAI*, 2002.
- [Andre, 2003] D. Andre. *Programmable Reinforcement Learning Agents*. PhD thesis, UC Berkeley, 2003.
- [Dechter, 1999] R. Dechter. Bucket elimination : a unifying framework for reasoning. *Artificial Intelligence*, 113:41–85, 1999.
- [Dietterich, 2000] T. Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *JAIR*, 13:227–303, 2000.
- [Guestrin *et al.*, 2002] C. Guestrin, M. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In *ICML*, 2002.
- [Guestrin *et al.*, 2003] C. Guestrin, D. Koller, C. Gearhart, and N. Kanodia. Generalizing plans to new environments in relational MDPs. In *IJCAI*, 2003.
- [Makar *et al.*, 2001] R. Makar, S. Mahadevan, and M. Ghavamzadeh. Hierarchical multi-agent reinforcement learning. In *ICRA*, 2001.
- [Ng *et al.*, 1999] A. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations : theory and application to reward shaping. In *ICML*, 1999.
- [Parr and Russell, 1997] R. Parr and S. Russell. Reinforcement learning with hierarchies of machines. In *Advances in Neural Information Processing Systems 9*, 1997.
- [Precup and Sutton, 1998] D. Precup and R. Sutton. Multi-time models for temporally abstract planning. In *Advances in Neural Information Processing Systems 10*, 1998.
- [Rohanimanesh and Mahadevan, 2003] K. Rohanimanesh and S. Mahadevan. Learning to take concurrent actions. In *NIPS*, 2003.
- [Russell and Zimdars, 2003] S. Russell and A. Zimdars. Q-decomposition for reinforcement learning agents. In *ICML*, 2003.

Topic and Role Discovery in Social Networks

Andrew McCallum, Andrés Corrada-Emmanuel, Xuerui Wang

Department of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003 USA
{mccallum, corrada, xuerui}@cs.umass.edu

Abstract

Previous work in social network analysis (SNA) has modeled the existence of links from one entity to another, but not the language content or topics on those links. We present the Author-Recipient-Topic (ART) model for social network analysis, which learns topic distributions based on the direction-sensitive messages sent between entities. The model builds on Latent Dirichlet Allocation (LDA) and the Author-Topic (AT) model, adding the key attribute that distribution over topics is conditioned distinctly on both the sender and recipient—steering the discovery of topics according to the relationships between people. We give results on both the Enron email corpus and a researcher’s email archive, providing evidence not only that clearly relevant topics are discovered, but that the ART model better predicts people’s roles.

1 Introduction and Related Work

Social network analysis (SNA) is the study of mathematical models for interactions among people, organizations and groups. With the recent availability of large datasets of human interactions [Shetty & Adibi, 2004; Wu et al., 2003], the popularity of services like Friendster.com and LinkedIn.com, and the salience of the connections among the 9/11 hijackers, there has been growing interest in social network analysis.

Historically, research in the field has been led by social scientists and physicists [Lorrain & White, 1971; Albert & Barabási, 2002; Watts, 2003; Wasserman & Faust, 1994], and previous work has emphasized binary interaction data, with directed and/or weighted edges. There has not, however, previously been significant work by researchers with backgrounds in statistical natural language processing, nor analysis that captures the richness of the *language contents* of the interactions—the words, the topics, and other high-dimensional specifics of the interactions between people.

Using pure network connectivity properties, SNA often aims to discover various categories of nodes in a network. For example, in addition to determining that a node-degree distribution is heavy-tailed, we can also find those particular nodes with an inordinately high number of connections, or with connections to a particularly well-connected subset of

the network. Furthermore, using these properties we can assign “roles” to certain nodes, *e.g.* [Lorrain & White, 1971; Wolfe & Jensen, 2003]. However, it is clear that network properties are not enough to discover all the roles in a social network. Consider email messages in a corporate setting, and imagine a situation where a tightly knit group of users trade email messages with each other in a roughly symmetric fashion. Thus, at the network level they appear to fulfill the same role. But perhaps, one of the users is in fact a manager for the whole group—a role that becomes obvious only when one accounts for the language content of the email messages.

Outside of the social network analysis literature, there has been a stream of new research in machine learning and natural language models for clustering words in order to discover the few underlying topics that are combined to form documents in a corpus. Latent Dirichlet Allocation [Blei et al., 2003] robustly discovers multinomial word distributions of these topics. Hierarchical Dirichlet Processes [Teh et al., 2004] can determine an appropriate number of topics for a corpus. The Author-Topic Model [Steyvers et al., 2004] learns topics conditioned on the mixture of authors that composed a document. However, none of these models are appropriate for SNA, in which we aim to capture the directed interactions and relationships between people.

The paper presents the *Author-Recipient-Topic* (ART) model, a directed graphical model of words in a message generated given their author and a set of recipients. The model is similar to the Author-Topic (AT) model, but with the crucial enhancement that it conditions the per-message topic distribution jointly on both the author and individual recipients, rather than on individual authors. Thus the discovery of topics in the ART model is influenced by the social structure in which messages are sent and received. Each topic consists of a multinomial distribution over words. Each author-recipient pair has a distribution over topics. We can also easily calculate marginal distributions over topics conditioned solely on an author, or solely on a recipient, in order to find the topics on which each person is most likely to send or receive.

Most importantly, we can also effectively use these person-conditioned topic distributions to measure similarity between people, and thus discover people’s roles by clustering using this similarity. For example, people who receive messages containing requests for photocopying, travel bookings, and meeting room arrangements can all be said to have the role

“administrative assistant,” and can be discovered as such because in the ART model they will all have these topics with high probability in their receiving distribution. Note that we can discover that two people have similar roles even if in the graph they are connected to very different sets of people.

We demonstrate this model on the Enron email corpus comprising 147 people and 23k messages, and also on about 10 months of incoming and outgoing mail of the first author, comprising 825 people and 23k messages. We show not only that ART discovers extremely salient topics, but also gives evidence that ART predicts people’s roles better than AT and SNA. Furthermore we show that the similarity matrix produced by ART is different from both the SNA matrix and the AT matrix in several appropriate ways.

We also describe an extension of the ART model that explicitly captures *roles* of people, by generating role associations for the authors and recipients of a message, and conditioning the topic distributions on role assignments. The model, which we term *Role-Author-Recipient-Topic* (RART), naturally represents that one person can have more than one role. We describe several possible RART variants, and describe preliminary experiments with one of these variants.

2 Author-Recipient-Topic Models

Before describing the ART model, we first describe three related models. Latent Dirichlet Allocation (LDA) is a Bayesian network that generates a document using a mixture of topics [Blei et al., 2003]. In its generative process, for each document d , a multinomial distribution θ over topics is randomly sampled from a Dirichlet with parameter α , and then to generate each word, a topic z is chosen from this topic distribution, and a word, w , is generated by randomly sampling from a topic-specific multinomial distribution ϕ_z . The robustness of the model is greatly enhanced by integrating out uncertainty about the per-document topic distribution θ .

The Author model (also termed a Multi-label Mixture Model) [McCallum, 1999], is a Bayesian network that simultaneously models document content and its authors’ interests with a 1-1 correspondence between topics and authors. For each document d , a set of authors $\mathbf{a}_d$ is observed. To generate each word, an author, z , is sampled uniformly from the set, and then a word, w , is generated by sampling from an author-specific multinomial distribution ϕ_z . The Author-Topic (AT) model is a similar Bayesian network, in which each authors’ interests are modeled with a *mixture* of topics [Steyvers et al., 2004]. In its generative process for each document d , a set of authors, $\mathbf{a}_d$, is observed. To generate each word, an author x is chosen uniformly from this set, then a topic z is selected from a topic distribution θ_x that is specific to the author, and then a word w is generated from a topic-specific multinomial distribution ϕ_z . However, as described previously, none of these models is suitable for modeling message data.

An email message has one sender and in general more than one recipients. We could treat both the sender and the recipients as “authors” of the message, and then employ the AT model, but this does not distinguish the author and the recipients of the message, which is undesirable in many real-world situations. A manager may send email to a secretary and vice

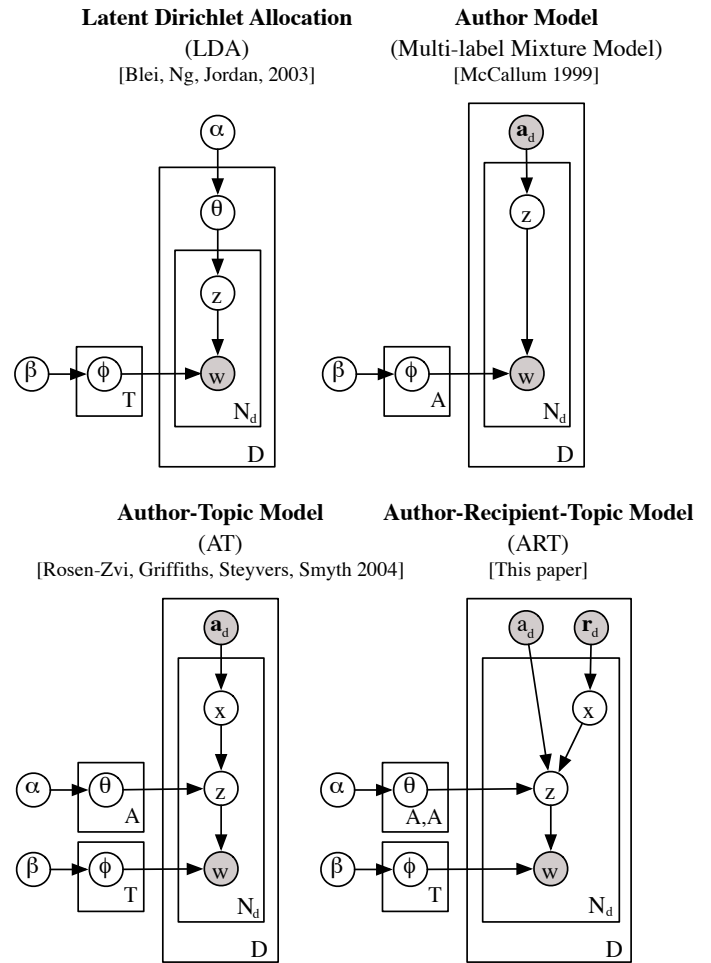


Figure 1: Three related models, and the ART model. In all models, each observed word, w , is generated from a multinomial word distribution, ϕ_z , specific to a particular topic/author, z , however topics are selected differently in each of the models.

versa, but the nature of the requests and language used may be quite different. Even more dramatically, consider the large quantity of junk email that we receive; modeling the topics of these messages as undistinguished from the topics we write about as authors would be extremely confounding and undesirable since they do not reflect our expertise or roles.

Alternatively we could still employ the AT model by ignoring the recipient information of email and treating each email document as if it only has one author. However, in this case (which is similar to the LDA model) we are losing all information about the recipients, and the connections between people implied by sender-recipient relationships.

Thus, we propose an Author-Recipient-Topic (ART) model for message data. The ART model captures topics and the directed social network of senders and recipients by conditioning the multinomial distribution over topics distinctly on both the author and one recipient of a message. Unlike the AT, the

ART model takes into consideration both author and recipients distinctly, in addition to modeling the email content as a mixture of topics.

The ART model is a Bayesian network that simultaneously models message content, as well as the directed social network in which the messages are sent. In its generative process, for each message d , an author, a_d , and a set of recipients, $\mathbf{r}_d$, are observed. To generate each word, a recipient, x , is chosen uniformly from $\mathbf{r}_d$, and then a topic z is chosen from a multinomial topic distribution $\theta_{a_d, x}$, where the distribution is specific to the author-recipient pair (a_d, x) . Finally, the word w is generated by sampling from a topic-specific multinomial distribution ϕ_z . The result is that the discovery of topics is guided by the social network in which the collection of message text was generated.

The Bayesian network for all models is shown in Figure 1.

In the ART model, given the hyperparameters α and β , an author $\mathbf{a}$, and a set of recipients $\mathbf{r}$, the joint distribution of the topic mixtures θ , the word mixtures ϕ , a set of recipients $\mathbf{x}$, a set of topics $\mathbf{z}$ and a set of words $\mathbf{w}$ in the corpus is given by:

$$p(\theta, \phi, \mathbf{x}, \mathbf{z}, \mathbf{w} | \alpha, \beta, \mathbf{a}, \mathbf{r}) \\ = p(\theta | \alpha) p(\phi | \beta) \prod_{d=1}^D \prod_{n=1}^{N_d} p(x_{dn} | \mathbf{r}_d) p(z_{dn} | \theta_{a_d, x_{dn}}) p(w_{dn} | \phi_{z_{dn}}).$$

Integrating over θ and ϕ , and summing over $\mathbf{x}$ and $\mathbf{z}$, we get the marginal distribution of a corpus:

$$p(\mathbf{w} | \alpha, \beta, \mathbf{a}, \mathbf{r}) = \iint p(\theta | \alpha) p(\phi | \beta) \prod_{d=1}^D \prod_{n=1}^{N_d} \sum_{x_{dn}} \sum_{z_{dn}} p(x_{dn} | \mathbf{r}_d) \\ \cdot p(z_{dn} | \theta_{a_d, x_{dn}}) p(w_{dn} | \phi_{z_{dn}}) d\phi d\theta.$$

3 Experimental Results

We present results with the Enron email corpus and the personal email of the first author of this paper (McCallum). After preprocessing, the Enron corpus we use contains 147 users and 23,488 emails, and the McCallum dataset consists of 23,488 messages written by 825 authors, sent or received by McCallum during Jan.-Oct., 2004. Gibbs sampling is employed to conduct all experiments (as detailed in [McCallum et al., 2004]).

3.1 Topics and Prominent Relations from ART

Table 1 shows the highest probability words from six topics in an ART model trained on the 147 Enron users with 50 topics. (The quoted titles are our own interpretation of a summary for the topics.) The clarity and specificity of these topics are typical of the topics discovered by the model.

Beneath the word distribution for each topic are the three author-recipient pairs with highest probability of discussing that topic—each pair separated by a horizontal line, with the author above the recipient. For example, Hain, the top author of messages in the “Legal Contracts” topic, was an in-house lawyer at Enron. In the “Operations” topic, it is satisfying to see Beck, who was the Chief Operating Officer at Enron. In the “Government Relations” topic, we see Dasovich, who was a Government Relation Executive, Shapiro who was Vice

Topic 5 “Legal Contracts”		Topic 17 “Doc. Review”		Topic 27 “Time Scheduling”	
section	0.0299	attached	0.0742	day	0.0419
party	0.0265	agreement	0.0493	friday	0.0418
language	0.0226	review	0.0340	morning	0.0369
contract	0.0203	questions	0.0257	monday	0.0282
date	0.0155	draft	0.0245	office	0.0282
enron	0.0151	letter	0.0239	wednesday	0.0267
parties	0.0149	comments	0.0207	tuesday	0.0261
notice	0.0126	copy	0.0165	time	0.0218
days	0.0112	revised	0.0161	good	0.0214
include	0.0111	document	0.0156	thursday	0.0191
M.Hain	0.0549	G.Nemec	0.0737	J.Dasovich	0.0340
J.Steffes		B.Tycholiz		R.Shapiro	
J.Dasovich	0.0377	G.Nemec	0.0551	J.Dasovich	0.0289
R.Shapiro		M.Whitt		J.Steffes	
D.Hyvl	0.0362	B.Tycholiz	0.0325	C.Clair	0.0175
K.Ward		G.Nemec		M.Taylor	
Topic 34 “Operations”		Topic 37 “Power Market”		Topic 41 “Gov. Relations”	
operations	0.0321	market	0.0567	state	0.0404
team	0.0234	power	0.0563	california	0.0367
office	0.0173	price	0.0280	power	0.0337
list	0.0144	system	0.0206	energy	0.0239
bob	0.0129	prices	0.0182	electricity	0.0203
open	0.0126	high	0.0124	davis	0.0183
meeting	0.0107	based	0.0120	utilities	0.0158
gas	0.0107	buy	0.0117	commission	0.0136
business	0.0106	customers	0.0110	governor	0.0132
houston	0.0099	costs	0.0106	prices	0.0089
S.Beck	0.2158	J.Dasovich	0.1231	J.Dasovich	0.3338
L.Kitchen		J.Steffes		R.Shapiro	
S.Beck	0.0826	J.Dasovich	0.1133	J.Dasovich	0.2440
J.Lavorato		R.Shapiro		J.Steffes	
S.Beck	0.0530	M.Taylor	0.0218	J.Dasovich	0.1394
S.White		E.Sager		R.Sanders	

Table 1: An illustration of several topics from a 50-topic run for the Enron Email Dataset. Each topic is shown with the top 10 words and their corresponding conditional probabilities. The quoted titles are our own summary for the topics. Below are prominent author-recipient pairs for each topic.

President of Regulatory Affairs, and Steffes, who was Vice President of Government Affairs. Results on the McCallum email dataset are reported in Table 2.

3.2 Stochastic Blockstructures and Roles

The stochastic equivalence hypothesis from SNA states that nodes in a network that behave stochastically equivalently must have similar roles. In the case of an email network consisting of message counts, the natural way to measure equivalence is to examine the probability that a node communicated with other nodes. If two nodes have similar probability distribution over their communication partners, we should consider them role-equivalent. We can measure a similarity symmetrically by calculating the Jensen-Shannon (JS) divergence, and inverting it.

Standard recursive graph-cutting algorithms on this matrix can be used to cluster users, rearranging the rows/columns to

Topic 5 “Grant Proposals”	Topic 31 “Meeting Setup”	Topic 38 “ML Models”	Topic 41 “Friendly Discourse”
proposal	today	model	great
data	tomorrow	models	good
budget	time	inference	don
work	ll	conditional	sounds
year	meeting	methods	work
glenn	week	number	wishes
nsf	talk	sequence	talk
project	meet	learning	interesting
sets	morning	graphical	time
support	monday	random	hear

smyth	ronb	casutton	mccallum
mccallum	mccallum	mccallum	culotta
mccallum	wellner	icml04-webadmin	mccallum
stowell	mccallum	icml04-chairs	casutton
mccallum	casutton	mccallum	ronb
lafferty	mccallum	casutton	mccallum
mccallum	mccallum	nips04workflow	mccallum
smyth	casutton	mccallum	saunders
pereira	mccallum	weinman	mccallum
lafferty	wellner	mccallum	pereira

Table 2: The four topics most prominent in McCallum’s email exchange with smyth (Padhraic Smyth), from a 50-topic run of ART on 10 months of McCallum’s email. The topics provide an extremely salient summary of McCallum and Smyth’s relationship during this time period: they wrote a grant proposal together; they set up many meetings; they discussed machine learning models; they were friendly with each other. Below are prominent author-recipient pairs for each topic. The people other than smyth also appear in very sensible associations: stowell is McCallum’s proposal budget administrator; McCallum also wrote a proposal with lafferty (John Lafferty) and pereira (Fernando Pereira); McCallum also sets up meetings, discusses machine learning and has friendly discourse with his graduate student advisees: ronb, wellner, casutton, and culotta; he does not, however, discuss the details of proposal-writing with them.

form approximately block-diagonal structures. This is the familiar process of ‘blockstructuring’ used in SNA. We perform such an analysis on two datasets: a small subset of the Enron users consisting mostly of people associated with the Transwestern Pipeline Division within Enron, and the entirety of McCallum’s email.

Beginning with the Enron data, Figure 2 shows the results from traditional SNA (in this case, JS divergence of distributions on recipients from each sender), ART (JS divergence of recipient-marginalized topic distributions for each sender) and AT (using the topics distributions from AT instead of our ART). Darker shading indicates higher similarity between people.

Consider Enron employee Geaconne (user 9 in all the matrices in Figure 2). According to the traditional SNA role measurement, Geaconne and McCarty (user 8) have very similar roles, however, both the AT and ART models indicate no special similarity. Inspection of the data reveals that Geaconne was an Executive Assistant, while McCarty was a Vice-

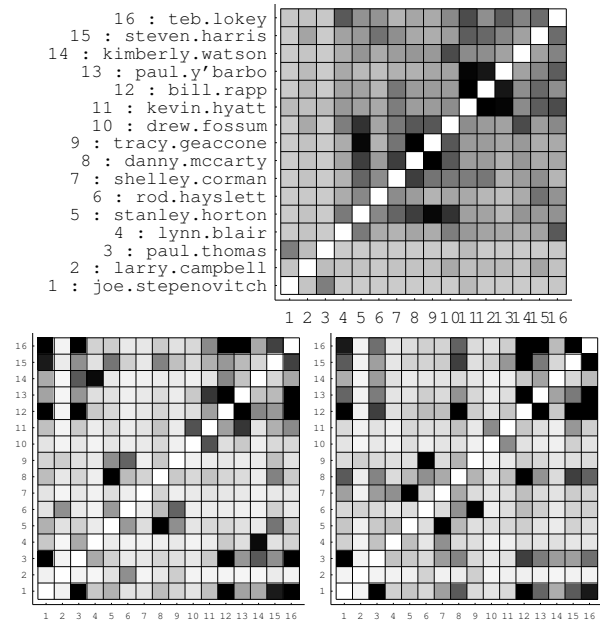


Figure 2: **Top:** SNA Inverse JS Network. **Left Bottom:** ART Inverse JS Network. **Right Bottom:** AT Inverse JS Network. Darker shades indicate higher similarity.

President—rather different roles—and, thus output of ART and AT is more appropriate. Thus, SNA analysis shows that they wrote email to similar sets of people, but the ART analysis illustrates that they used very different language.

Here ART and AT provide similar role distance, but they show their differences elsewhere. For example, AT indicates a very strong role similarity between Geaconne and Hayslett (user 6), who was her boss (and CFO & Vice President in the Division); on the other hand, ART more correctly designates a low role similarity for this pair—in fact, ART assigns low similarity between Geaconne and all others in the matrix, which is appropriate because she is the only executive assistant in this small sample of Enron employees.

Another interesting pair is Blair (user 4) and Watson (user 14). ART predicts them to be role-similar, while the SNA and AT models do not. ART’s prediction seems more appropriate since Blair worked on “gas pipeline logistics” and Watson worked on “pipeline facility planning”, two very similar jobs.

Based on the above examples, and other similar examples, we would claim that the ART model is clearly better than the SNA model in predicting role-equivalence between users, and somewhat better than the AT model in this capacity.

We also carried out this analysis with the personal email for McCallum to further validate the difference between the ART and SNA predictions. There are 825 users in this email corpus. Table 3 shows the closest pairs, as calculated by the ART model and SNA model. The difference in quality between the ART and SNA halves of the table is striking.

Almost all the pairs predicted by the ART model look reasonable while many of those predicted by SNA are not at all. For example, ART matches mike and mikem are actually two different email addresses for the same person. (Most

Pairs considered most alike by ART	
User Pair	Description
editor reviews	Both journal review management
mike mikem	Same person! (manual coref error)
aepshtey smucker	Both students in McCallum's class
coe laurie	Both UMass admin assistants
mcollins mitchell	Both ML researchers on SRI project
Pairs considered most alike by SNA	
User Pair	Description
aepshtey rasmith	Both students in McCallum's class
donna editor	Spouse is unrel. to journal editor
donna krishna	Spouse is unrel. to conf. organizer
donna ramshaw	Spouse is unrel. to researcher at BBN
donna reviews	Spouse is unrel. to journal editor

Table 3: Pairs considered most alike by ART and SNA on McCallum email. All pairs produced by the ART model are accurately quite similar. This is not so for the top SNA pairs. Many users are considered similar by SNA merely because they appear in the corpus mostly sending email only to McCallum. However, this causes people with very different roles to be incorrectly declared similar—such as McCallum’s spouse (donna) and the JMLR editor.

other correferent email addresses were pre-collapsed by hand during preprocessing; here ART has pointed out out a mistaken omission, indicating the potential for ART to be used as a helpful component of an automated coreference system.) Users coe and laurie are both UMass CS Department administrative assistants; they rarely send email to the same people, but they write about similar things. On the other hand, the pairs declared most similar by the SNA model are mostly extremely poor. Most of the pairs include donna, and indicate pairs of people who are similar only because in this corpus they appeared mostly sending email only to McCallum, and not others. User donna is McCallum’s spouse.

4 Role-Author-Recipient-Topic Models

To better explore the roles of authors, an additional level of latent variables can be introduced to explicitly model roles. Of particular interest is capturing the notion that a person can have multiple *roles* simultaneously—for example, a person can be both a professor and a mountain climber. Each role is associated with a set of topics, and these topics may overlap. For example, professors’ topics may prominently feature research, meeting times, grant proposals, and friendly relations; climbers’ topics may prominently feature mountains, climbing equipment, and also meeting times and friendly relations.

We incorporate into the ART model a new set of variables that take on values indicating role, and we term this augmented model the *Role-Author-Recipient-Topic* (RART) model. In RART, authors, roles and message-contents are modeled simultaneously. Each author has a multinomial distribution over roles. Authors and recipients are mapped to some role assignments, and a topic is selected based on these roles. Thus we have a clustering model, in which appearances of topics are the underlying data, and sets of corre-

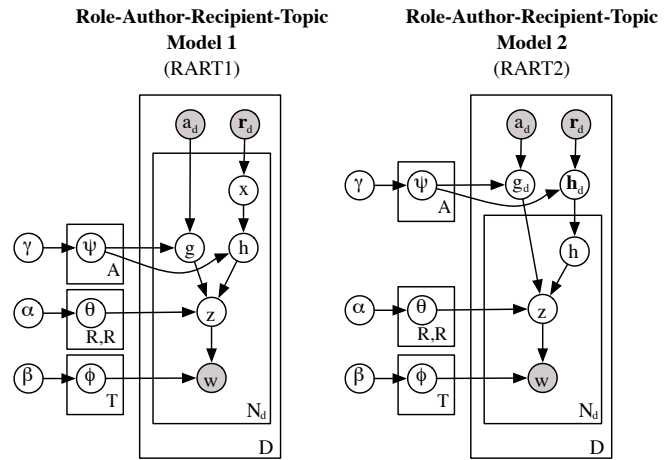


Figure 3: Two possible variants for the Role-Author-Recipient-Topic (RART) model.

lated topics gather together clusters that indicate roles. Each sender-role and recipient-role pair has a multinomial distribution over topics, and each topic has a multinomial distribution over words.

As shown in Figure 3, different strategies can be employed to incorporate the “role” latent variables. First in RART1, role assignments can be made separately for each word in a document. This model represents that a person can change role during the course of the email message. In RART2, on the other hand, a person chooses one role for the duration of the message. Here each recipient of the message selects a role assignment, and then for each word, a recipient (with corresponding role) is selected on which to condition the selection of topic. Some other variants are possible, for example (not shown in Figure 3), the recipients together result in the selection of a common, shared role, which is used to condition the selection of every word in the message. This last model may help capture the fact that a person’s role may depend on the other recipients of the message, but also restricts all recipients to a single role.

The generative process of RART models is similar to that for ART, as described in Section 2. The Gibbs sampling formulae for RART models can be derived in the same way as for ART, but in a more complex form.

5 Experimental Results with RART

Preliminary experiments have been conducted with the RART1 model. Because we introduce two sets of additional latent variables (author role and recipient role), the sampling procedure at each iteration is significantly more complex. To make inference more efficient, we can instead perform it in two distinct parts. One strategy we have found useful is to first train an ART model, and use a sample to obtain topic assignments and recipient assignments for each word token. Then, in the next stage, we treat topics and recipients as observed (locked). Although such a strategy may not be recommended for arbitrary graphical models, we feel this is rea-

sonable here because we find that a single sample from Gibbs sampling on the ART model to yield good assignments. The following results are based on a 15-group, 50-topic run of RART1 on McCallum email dataset.

Our results show that the RART model indeed clearly discovers automatically person-role information by its explicit inclusion of a role variables. For example, the users most prominent in Role 3 include, (in probability order): olc, gauthier, irsystem, system, allan, valerie, tech, and steve. They are all employees (or mailing lists) of the IT support staff at UMass CS, except for allan, who, however, was the professor chairing the department's computing committee. Role 4 seems to represent "working on the SRI CALO project." Its most prominent members include (in probability order): pereira, claire, israel, moll, mgervasio, melinda.gervasio, majordomo, and colin.evans. Most of them are researchers working on CALO project, many of them at SRI. The sender majordomo sends messages from an SRI CALO mailing list. The users mgervasio and melinda.gervasio are actually the same person; satisfyingly RART found that they have very similar role distributions.

One objective of the RART model is to capture the multiple roles that a person has. For example, user allan (James Allan) mentioned above has a role in "IT support," but also has a role as "researcher." Consider also user pereira (Fernando Pereira); his top five role assignments are Role 2 "NLP research", Role 4 "SRI CALO", Role 6 "proposal writing," Role 10 "grant issues," and Role 8 "guests at McCallum's house"—all exactly appropriate, as viewed through McCallum's email. Note that the difference between roles can be subtle, for example, Role 6 and Role 10 overlap.

As expected, one can observe interesting differences in the sender versus recipient topic distributions associated with each role. For instance, in Role 4 "SRI CALO," the top three topics for a sender role are Topic 27 "CALO information," Topic 11 "mail accounts," and Topic 36 "program meetings," but for its recipient roles, most prominent are Topic 48 "task assignments," Topic 46 "a particular CALO-related research paper," and Topic 40 "java code". Space limitations prevent inclusion of tables showing the full distributions associated many topics, roles and people, however these will be available in an accompanying technical report.

6 Conclusions

We have presented the Author-Recipient-Topic model, a Bayesian network for social network analysis that discovers discussion topics conditioned on the sender-recipient relationships in a corpus of messages. To the best of our knowledge, this model combines for the first time the directionalized connectivity graph from social network analysis with the clustering of words to form topics from probabilistic language modeling.

The model can be applied to discovering topics conditioned on message sending relationships, clustering to find social roles, and summarizing and analyzing large bodies of message data. The model would form a useful component in systems for routing requests, expert-finding, message recommendation and prioritization, and understanding the interac-

tions in an organization in order to make recommendations about improving organizational efficiency.

The Role-Author-Recipient-Topic (RART) models explicitly capture the multiple roles of people, based on messages sent and received. Additional work on other models that explicitly capture roles and groups is ongoing.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval, the Central Intelligence Agency, the National Security Agency, the National Science Foundation under NSF grant #IIS-0326249, and by the Defense Advanced Research Projects Agency, through the Department of the Interior, NBC, Acquisition Services Division, under contract #NBCHD030010.

References

- Albert, R., & Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74, 47–97.
- Blei, D. M., Ng, A. Y., & Jordan, M. J. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.
- Lorrain, F., & White, H. C. (1971). The structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1, 49–80.
- McCallum, A. (1999). Multi-label text classification with a mixture model trained by EM. *AAAI Workshop on Text Learning*.
- McCallum, A., Corrada-Emmanuel, A., & Wang, X. (2004). *The author-recipient-topic model for topic and role discovery in social networks: Experiments with enron and academic email* (Technical Report). University of Massachusetts, Amherst, UM-CS-2004-096.
- Shetty, J., & Adibi, J. (2004). *The Enron email dataset database schema and brief statistical report* (Technical Report). Information Sciences Institute.
- Steyvers, M., Smyth, P., Rosen-Zvi, M., & Griffiths, T. (2004). Probabilistic author-topic models for information discovery. *The Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2004). *Hierarchical dirichlet processes* (Technical Report). UC Berkeley Statistics.
- Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge University Press.
- Watts, D. J. (2003). *Six degrees: The science of a connected age*. Norton.
- Wolfe, A. P., & Jensen, D. (2003). Playing multiple roles: Discovering overlapping roles in social networks. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Wu, F., Huberman, B. A., Adamic, L. A., & Tyler, J. R. (2003). Information flow in social groups. <http://arXiv.org/abs/cond-mat/0305305>.

Training without data: Knowledge Insertion into RBF Neural Networks

Ken McGarry and Stefan Wermter

University of Sunderland

Department of Computing and Technology

David Goldman Informatics Centre, St Peters Campus, St Peters Way, SR6 ODD

ken.mcgarry@sunderland.ac.uk, www.his.sunderland.ac.uk

Abstract

A major problem when developing neural networks for machine diagnostics situations is that no data or very little data is available for training on fault conditions. However, the domain expert often has a good idea of what to expect in terms of input and output parameter values. If the expert can express these relationships in the form of rules, this would provide a resource too valuable to ignore. Fuzzy logic is used to handle the imprecision and vagueness of natural language and provides this additional advantage to a system. This paper investigates the development of a novel knowledge insertion algorithm that explores the benefits of pre-structuring RBF neural networks by using prior fuzzy domain knowledge and previous training experiences. Pre-structuring is accomplished by using fuzzy rules gained from a domain expert and using them to modify existing Radial Basis Function (RBF) networks. The benefits and novel achievements of this work enable RBF neural networks to be trained without actual data but to rely on input to output mappings defined through expert knowledge.

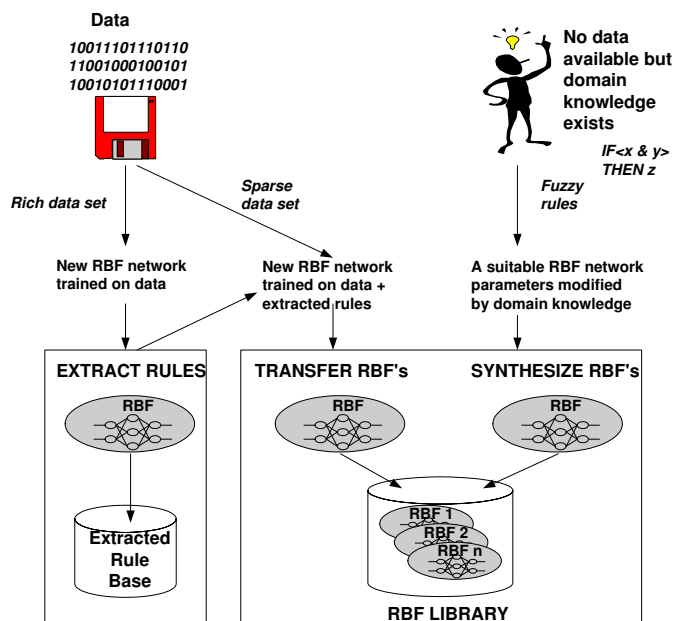


Figure 1: Overview of Architectural Transformations

1 Introduction

Radial Basis Function (RBF) neural networks are a form of local learning and are an efficient alternative to Multi-layer Perceptron (MLP) neural networks. In this paper we investigate how symbolic knowledge can be inserted into an RBF network, the advantage being to capitalize on those situations where no training data exists but an domain expert may be able to formulate IF..THEN rules. Figure 1 presents an overview of a much larger system that was developed, the work on rule extraction and knowledge transfer described elsewhere. It is the module related to the synthesis of RBF networks that is the focus of this paper.

Neural networks, like any other inductive learning algorithm such as decision trees and clustering techniques are data driven. Without a sufficient supply of training data these algorithms cannot be used to produce a reliable model of the problem domain. In such cases, the usual solution to these problems range from duplicating data with additional noise

added or to simulate the data. The extraction of knowledge in the form of rules has been successfully explored before on RBF networks using the *hREX* algorithm [McGarry *et al.*, 2001]. This work inspired the authors to develop knowledge synthesis or knowledge insertion by manipulating the RBF network parameters but information flow/extraction was in the opposite direction.

Domain experts often express their knowledge in vague and imprecise linguistic terms. It is therefore a natural step to use the ability of fuzzy sets and fuzzy logic to model this imprecision. Fuzzy sets and rules created by the knowledge elicited from a domain expert are used to manipulate the parameters of selected RBF hidden units. Knowledge synthesis is a technique to be used in those situations where data for training is unavailable but specific knowledge about the application is at hand. Converting the fuzzy rules into RBF network format avoids the integration issue of assigning con-

fidence factors when interpreting conflicting module outputs. Therefore, the local representation characteristics of the RBF networks are used to integrate hidden units placed by data driven learning with artificially synthesized hidden units.

The remainder of this paper is structured as follows: section two describes the Radial Basis Function neural network, and the how certain functional equivalences to fuzzy systems are exploited. Section three describes the work on knowledge synthesis for RBF networks for improved performance. Section four outlines the details of the fuzzy system employed and how it integrates the domain rules with the RBF network. Section five discusses the experimental results. Section six provides a brief overview of the related work and finally section seven presents the conclusions and areas for future work.

2 Radial Basis Function Networks

Radial basis function (RBF) neural networks are a model that has functional similarities found in many biological neurons. In biological nervous systems certain cells are responsive to a narrow range of input stimuli, for example in the ear there are cochlear stereocilia cells which are locally tuned to particular frequencies of sound [Moody and Darken, 1989]. The RBF network consists of a feedforward architecture with an input layer, a hidden layer of RBF "pattern" units and an output layer of linear units. The input layer simply transfers the input vector to the hidden units, which form a localized response to the input pattern. The activation levels of the output units provide an indication of the nearness of the input vector to the classes. Learning is normally undertaken as a two-stage process.

The radial basis functions in the hidden layer are implemented by kernel functions, which operate over a localized area of input space. The effective range of the kernels is determined by the values allocated to the centre and width of the radial basis function. The Gaussian function has a response characteristic determined by equation 1.

$$Z_j(x) = \exp\left(-\frac{\|x - \mu\|^2}{\sigma_j^2}\right) \quad (1)$$

where: μ = n-dimensional parameter vector, σ = width of receptive field, and x = input vector.

The response of the output units is calculated using equation 2.

$$\sum_{j=1}^J W_{lj} Z_j(x) \quad (2)$$

where: W = weight matrix, Z = hidden unit activations and x = input vector.

RBF networks are an appropriate choice for both classification tasks and function approximation. It is interesting to note that already Jang [Jang and Sun, 1993] discovered certain functional similarities exist between RBF networks and fuzzy systems. Jang identified five criteria that are necessary for the two techniques to become functionally equivalent.

1. The number of RBF units must be equal to the number of fuzzy rules.

2. The output of each fuzzy rule is a constant.
3. The membership functions within each rule are gaussians with the same widths.
4. The t-norm operator computes the firing strength of each rule through multiplication.
5. The RBF network and the fuzzy inference system both use either normalized or unnormalized output calculations.

The main advantage of the functional equivalence relationship is the ability of using one model's learning rules for the other and vice-versa. Other Fuzzy-RBF researchers [Hunt *et al.*, 1996; Halgamuge, 1997] have extended Jang's work to produce hybrid learning models. Fuzzy models are generally robust when faced with noisy and missing data and are more comprehensible than a purely neural network approach [Sun, 1994].

3 Knowledge Synthesis

Knowledge synthesis is a technique intended for those situations in which no actual training data is available but some form of domain knowledge is at hand. The experts knowledge is encoded as fuzzy sets and rules which are used to synthesize new hidden unit parameters for incorporation into a new or existing network. The fuzzy rules describe a set of output classes and the possible input values denoting their characteristics (based on the experts opinions). The objective of converting from fuzzy rules to RBF networks is to have the knowledge in a consistent format. It would be possible to have the domain knowledge in the form of a stand-alone fuzzy module, interacting with the RBF based system in some loosely or tightly coupled protocol. However, by converting the fuzzy rules into an RBF architecture they can be subjected to further analysis by rule extraction and it also avoids hybrid system integration issues.

The RBF hidden layer parameters (spread and centre values) can be converted from fuzzy sets by the rule defined by [Jang and Sun, 1993]:

$$\mu_{A_1}(x_1) = \exp\left[-\frac{(x_1 - c_{A_1})^2}{\sigma_1^2}\right] \quad (3)$$

Where: $\mu_{A_1}(x_1)$ is the fuzzy linguistic labels containing the domain knowledge, c is the centre of the receptive field i.e. the RBF function, σ_1^2 is the RBF spread. Modification of the RBF output layer parameters was accomplished by use of the pseudoinverse matrix X^P [Kubat, 1998].

$$W = (X^T X)^{-1} X^T C = X^P C \quad (4)$$

Where: C is the classification matrix containing class labels, W is the hidden-to-output unit weight matrix, X is the matrix containing the converted information from fuzzy sets (gained from eqn 3).

3.1 The Problem Domain

An industrial machine vibration data set was chosen because of the availability of several domain rules relating spectral data to fault conditions. This knowledge is available in text

books, journals and rules taken from a large European collaborative project.

A common problem encountered with many diagnostic applications is the lack of data for certain system conditions especially fault situations. The diagnosis data obtained for the experimental work described in this paper overcame such problems by using a mixture of simulated and test rig data. This data enabled an experiment to be conducted which assumed that data collected at low motor speeds was available but not at high speeds. At high motor speeds the values of several input features can be expected to change (increase or decrease) also the severity of the faults may appear different. However, by using vibration theory heuristics it was possible to predict how certain parameters and fault conditions would behave at higher speeds. The following method was used for the vibration data experiment:

3.2 Methodology

1. The vibration data was divided into two groups, i. motor speed classed as low and ii. motor speed classed as high.
2. An RBF network was trained on the low speed data and the accuracy was examined with a separate low speed test data.
3. The RBF network was tested with the high speed data (i.e. the data it was *not* trained on) and the accuracy was noted.
4. The domain rules were converted into fuzzy sets and fuzzy rules.
5. The *h*REX algorithm was used to assign hidden units to the output classes.
6. For each of the three fault classes considered the hidden units assigned to them were duplicated and modified using the fuzzy inferencing system.
7. The modified RBF network had its hidden-to-output weights and biases recalculated.
8. The modified network was re-tested on the high speed data and the accuracy noted.

The vibration data was divided into two sets based on the speed of the motor: low speed data consisting of examples with motor speeds of 500-1000 RPM (rotations per minute) and high speed data containing data from motor speeds of 1500-2000 RPM. However, both sets of data still consisted of 11 input features and eight output classes. An RBF network was trained on the low speed data and its accuracy was observed on both low speed and high speed test data. The results are shown in figure 2.

The RBF network trained on low speed data has an accuracy 90% on the low speed test set. Unfortunately, the accuracy falls dramatically to 51.3% when introduced to the high speed data. This is a common phenomena that RBF neural networks can generate well to new patterns only if they stem from the same distribution of input patterns. If the task is very different they do not generalise well. The unreliability of such a classifier would prevent it from being deployed in any application running at high speed. Therefore, a substantial increase in accuracy is required for knowledge synthesis to be of practical use.

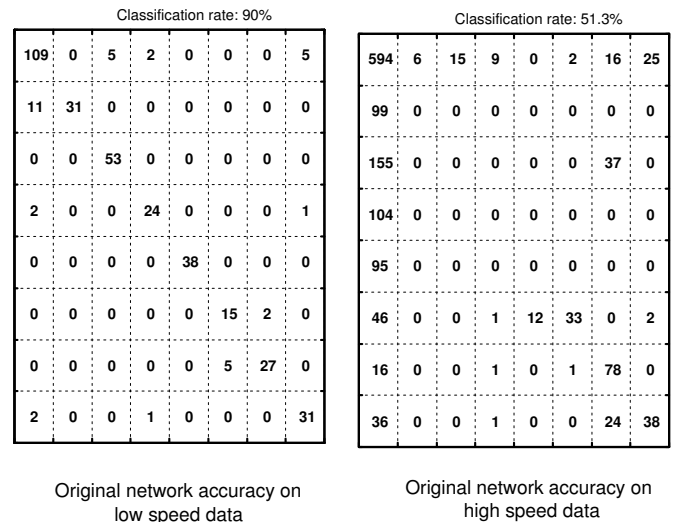


Figure 2: Confusion matrix showing accuracy of the original RBF network. The numbers represent test cases and those lying on the diagonal have been classified accurately, while those off the diagonal have been misclassified. The accuracy of the RBF network trained on low speed data is 90.0%, but when introduced to test data with high speed characteristics, accuracy falls to 51.3%. High speed training data is not available, thus motivating the need to integrate some form of knowledge insertion.

Although the original RBF uses 11 input parameters only three parameters were used in the experimental work. This is because the majority of heuristic rules use only these (rpm1, rpm2 and rpm3) parameters because most faults can be identified by them. In addition, only three machine conditions were considered: unbalance, misalignment and looseness.

4 Defining the Fuzzy System

The knowledge for the fuzzy system was obtained from a domain expert and the available machine fault diagnosis literature. The fuzzy rules, sets and membership functions were manually developed through a process of trial and error.

4.1 Defining the fuzzy sets

A number of criteria were under consideration for the construction of the fuzzy sets.

- Type of membership function (Gaussian, triangular etc).
- Number of membership functions per set.
- Number of rules to required to cover the intended input/output space.
- Defuzzification method used.

The type of membership function to be used proved not to be critical, as the set for the motor speed uses triangular but similar results were achieved for Gaussian. As long as there

is some overlap between the membership functions then the actual type of function does not appear to affect the results to any great extent (this is true for the vibration domain).

The motor speed fuzzy set was used as an input to the inferencing system. Triangular functions were used with the apex of the triangle at the mid-value of the set. The “slow” membership function is present only for completeness as it did not participate in the fuzzy inferencing process. Figure 3 shows the membership functions for the fault severity. This set was used as an input to the system and the values for the membership were determined by the domain experts belief as how much a fault has to be present (percentage) to constitute a problem. In figure 3 the looseness fault can be seen as a major problem for the motor since only a small amount of looseness is required to be classed as a serious fault. Looseness refers to the amount of physical movement experienced by the motor and usually indicates that the mounting bolts are not secure.

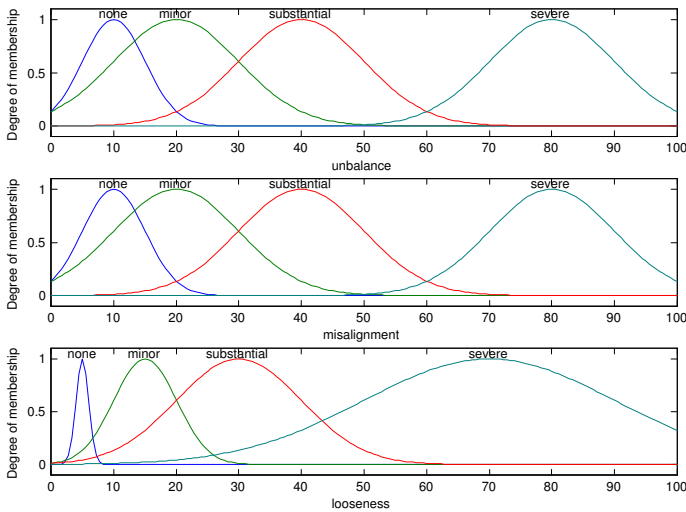


Figure 3: Fuzzy sets defining the fault severity

Figure 4 shows the membership functions for the rpm parameters. These fuzzy sets were used as outputs by the system and the values for the rpm parameters were calculated by the inferencing system. Those values were then used to modify the centre positions of the new RBF hidden units. The membership values were mainly based on averages of both simulated and test-rig high speed data. The diagnostic system used rules based on testing the ratios of the rpm parameters as opposed to absolute values. Some of these rules provided useful information that was incorporated into the fuzzy rules to determine the rpm parameter values.

4.2 Defining the fuzzy rules

Figure 5 shows 10 of the 18 rules for the three fault conditions: unbalance, misalignment and looseness. Rule 19 was present to prevent the fuzzy system from making a decision when no fault conditions were present. Each fuzzy rule consisted of an antecedent and consequent. The antecedent was

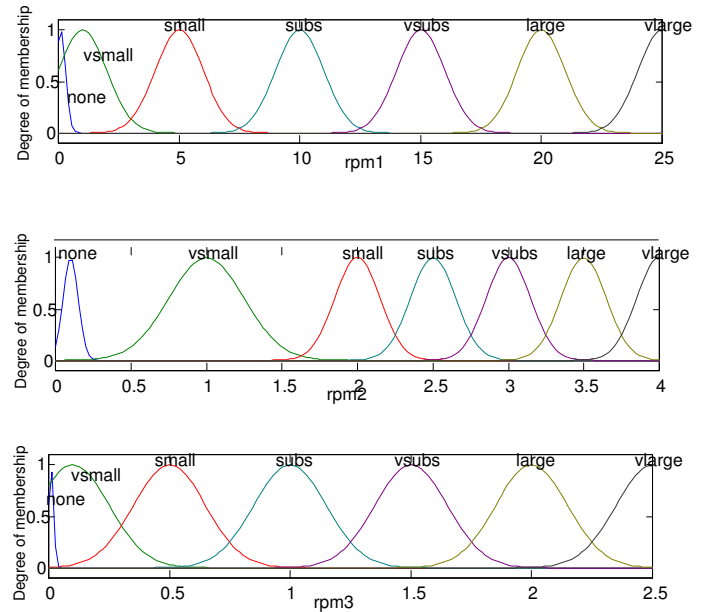


Figure 4: Fuzzy sets defining the parameter modifications

quite simple compared with the rules extracted from the RBF networks as there were only two input features i.e. the motor speed and fault severity. The consequent was more complex as it consisted of values assigned to the three output parameters.

The structure of the rules were almost identical to the reasoning of the domain expert. In addition to comprehensibility there is also the added advantage of system maintainability. Adding further fuzzy rules to cover extra conditions could be easily made but without the problems of “side-effects” that can be encountered in non-fuzzy systems i.e. adding extra rules may cause unanticipated logic errors. Also the coding complexity in non-fuzzy systems can be greatly increased. This was avoided to a great extent with the fuzzy system because a lot of the knowledge was encoded into the fuzzy sets. Therefore, if required the fuzzy sets could be recalibrated/adjusted to take advantage of new situations without recoding the rules.

4.3 Fuzzy Inferencing System

The inferencing method used is the Mamdani method which is the most commonly used method for this type of fuzzy system [Mamdani and Baakini, 1974]. It enabled multiple fuzzy sets to be used as outputs (i.e. in the rule consequents). These fuzzy sets are eventually used to modify the RBF internal parameters (after defuzzification).

The system operates by combining the consequents from each rule through the aggregation operator and returning a defuzzified fuzzy set to provide the system output. The fuzzy Mamdani model is composed of rule sets of IF..THEN type rules [Mitra and Hayashi, 2000].

In order to identify which particular hidden RBF units to

-
- Rule 1: If (MSpeed is fast) and (unbalance is minor)
then (rpm1 is vsmall)(rpm2 is none)(rpm3 is none)
- Rule 2: If (MSpeed is fast) and (unbalance is substantial)
then (rpm1 is small)(rpm2 is vsmall)(rpm3 is vsmall)
- Rule 3: If (MSpeed is fast) and (unbalance is severe)
then (rpm1 is subs)(rpm2 is subs)(rpm3 is subs)
- Rule 4: If (MSpeed is vfast) and (unbalance is minor)
then (rpm1 is vsubs)(rpm2 is vsubs)(rpm3 is vsubs)
- Rule 5: If (MSpeed is vfast) and (unbalance is substantial)
then (rpm1 is large)(rpm2 is large)(rpm3 is large)
- Rule 6: If (MSpeed is vfast) and (unbalance is severe)
then (rpm1 is vlarge)(rpm2 is vlarge)(rpm3 is vlarge)
- Rule 7: If (MSpeed is fast) and (misalignment is minor)
then (rpm1 is small)(rpm2 is vsmall)(rpm3 is none)
- Rule 8: If (MSpeed is fast) and (misalignment is substantial)
then (rpm1 is small)(rpm2 is vsmall)(rpm3 is vsmall)
- Rule 9: If (MSpeed is fast) and (misalignment is severe)
then (rpm1 is subs)(rpm2 is subs)(rpm3 is subs)
- Rule 10: If (MSpeed is vfast) and (misalignment is minor)
then (rpm1 is subs)(rpm2 is subs)(rpm3 is subs)
-

Figure 5: Fuzzy domain rules for machine vibration domain, where: *looseness*, *misalignment*, *unbalance* are the faults that can be encountered by the machines, *minor*, *substantial* *severe*, describe the extent of the fault, *rpm1*, *rpm2*, *rpm3* are the parameters that indicate the presence or absence of the faults, *none*, *small*, *vsml*, *subs* are the amount of change required to the parameters.

modify, the *hREX* algorithm was implemented [McGarry *et al.*, 2001] and was run on the RBF network which produced a list of hidden units contributing to the identification of the three fault classes. Table 1 details the hidden unit allocations for the classes.

Having identified the relevant hidden units, their associated centres, spreads and hidden-to-output unit weights were modified by the fuzzy inferencing system according to the values of the fuzzy sets.

5 Analysis of Knowledge Synthesis Results

The results are shown in figure 6. Overall, an improvement of 25% was made with unbalance gaining 88 correct, misalignment gaining 67 and looseness gaining 30 correct classifications. The original RBF network had an overall tendency to misclassify most examples as “ok” but now the domain rules

Table 1: Identification of hidden units with output classes

Fault Class	hidden units responsive to fault class
Unbalance	7, 9, 12, 13, 15, 23, 28, 31
Misalignment	3, 8, 15
Looseness	13, 14, 16, 21, 32

enabled tighter bounds to be placed around the “ok” category.

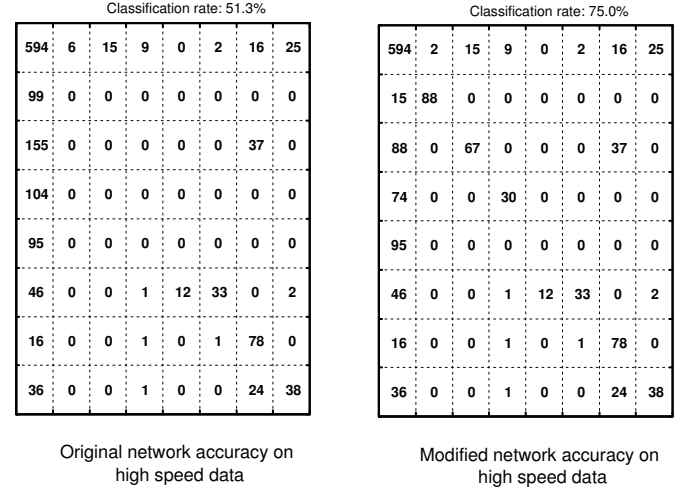


Figure 6: Confusion matrix showing accuracy of the original RBF network compared with synthesized RBF. The numbers represent test cases and those lying on the diagonal have been classified accurately, while those off the diagonal have been misclassified. The original network has an accuracy of 51.3% on the high speed data but when it is modified by inserting domain rules that are characteristic of the nature of high speed data, the accuracy goes up to 75.0%.

The effects of the large number of shared hidden units (33%) within the network are still an open issue. Theoretically, since each shared hidden unit is duplicated and modified for each class there should be no undue interference effects. However, with additional rules these results could be improved upon but obtaining such rules would be difficult as the inter-parameter relationships are complex.

The existing rules cover a relatively simple input to output space mapping that was common to most machine behavior (generic) and was easy to produce from the knowledge acquisition process. It would be possible to conduct a statistical analysis of the high speed data for particular relationships and then to produce rules which could be used on different types of machines i.e. the analysis might have proved that certain relationships exist between low and high speed data on machine “X” and then to have applied these rules to machine “Y”. The difficulty of this approach is that of modelling the specific data too closely and not the underlying relationships (if they exist).

6 Related Work on Knowledge Synthesis

Previous work involving assigning prior knowledge into locally responsive units used a Bayesian framework which enabled the incorporation of an inductive bias [Roscheisen *et al.*, 1994]. Other approaches have considered the problem in terms of inserting symbolic rules into a RBF network [Andrews and Geva, 1996; Tresp *et al.*, 1997].

In the past researchers have attempted to incorporate domain knowledge in the form of fuzzy rules within a neural network. Jin and Sendhoff [Jin and Sendhoff, 1999] use fuzzy rules in the form of hints as described by Abu-Mostafa [Abu-Mostafa, 1993]. The “hint” is expressed as a fuzzy rule that describes qualitatively the function to be learned and is used as a regularization term by the learning algorithm. Narazaki describes the use of fuzzy rules to manipulate the classification boundaries of the hidden units in a MLP network [Narazaki, 1996]. Casalino developed a fuzzy basis function (FBF) network that has some similarity of operation with an RBF network [Casalino *et al.*, 1998]. However, the main conclusion of Casalino work was to identify that the FBF characteristics were closer to a competitive learning model than a feedforward. The work by Kishore [Kishore and Rao, 1997] has the strongest similarities to the work presented in this paper. However, Kishore defined fuzzy sets to represent the step-size of the learning rate for training RBF networks.

7 Conclusions

Knowledge synthesis i.e. the modification of existing RBF networks using heuristic rules has obvious benefits when used in certain situations. The use of knowledge synthesis only makes sense when the available data is insufficient to build a reliable classifier. In such a situation it is advantageous to use heuristic rules to modify an existing RBF network to detect infrequently encountered input vectors that would otherwise be misclassified. However, care must be taken when applying the domain rules. Unless the domain rules can cover a large proportion of the synthesized input to output space it is likely that not all the necessary centre positions will be moved into appropriate locations. This effect will reduce the effectiveness of the new centres. Fortunately, because of the local characteristics of the RBF network the existing centres will be relatively unaffected and should still be able to classify with the same accuracy before knowledge synthesis occurred.

References

- [Abu-Mostafa, 1993] Y. Abu-Mostafa. Hints and the VC dimension. *Neural Computation* 5, pages 278–288, 1993.
- [Andrews and Geva, 1996] R. Andrews and S. Geva. Rules and local function networks. In *Rules and Networks-Proceedings of the Rule Extraction From Trained Artificial Neural Networks Workshop, Artificial Intelligence and Simulation of Behaviour*, Brighton UK, 1996.
- [Casalino *et al.*, 1998] F. Casalino, F. Masulli, and A. Sperduti. Rule specialization in networks of fuzzy basis functions. *Intelligent Automation and Soft Computing*, 4(1):73–82, 1998.
- [Halgamuge, 1997] S. Halgamuge. Self-evolving neural networks for rule-based data processing. *IEEE Transactions on Signal Processing*, 45(11):2766–2773, 1997.
- [Hunt *et al.*, 1996] K. J. Hunt, R. Haas, and R. Murray-Smith. Extending the functional equivalence of radial basis function networks and fuzzy inference systems. *IEEE Transactions on Neural Networks*, 7(3):776–781, May 1996.
- [Jang and Sun, 1993] J. S. R. Jang and C. T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Transactions Neural Networks*, 4(1):156–159, January 1993.
- [Jin and Sendhoff, 1999] Y. Jin and B. Sendhoff. Knowledge incorporation into neural networks from fuzzy rules. *Neural Processing Letters*, 10:231–242, 1999.
- [Kishore and Rao, 1997] A. V. Kishore and M. V. Rao. A novel based algorithm for radial basis function network. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 489, pages 2007–2011, 1997.
- [Kubat, 1998] M. Kubat. Decision trees can initialize radial basis function networks. *IEEE Transactions on Neural Networks*, 9(5):813–821, 1998.
- [Mamdani and Baakini, 1974] E. H. Mamdani and Baakini. Prescriptive method for deriving control policy in a fuzzy-logic controller. *Electronics Letters*, 11(25/26):625–626, 1974.
- [McGarry *et al.*, 2001] K. McGarry, S. Wermter, and J. MacIntyre. Knowledge extraction from local function networks. In *Seventeenth International Joint Conference on Artificial Intelligence*, volume 2, pages 765–770, Seattle, USA, August 4th–10th 2001.
- [Mitra and Hayashi, 2000] S. Mitra and Y. Hayashi. Neuro-fuzzy rule generation: survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3):748–768, 2000.
- [Moody and Darken, 1989] J. Moody and C. J. Darken. Fast learning in networks of locally tuned processing units. *Neural Computation*, pages 281–294, 1989.
- [Narazaki, 1996] H. Narazaki. Reorganizing knowledge in neural networks: An explanatory mechanism for neural networks in data classification problems. *IEEE Transactions on Systems, Man and Cybernetics*, 26(1):107–117, 1996.
- [Roscheisen *et al.*, 1994] M. Roscheisen, R. Hofmann, and V. Tresp. Incorporating prior knowledge into networks of locally-tuned units. In *Computational learning Theory and Natural Learning Systems*, volume III, pages 53–64. 1994.
- [Sun, 1994] C. T. Sun. Rule-base structure identification in an adaptive-network-based fuzzy inference system. *IEEE Transactions on Fuzzy Systems*, 2(1):64–73, 1994.
- [Tresp *et al.*, 1997] V. Tresp, J. Hollatz, and S. Ahmad. Representing probabilistic rules with networks of gaussian basis functions. *Machine Learning*, 27:173–200, 1997.

Unsupervised Dimensionality Estimation and Manifold Learning in high-dimensional Spaces by Tensor Voting

Philippos Mordohai and Gérard Medioni

University of Southern California

email: {mordohai,medioni}@iris.usc.edu

Abstract

We address dimensionality estimation and nonlinear manifold inference starting from point inputs in high dimensional spaces using tensor voting. The proposed method operates locally in neighborhoods and does not involve any global computations. It is based on information propagation among neighboring points implemented as a voting process. Unlike other local approaches for manifold learning, the quantity propagated from one point to another is not a scalar, but is in the form of a tensor that provides considerably richer information. The accumulation of votes at each point provides a reliable estimate of local dimensionality, as well as of the orientation of a potential manifold going through the point. Reliable dimensionality estimation at the point level is a major advantage over competing methods. Moreover, the absence of global operations allows us to process significantly larger datasets. We demonstrate the effectiveness of our method on a variety of challenging datasets.

1 Introduction

A number of algorithms for manifold learning from unorganized points have been recently proposed in the machine learning literature. These include kernel PCA [Schölkopf *et al.*, 1998], locally linear embedding (LLE) [Roweis and Saul, 2000], Isomap [Tenenbaum *et al.*, 2000] and charting [Brand, 2003], that are briefly described in Section 2. They aim at reducing the dimensionality of the input space in a way that preserves some geometric or statistic properties of the data. Isomap, for instance, attempts to preserve the geodesic distances between all points as the manifold is “unfolded” and mapped to a space of lower dimension. A common assumption is that the desired manifold consists of locally linear patches. We relax this assumption by only requiring that manifolds be smooth almost everywhere. Smoothness in this context is the property of the manifold’s orientation to vary gradually as one moves from point to point. This property is encoded in the votes that each point casts to its neighbors.

The representation of each point is in the form of a second order, symmetric, nonnegative definite tensor whose eigenvalues and eigenvectors fully describe the dimensionality and

orientation of each point [Medioni *et al.*, 2000]. The votes are also tensors and their accumulation is a process that is considerably more powerful than the accumulation of scalars or vectors in terms of both the types of structure that can be described, as well as its robustness to noise. A point casts a vote to a neighboring point that carries an estimate of the orientation of the receiver given the orientation of the voter. Even in the case of unoriented inputs, meaningful votes can be cast as a function of the relative position of the two points. The result of voting is a new tensor at each point that encompasses the weighted contribution of the point’s neighbors.

The dimensionality d of the manifold at each point in a D -dimensional space is indicated by the maximum gap in the eigenvalues of the tensor, while the top d eigenvectors span the normal space of the manifold. The first property makes dimensionality estimation a procedure with no parameters that require tuning. These estimates are reliable without averaging, as shown in Section 4, and allow us to handle data with varying dimensionality. This is arguably one of the most important contributions of our approach. Moreover, the presence of outliers, boundaries, intersections or multiple manifolds pose no additional difficulties, due to the unsupervised learning ability of our method. Furthermore, non-manifolds, such as hyper-spheres, can be inferred since we do not attempt to estimate a global “unfolding”.

We do not attempt to estimate a mapping from the input space to the embedding space, as in [Brand, 2003] [Teh and Roweis, 2003]. Even though certain advantages can be derived from such a mapping, it is not always feasible, as in the case of hyper-spheres or multiple manifolds, and it is not necessary for many tasks, including interpolation. Valid paths on the manifold can be followed using the estimated tangent subspaces at the input points by projecting the desired direction on them and moving on the projection.

Since all operations are local, time complexity is $O(Dn \log n)$ where n is the number of points, enabling us to handle datasets with orders of magnitude more points than the current state of the art without incurring unreasonable computational cost. In terms of storage, the requirements at each point are $O(D^2)$. Our current implementation is probably limited to D in the order of a few hundreds. As all manifold learning methods, we operate under the assumption that the data are randomly sampled from the manifold in a way that is close to uniform, or at least does not favor certain parts or

directions of the manifold.

The paper is organized as follows: the next section briefly reviews related work; Section 3 describes the tensor voting framework; Section 4 contains experimental results on classic datasets; finally, Section 5 concludes the paper and discusses the directions of our future work.

2 Related Work

In this section we briefly present recent approaches for learning low dimensional embeddings from points in high dimensional spaces. Most of them are extensions of linear techniques, such as Principal Component Analysis (PCA) and Multi-Dimensional Scaling (MDS), based on the assumption that nonlinear manifolds can be approximated by locally linear patches. In contrast to other methods, Schölkopf *et al.* [1998] propose kernel PCA that attempts to find linear patches using PCA in a space of typically higher dimensionality than the input space. Correct kernel selection can reveal the low dimensional structure of the input data. For instance a second order polynomial kernel can “flatten” quadratic manifolds.

Roweis and Saul [2000] address the problem by Locally Linear Embedding (LLE). Processing involves computing reconstruction weights for each point from its neighbors. The same weights should also reconstruct the point from its neighbors in the low dimensional embedding space, since the neighborhoods are assumed to be linear. The dimensionality of the embedding however has to be given as a parameter since it cannot always be estimated from the data [Saul and Roweis, 2003]. LLE assures that local neighborhood structure is preserved during dimensionality reduction.

Isomap Tenenbaum *et al.* [2000] approximates geodesic distances between points by graph distances. Then, MDS is applied on the geodesic instead of Euclidean distances to compute an embedding that forces nearby points to remain nearby and faraway points to remain that way. A variation of Isomap that can be applied to data with intrinsic curvature, but known distribution, and a faster alternative that only uses a few landmark point for distance computations have been proposed in [de Silva and Tenenbaum, 2003]. Isomap and its variants are limited to convex datasets.

Belkin and Niyogi [2003] compute the graph Laplacian of the adjacency graph of the input data, which is an approximation of the Laplace-Beltrami operator on the manifold, and exploit its locality preserving properties that were first observed in the field of clustering. The Laplacian eigenmaps algorithm can be viewed as a generalization of LLE, since the two become identical when the weights of the graph are chosen according to the criteria of the latter. The dimensionality of the manifold however cannot be reliably estimated from the data, as in most of the work reviewed here. Donoho and Grimes [2003] propose a similar scheme which computes the Hessian instead of the Laplacian. The authors claim that the Hessian is better suited for detecting linear patches on the manifold. The major contribution of this approach is that it proposes a global method, which, unlike Isomap, can be applied to non-convex datasets.

Weinberger and Saul [2004] address the problem of man-

ifold learning by enforcing local isometry. The lengths of the sides of triangles of neighboring points are preserved during the embedding. These constraints can be expressed in terms of pairwise distances and the optimal embedding can be found by semi-definite programming. The method is more computationally demanding than LLE and Isomap, but can reliably estimate the underlying dimensionality of the inputs by locating the largest gap between the eigenvalues of the Gram matrix of the outputs. Similarly to our approach, this estimate does not require a threshold, as do approaches that estimate the residual error as a function of the dimensionality of the fitted model.

An algorithm that computes a mapping, and not just an embedding, of the data is described in [Brand, 2003]. The intrinsic dimensionality of the manifold is estimated by examining the rate of growth of the number of points contained in hyper-spheres as a function of the radius. Linear patches, areas of curvature and noise can be discriminated using the proposed measure. Affine transformations that align the coordinate systems of the linear patches are computed at the second stage. This defines a global coordinate system for the embedding and thus a mapping between the input space and the embedding space.

Finally, we briefly review approaches that estimate intrinsic dimensionality without any attempts to learn local structure. One such approach was presented in [Bruske and Sommer, 1998]. An optimally topology preserving map (OTPM) is constructed for a subset of the data, which is produced after vector quantization. Then PCA is performed for each node of the OTPM under the assumption that the data are locally linear. The average of the number of significant singular values at the nodes is the estimate of the intrinsic dimensionality. Kégl [2005] estimates the capacity dimension of the manifold, which does not depend on the distribution of the data and is equal to the topological dimension. An efficient approximation based on packing numbers is proposed. The algorithm takes into account dimensionality variations with scale and is based on a geometric property of the data, rather than successive projections to increasingly higher-dimensional subspaces until a certain percentage of the data is explained. Costa and Hero [2004] estimate the intrinsic dimension of manifold and the entropy of the samples using geodesic-minimal-spanning trees. The method is similar to Isomap and thus produces a single global estimate. Levina and Bickel [2005] compute maximum likelihood estimates of dimensionality by examining the number of neighbors included in spheres whose radius is selected in such a way that the density of the data can be assumed constant, and enough neighbors are included. These requirements cause an underestimation of the dimensionality when it is very high.

The difference between our approach and those of [Bruske and Sommer, 1998][Brand, 2003] [Kégl, 2005] [Weinberger and Saul, 2004] [Costa and Hero, 2004] and [Levina and Bickel, 2005] is that it produces reliable dimensionality estimates at the point level, which do not have to be averaged over the entire dataset. We are also able to estimate the local orientation of the manifold without being restricted to simple linear models.

3 Tensor Voting

In this section we first review the tensor voting framework [Medioni *et al.*, 2000], which can infer structures from sparse and noisy data via a local voting process. Results have been published mostly in 2- and 3-D domains, but the framework can be extended to higher dimensions, as in [Tang *et al.*, 2001]. We begin by describing the basics of the framework: data representation and voting. Then, we present the contribution of this paper to the methodology, which is an efficient implementation, in terms of both space and time, of tensor voting in high-dimensional spaces. We are able to operate in spaces where that seemed impractical or impossible based on the formulation of [Medioni *et al.*, 2000].

3.1 Data Representation

The representation at each point is in the form of a second order, symmetric, nonnegative definite tensor. The tensor can also be viewed as a matrix or an ellipsoid. It represents the **normal** space at the point. For instance, a hyper-plane has one normal $\vec{n}$, which can be encoded in tensor form as $\vec{n}\vec{n}^T$. A structure with a normal space of rank d has d normals and is represented by a tensor of the form:

$$T = \sum_d \vec{n}_d \vec{n}_d^T \quad (1)$$

A point without orientation information can be equivalently viewed as one having all possible normals and is encoded as the identity matrix. A tensor in this form represents an equal preference for all orientations and is called a “ball tensor”, since the corresponding ellipsoid is a sphere or hyper-sphere. On the other hand, a tensor that contains only one orientation is called a “stick tensor”. Stick tensors represent the hyper-plane of a D -dimensional space, which has one normal and $D - 1$ tangents.

Depending on the type of structure the point belongs to, it has a different number of normals and tangents. The tensor’s eigensystem encodes this information. Eigenvectors that belong to the tangent space correspond to zero eigenvalues, while those that belong to the normal space correspond to nonzero eigenvalues (typically equal to 1). Therefore, points with known orientation can be encoded in this representation by appropriately constructed tensors, as in (1). Points with unknown orientation are represented by identity matrices (ball tensors) See Fig. 1(a) for example tensors in 3-D.

On the other hand, given a second order, symmetric, non-negative definite tensor, the type of structure encoded in it can be found by examining its eigensystem. Any tensor that has these properties can be decomposed as in the following equation:

$$\begin{aligned} T &= \sum_{d=1}^D \lambda_d \hat{e}_d \hat{e}_d^T = \\ &= (\lambda_1 - \lambda_2) \hat{e}_1 \hat{e}_1^T + (\lambda_2 - \lambda_3) (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T) \\ &\quad + \dots + \lambda_D (\hat{e}_1 \hat{e}_1^T + \hat{e}_2 \hat{e}_2^T + \dots + \hat{e}_D \hat{e}_D^T) = \\ &\sum_{d=1}^{D-1} [(\lambda_d - \lambda_{d+1}) \sum_{k=1}^d \hat{e}_k \hat{e}_k^T] + \lambda_D (\hat{e}_1 \hat{e}_1^T + \dots + \hat{e}_D \hat{e}_D^T) \end{aligned} \quad (2)$$

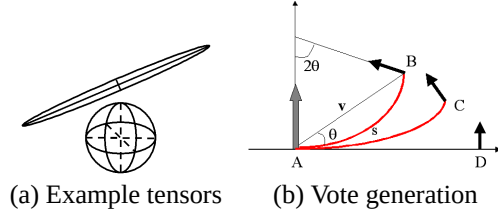


Figure 1: Tensor Voting. (a) The shape of the tensor indicates the type of structure at the location, while its size indicates the confidence of this information. The top tensor has a strong preference of orientation and has higher confidence than the bottom one, which has no preference for any orientation. (b) Vote generation as a function of the relative position of a stick voter and the receiver and the orientation of the voter.

where λ_d are the eigenvalues in descending order and $\hat{e}_d$ are the corresponding eigenvectors. The tensor simultaneously encodes *all* possible types of structure. The confidence we have in the type that has d normals is encoded in the difference $\lambda_d - \lambda_{d+1}$.

3.2 The Voting Process

The tensor voting framework was designed to enforce constraints, such as proximity, co-linearity and co-curvilinearity with human perception in 2- and 3-D in mind. These constraints still hold in high dimensional spaces as long as one is looking for structures that are smooth almost everywhere. In this section we describe how information is propagated from one point, the voter, to another, the receiver.

During the voting process, each point casts votes to each of its neighboring points. The votes are also second-order, symmetric, nonnegative definite tensors. They encode the orientation the receiver would have, if the voter and receiver were in the same structure. We begin by examining the case of a voter that is associated with a stick tensor of unit length. The magnitude of a vote cast by the stick tensor decays with respect to the length of the smooth circular path connecting the voter and receiver. The circle was selected since it maintains constant curvature. It degenerates to a straight line if the vector connecting the voter and receiver is orthogonal to the normal of the voter. The orientation of the vote is towards the center of the circle defined by the two points and the orientation of the voter. The vote is generated according to the following equation and then transformed to the appropriate coordinate system.

$$\begin{aligned} S(s, l, \theta) &= e^{-\left(\frac{s^2 + c\kappa^2}{\sigma^2}\right)} \begin{bmatrix} -\sin(2\theta) \\ \cos(2\theta) \end{bmatrix} \begin{bmatrix} -\sin(2\theta) & \cos(2\theta) \end{bmatrix} \\ s &= \frac{\theta \|\vec{v}\|}{\sin(\theta)}, \quad \kappa = \frac{2\sin(\theta)}{\|\vec{v}\|} \end{aligned} \quad (3)$$

s is the length of the arc between the voter and receiver, and κ is its curvature (see Fig. 1(b)), σ is the scale of voting, and c is a constant. No vote is generated if the angle θ is more than 45° . Also, the field is truncated to the extent where the magnitude of the vote is more than 3% of the magnitude of the voter.

Since tensor voting is a function of the position of the voter and the receiver and the orientation of the voter, vote generation by a stick tensor occurs in a 2-D subspace defined by the vector $\vec{v}$ that connects the two points and the normal of the voter. Therefore, stick voting is fully defined by Fig. 1(b) and Eq. 3, regardless of the dimension of the input space. According to [Medioni *et al.*, 2000], the votes cast by other types of tensors were pre-computed and stored in look-up tables, called voting fields. The presence of a normal space of rank more than one is simulated by a rotating stick tensor that spans the space. Since no closed form solution exists, the integration is numerically approximated. This process becomes impractical as the dimensionality of the space increases. Space requirements for storing D D -dimensional voting fields with k samples per axis are $O(Dk^D)$. At the same time, the advantage of re-using pre-computed votes vanishes as the dimensionality increases. We propose instead a novel, efficient way to directly compute votes. Since integration is infeasible, we seek an approximate solution.

3.3 Efficient Voting Scheme

The new scheme, proposed here, generates votes that contain the orientation information that should be communicated to the receiver, but not the uncertainty. The latter, in the original framework, was conveyed by the ball component of the vote that was due to the integration of votes cast by the rotating stick tensor. For instance, according to [Medioni *et al.*, 2000], a voting tensor that encodes a curve in 3-D (that has two normals and a tangent) casts a vote that has a dominant curve component and also some uncertainty. Here, we propose to propagate only the curve orientation at the receiver given the orientation of the voter. The uncertainty in the new formulation comes only from the accumulation of votes that are not perfectly aligned.

Let $\vec{v}$ be the vector connecting the voting and receiving points. It can be decomposed into $\vec{v}_t$ in the tangent space of the voter (null in the case of a ball voter) and $\vec{v}_n$ in the normal space. The new vote generation process is based on the observation that curvature in Eq. 3 is not a factor when θ is zero, or in other words, if the voting stick is orthogonal

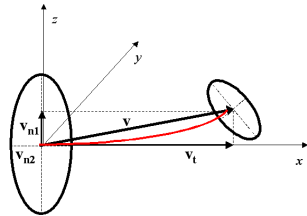


Figure 2: New scheme for vote generation. The voter here is a tensor with two normals in 3-D. The vector connecting the voter and receiver is decomposed into $\vec{v}_n$ and $\vec{v}_t$ that lie in the normal and tangent space of the voter respectively. A new basis that includes $\vec{v}_n$ is defined for the normal space and each basis component casts a stick vote. Only the vote generated by the orientation parallel to $\vec{v}_n$ is not parallel to the normal space. Tensor addition of the stick votes produces the combined vote.

to $\vec{v}_n$. We can exploit this by defining a new basis for the normal space of the voter that includes $\vec{v}_n$. Then, the vote is constructed as the tensor addition of the votes cast by stick tensors parallel to the new basis vectors. Among those votes, only the one generated by the stick tensor parallel to $\vec{v}_n$ is not parallel to the normal space of the voter. See Fig. 2 for an illustration in 3-D. This scheme is applicable without changes in the case of ball voters. Since two unoriented points define a straight line in any space, the vote from a ball tensor should have $D - 1$ normals and one tangent. Since $\vec{v}_t = \vec{0}$, the tensor parallel to $\vec{v}_n$ does not cast a vote due to the 45° cut-off rule. The remaining $D - 1$ tensors in the new basis cast votes that are equal in magnitude and span the space orthogonal to $\vec{v}$, and thus represent a straight line connecting the two points.

Tensors with unequal eigenvalues are decomposed before voting according to Eq. 2. Then, each component votes separately and the vote is weighted by $\lambda_d - \lambda_{d+1}$, except the ball component whose vote is weighted by λ_D . This implementation of tensor voting is more efficient in terms of space, since there is no need to store voting fields, and in terms of time since we have devised a direct method for computing votes that replaces the numerical integration of [Medioni *et al.*, 2000]. The new computation requires at most D computations of a stick vote according to Eq. 3, followed by the addition of all the votes, instead of integration over D dimensions. Experiments in 2- and 3-D, that cannot be included here due to space limitations, validate the accuracy of the new vote generation scheme.

3.4 Vote Analysis

Votes are accumulated at each point by tensor addition, which is equivalent to the addition of matrices. After voting is completed, the eigensystem of the new tensor is analyzed and the tensor is decomposed as in Eq. 2. In 3-D, the inference of the type of structure that a point belongs is accomplished as follows. The difference between the two largest eigenvalues encodes surface confidence, with a surface normal given by $\vec{e}_1$. The difference between the second and third eigenvalue encodes curve confidence, with a the normals to the curve being $\vec{e}_1$ and $\vec{e}_2$. The smallest eigenvalue encodes junction confidence. Outliers receive little and inconsistent support from their neighborhood and can be identified by their low eigenvalues. Generalization to higher dimensions is straightforward, considering that there are D manifold types in D dimensions. The dimensionality d is estimated by finding the maximum difference $\lambda_d - \lambda_{d+1}$.

4 Experimental Results

In this section we present experimental results in dimensionality estimation and local structure inference. All inputs consist of un-oriented points and are encoded as ball tensors.

Swiss Roll The first experiment is on the “Swiss Roll” dataset, which is available at <http://isomap.stanford.edu/>. It contains 20,000 points on a 2-D manifold in 3-D (Fig. 3). We perform a simple evaluation of the quality of the orientation estimates by projecting the nearest neighbors of each



Figure 3: The “Swiss Roll” dataset in 3-D

point on the estimated tangent space and measuring the percentage of the distance that has been recovered. The accuracy of dimensionality estimation at each point, the percentage of recovered distances for the 8 nearest neighbors and execution times on a Pentium 4 at 2.8 GHz, as a function of σ , can be seen in Table 1. Performance is the same at boundaries. The number of votes cast by each point ranges from 187 for $\sigma^2 = 50$ to 5440 for $\sigma^2 = 1000$. The accuracy is high for a large range of σ .

σ^2	Correct Dim. (%)	Dist. Rec. (%)	Time (sec)
50	93.10	91.75	11
100	99.24	93.07	22
200	99.91	93.21	50
300	99.96	93.20	81
500	99.95	93.18	144
700	99.88	93.13	202
1000	99.67	93.02	307

Table 1: Rate of correct dimensionality estimation and execution times as functions of σ for the “Swiss Roll” dataset

Structures with varying dimensionality The second dataset is in 4-D and contains points sampled from three structures: a line, a 2-D cone and a 3-D hyper-sphere. The hyper-sphere is a structure with three degrees of freedom. It cannot be unfolded unless we remove a small part from it. Figure 4(a) shows the first three dimensions of the data. The dataset contains a total 135,864 points, voting is performed with $\sigma^2 = 200$ and takes 2 hours and 7 minutes. Figures 4(c-d) show the points classified according to their dimensionality. Performing the same analysis as above for the accuracy of the tangent space estimation, 91.04% of the distances of the 8 nearest neighbors of each point lie on the tangent space, even though both the cone and the hyper-sphere have intrinsic curvature and cannot be accurately approximated by linear models. All the methods presented in Sec. 2 would fail for this dataset because of the presence of structures with different dimensionalities and because the hyper-sphere cannot be unfolded.

Data in high dimensions The datasets for this experiment were generated by sampling a few thousand points with low intrinsic dimensionality (3 or 4) and mapping them to a medium dimensional space (14- to 16-D) using linear and quadratic functions. The generated points were then rotated

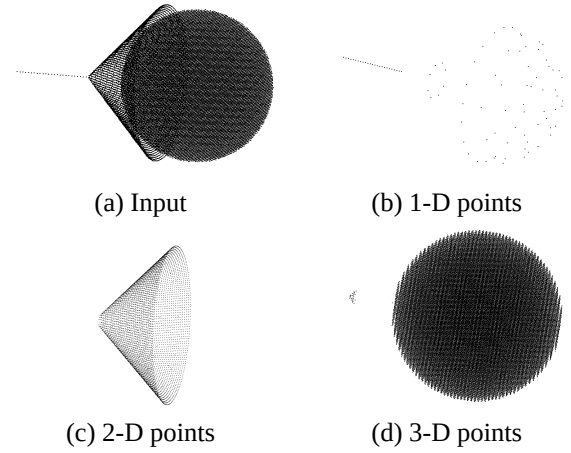


Figure 4: Data of varying dimensionality in 4-D. The first three axes of the input and the classified points are shown.

and embedded in a 50- to 150-D space while uniform noise was added to all dimensions. The accuracy of dimensionality estimation after tensor voting can be seen in Table 2.

Intrinsic Dim.	Linear Mappings	Quadratic Mappings	Space Dim.	Dim. Est. (%)
4	10	6	50	93.6
3	8	6	100	97.4
4	10	6	100	93.9
3	8	6	150	97.3

Table 2: Rate of correct dimensionality estimation for high dimensional data

5 Conclusions

We have presented an approach to manifold learning that offers certain advantages over the state-of-the-art. First, we are able to obtain accurate estimates of the intrinsic dimensionality at the point level. Moreover, since the dimensionality is found as the maximum gap in the eigenvalues of the tensor at the point, no thresholds are needed. In most other approaches the dimensionality has to be provided, or, at best, an average intrinsic dimensionality is estimated for the entire dataset, as in [Bruske and Sommer, 1998] [Brand, 2003] [Kégl, 2005] [Weinberger and Saul, 2004] [Costa and Hero, 2004]. Second, even though tensor voting on the surface looks like a local covariance estimation, the fact that the votes are tensors and not scalars reveals more information for each point. The properties of the tensor representation, which can handle the simultaneous presence of multiple orientations, allow the reliable inference of the normal and tangent space at each point. Due to its unsupervised nature, tensor voting is very robust against outliers, as demonstrated in [Medioni *et al.*, 2000] for the 2- and 3-D case. This property holds in higher dimensions, where random noise is even more scattered. Lack of space did not allow us to include outlier rejection results here.

An important advantage of tensor voting is the absence of global computations, which makes time complexity $O(Dn \log n)$. This enables us to process datasets with very large number of points. Computation time scales linearly with the number of points assuming that more points are added to the dataset in a way that the density remains constant. In this case, the number of votes cast per point remains constant and time requirements grow linearly. Complexity is adversely affected by the dimensionality of the space D , since eigen-decompositions of $D \times D$ tensors have to be performed. For most practical purposes, the number of points has to be considerably larger than the dimensionality of the space ($n \gg D$) to allow structure inference. Computational complexity, therefore, is reasonable with respect to the largest parameter.

It should also be noted that the votes attenuate with distance and curvature. This is a more intuitive formulation than using the N nearest neighbors with equal weights, since some of them may be too far or belong to a different part of the structure. For both the ϵ distance and the N nearest neighbor formulations, however, the distance metric in the input space has to be meaningful. This is also the case for all the methods presented in Sec. 2.

The representation by tensors allows us to deal with structures that are not necessarily manifolds. These include non-manifolds, such as hyper-spheres, intersecting structures and datasets containing structures of different dimensionality. A mapping from the input to the embedding space cannot be computed in any of these cases, but it is not necessary for many tasks including interpolation, function approximation and local orientation (gradient) estimation. One can also view our algorithm as a preprocessing stage that facilitates further processing. Its outputs can be used as inputs for tasks such as dimensionality reduction and mapping to a low dimensional space, if desired.

Two issues require further investigation. The first is a more thorough analysis of the required properties of the data distribution. In line with other methods for manifold learning ([Roweis and Saul, 2000][Tenenbaum *et al.*, 2000][Brand, 2003]), we assume that the data are distributed regularly in space. A uniform distribution is ideal but not necessary, as shown in Sec. 4, especially in the last experiment where the points are mapped to the high dimensional space nonlinearly. The second issue is that of data sufficiency for structure inference. As the dimensionality of the space and the intrinsic dimensionality of the structure grow, the number of points necessary to infer the latter correctly also grows. We intend to investigate the minimum requirements in terms of the number of points for inferring a certain type of structure in a space of high dimensions.

Acknowledgement

This research has been supported by the U.S. National Science Foundation grant IIS 03 29247.

References

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Com-*

putation, 15(6):1373–1396, 2003.

- M. Brand. Charting a manifold. In *Advances in Neural Information Processing Systems 15*, pages 961–968. MIT Press, Cambridge, MA, 2003.
- J. Bruske and G. Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(5):572–575, May 1998.
- J. Costa and A.O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *IEEE Trans. on Signal Process.*, 52(8):2210–2221, Aug. 2004.
- V. de Silva and J.B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, Cambridge, MA, 2003.
- D. Donoho and C. Grimes. Hessian eigenmaps: new tools for nonlinear dimensionality reduction. In *Proceedings of National Academy of Science*, pages 5591–5596, 2003.
- B. Kégl. Intrinsic dimension estimation using packing numbers. In *Advances in Neural Information Processing Systems 15*, pages 681–688. MIT Press, Cambridge, MA, 2005.
- E. Levina and P. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in Neural Information Processing Systems 17*, pages 777–784. MIT Press, Cambridge, MA, 2005.
- G. Medioni, M.S. Lee, and C.K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, New York, 2000.
- S.T. Roweis and L.K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- L. K. Saul and S. T. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.
- B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.
- C.K. Tang, G. Medioni, and M.S. Lee. N-dimensional tensor voting and application to epipolar geometry estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(8):829–844, August 2001.
- Y.W. Teh and S. Roweis. Automatic alignment of local representations. In *Advances in Neural Information Processing Systems 15*, pages 841–848, Cambridge, MA, 2003. MIT Press.
- J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.
- K.Q. Weinberger and L.K. Saul. Unsupervised learning of image manifolds by semidefinite programming. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, pages II: 988–995, 2004.

Generalization Error of Linear Neural Networks in an Empirical Bayes Approach

Shinichi Nakajima^{†‡} and Sumio Watanabe[†]

[†] Tokyo Institute of Technology

Mailbox R2-5, 4259 Nagatsuda, Midori-ku, Yokohama, Kanagawa, 226-8503 Japan

nakajima.s@cs.pi.titech.ac.jp, swatanab@pi.titech.ac.jp

[‡] Nikon Corporation

201-9 Oaza-Miizugahara, Kumagaya, Saitama, 360-8559 Japan

Abstract

It is well known that in unidentifiable models, the Bayes estimation has the advantage of generalization performance to the maximum likelihood estimation. However, accurate approximation of the posterior distribution requires huge computational costs. In this paper, we consider an empirical Bayes approach where a part of the parameters are regarded as hyperparameters, which we call a subspace Bayes approach, and theoretically analyze the generalization error of three-layer linear neural networks. We show that a subspace Bayes approach is asymptotically equivalent to a positive-part James-Stein type shrinkage estimation, and behaves similarly to the Bayes estimation in typical cases.

1 Introduction

Unidentifiable parametric models, such as neural networks, mixture models, and so on, have a wide range of applications. These models have singularities in the parameter space, hence the conventional learning theory of the regular statistical models does not hold. Recently, generalization performance of some unidentifiable models has been theoretically clarified. In the maximum likelihood (ML) estimation, which is asymptotically equivalent to the maximum a posterior (MAP) estimation, the generalization error of linear neural networks was proved to be greater than that of the regular models whose dimension of the parameter space is the same when the model is redundant to learn the true distribution [Fukumizu, 1999]. On the other hand, in the Bayes estimation, the generalization error of neural networks, linear neural networks, mixture models, and so on was proved to be less than that of the regular models [Watanabe, 2001; Aoyagi and Watanabe, 2004; Yamazaki and Watanabe, 2003].

However, the Bayes posterior distribution can seldom be exactly realized. Furthermore, Markov chain Monte Carlo (MCMC) methods, often used for approximation of the posterior distribution, require huge computational costs. As an alternative, the variational Bayes approach, where the correlation between parameters and the other parameters, or the correlation between the parameters and the hidden variables

is neglected, was proposed [Hinton and van Camp, 1993; MacKay, 1995; Attias, 1999; Ghahramani and Beal, 2000].¹

In this paper, we consider another alternative, which we call a subspace Bayes (SB) approach. An SB approach is an empirical Bayes (EB) approach where a part of the parameters of a model are regarded as hyperparameters. If we regard the parameters of one layer as hyperparameters, we can analytically calculate the marginal likelihood in some three-layer models. Consequently, what we have to do is only to find the hyperparameter value maximizing the marginal likelihood. The computational costs of an SB approach is thus much less than that of posterior distribution approximation by MCMC methods. At first in this paper, we prove that in three-layer linear neural networks, an SB approach is equivalent to a positive-part James-Stein (JS) type shrinkage estimation [James and Stein, 1961]. Then, we clarify its generalization error, also considering *delicate* situations, the most important situations in model selection problems and in statistical tests, when the Kullback-Leibler divergence of the true distribution from the singularities is comparable to the inverse of the number of training samples.² We conclude that an SB approach provides as good performance as the Bayes estimation in typical cases.

In Section 2, neural networks and linear neural networks are briefly introduced. The framework of the Bayes estimation, that of an EB approach, and that of an SB approach are described in Section 3. The significance of singularities for generalization performance and the importance of analysis of *delicate* situations are explained in Section 4. The SB solution and its generalization error are derived in Section 5. Discussions and conclusions follow in Section 6 and in Section 7, respectively.

2 Linear Neural Networks

Let $x \in \mathbb{R}^M$ be an input (column) vector, $y \in \mathbb{R}^N$ an output vector, and w a parameter vector. A neural network model can be described as a parametric family of maps $\{f(\cdot; w) : \mathbb{R}^M \mapsto \mathbb{R}^N\}$. A three-layer neural network with H hidden

¹We have just derived the variational Bayes solution of linear neural networks and clarified its generalization error and training error in [Nakajima and Watanabe, 2005b].

²We have also clarified the training error, which is put into [Nakajima and Watanabe, 2005a].

units is defined by

$$f(x; w) = \sum_{h=1}^H b_h \psi(a_h^t x), \quad (1)$$

where $w = \{(a_h, b_h) \in \mathbb{R}^M \times \mathbb{R}^N; h = 1, \dots, H\}$ summarizes all the parameters, $\psi(\cdot)$ is an activation function, which is usually a bounded, non-decreasing, antisymmetric, nonlinear function like $\tanh(\cdot)$, and t denotes the transpose of a matrix or vector. Assume that the output is observed with a noise subject to $\mathcal{N}_N(0, \sigma^2 I_N)$, where $\mathcal{N}_d(\mu, \Sigma)$ denotes the d -dimensional normal distribution with average vector μ and covariance matrix Σ , and I_d denotes the $d \times d$ identity matrix. Then, the conditional distribution is given by

$$p(y|x, w) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{\|y - f(x; w)\|^2}{2\sigma^2}\right). \quad (2)$$

In this paper, we focus on linear neural networks, whose activation function is linear, as the simplest multilayer models.³ A linear neural network model (LNN) is defined by

$$f(x; A, B) = B A x, \quad (3)$$

where $A = (a_1, \dots, a_H)^t$ is an $H \times M$ input parameter matrix and $B = (b_1, \dots, b_H)$ is an $N \times H$ output parameter matrix. Because the transform $(A, B) \mapsto (TA, BT^{-1})$ does not change the map for any non-singular $H \times H$ matrix T , the parameterization in Eq.(3) has trivial redundancy. Accordingly, the essential dimension of the parameter space is given by

$$K = H(M + N) - H^2. \quad (4)$$

We assume that $H \leq N \leq M$ throughout this paper.

3 Framework of Learning Methods

3.1 Bayes Estimation

Let $X^n = \{x_1, \dots, x_n\}$ and $Y^n = \{y_1, \dots, y_n\}$ be arbitrary n training samples independently and identically taken from the true distribution $q(x, y) = q(x)q(y|x)$. The marginal conditional likelihood of a model $p(y|x, w)$ is given by

$$Z(Y^n|X^n) = \int \phi(w) \prod_{i=1}^n p(y_i|x_i, w) dw, \quad (5)$$

where $\phi(w)$ is the prior distribution. The posterior distribution is given by

$$p(w|X^n, Y^n) = \frac{\phi(w) \prod_{i=1}^n p(y_i|x_i, w)}{Z(Y^n|X^n)}, \quad (6)$$

and the predictive distribution is defined as the average of the model over the posterior distribution as follows:

$$p(y|x, X^n, Y^n) = \int p(y|x, w) p(w|X^n, Y^n) dw. \quad (7)$$

The generalization error, a criterion of generalization performance, is defined by

$$G(n) = \langle G(X^n, Y^n) \rangle_{q(X^n, Y^n)}, \quad (8)$$

³A linear neural network model is not a toy but an useful model, known as a reduced-rank regression model, in many applications [Reinsel and Velu, 1998].

where

$$G(X^n, Y^n) = \int q(x) q(y|x) \log \frac{q(y|x)}{p(y|x, X^n, Y^n)} dx dy \quad (9)$$

is the Kullback-Leibler (KL) divergence of the predictive distribution from the true distribution, and $\langle \cdot \rangle_{q(X^n, Y^n)}$ denotes the expectation value over all sets of n training samples.

3.2 Empirical Bayes Approach and Subspace Bayes Approach

We often have little information about the prior distribution, with which an EB approach was originally proposed to cope. We can introduce hyperparameters in the prior distribution; for example, when we use a prior distribution that depends on a hyperparameter τ_1 such as

$$\phi(w) = \frac{1}{(2\pi\tau_1^2)^{K/2}} \exp\left(-\frac{\|w\|^2}{2\tau_1^2}\right), \quad (10)$$

the marginal likelihood, Eq.(5), also depends on τ_1 . In an EB approach, τ_1 is estimated by maximizing the marginal likelihood or by a slightly different way [Efron and Morris, 1973; Akaike, 1980; Kass and Steffey, 1989]. Extending the idea above, we can introduce hyperparameters also in a model distribution. What we call an SB approach is an EB approach where a part of the parameters of a model are regarded as hyperparameters. In the following sections, we analyze two versions of SB approach: in the first one, we regard the output parameter matrix B of the map, Eq.(3), as a hyperparameter and then marginalize the likelihood in the input parameter space (MIP); and in the other one, we regard the input parameter matrix A , instead of B , as a hyperparameter and then marginalize in the output parameter space (MOP).

4 Unidentifiability and Singularities

We say that a parametric model is unidentifiable if the map from the parameter to the probability distribution is not one-to-one. A neural network model, Eq.(1), is unidentifiable because the model is independent of a_h when $b_h = 0$, or vice versa. The continuous points denoting the same distribution are called the singularities, because the Fisher information matrix on them degenerates. When the true model is not on the singularities, asymptotically they do not affect prediction, and therefore, the conventional learning theory of the regular models holds. On the other hand, when the true model is on the singularities, they significantly affect generalization performance as follows: in the ML estimation, the extent of the set of the points denoting the true distribution increases its neighborhoods and hence the flexibility of imitating noises, and therefore, accelerates overfitting; while in the Bayes estimation, the large entropy of the singularities increases the weights of the distributions near the true one, and therefore, suppresses overfitting. In LNNs, the former property appears as acceleration of overfitting by selection of the largest singular value components of a random matrix, and in the SB approaches of LNNs, the latter property appears as James-Stein type shrinkage, as shown in the following sections.

Suppression of overfitting accompanies insensitivity to the true components with small amplitude. There is a trade-off,

which would, however, be ignored in asymptotic analysis if we would consider only situations when the true model is *distinctly* on the singularities or not. Therefore, in this paper, we also consider *delicate* situations when the KL divergence of the true distribution from the singularities is comparable to the inverse of the number of training samples, n^{-1} , which are important situations in model selection problems and in statistical tests with finite number of samples for the following reasons: first, that there naturally exist a few true components with amplitude comparable to $n^{-1/2}$ when neither the smallest nor the largest model is selected; and secondly, that whether the selected model involves such components essentially affects generalization performance.

5 Theoretical Analysis

5.1 Subspace Bayes Solution

By $\parallel$ we, hereafter, distinguish the hyperparameter τ from the parameter w , for example, $p(y|x, w|\tau)$. Assume that the variance of a noise is known and equal to unity. Then, the conditional distribution of an LNN in the MIP version of SB approach is given by

$$p(y|x, A||B) = \frac{1}{(2\pi)^{N/2}} \exp\left(-\frac{\|y - BAx\|^2}{2}\right). \quad (11)$$

We use the following prior distribution:

$$\phi(A) = \frac{1}{(2\pi)^{HM/2}} \exp\left(-\frac{\text{tr}(A^t A)}{2}\right). \quad (12)$$

Note that we can similarly prepare $p(y|x, B||A)$ and $\phi(B)$ for the MOP version. We assume that the true conditional distribution is $p(y|x, A^*||B^*)$, where B^*A^* is the true map with rank $H^* \leq H$. We denote by $*$ the true value of a parameter as above, and by a *hat* an estimator of a parameter, for example, $\hat{A}$, $\hat{b}_h$, etc.. For simplicity, we assume that the input vector is orthonormalized so that $\int x x^t q(x) dx = I_M$. Consequently, the central limit theorem leads to the following two equations:

$$Q(X^n) = n^{-1} \sum_{i=1}^n x x^t = I_M + O(n^{-1/2}), \quad (13)$$

$$R(X^n, Y^n) = n^{-1} \sum_{i=1}^n y x^t = B^*A^* + O(n^{-1/2}), \quad (14)$$

where $Q(X^n)$ is an $M \times M$ symmetric matrix and $R(X^n, Y^n)$ is an $N \times M$ matrix. Hereafter, we abbreviate $Q(X^n)$ as Q , and $R(X^n, Y^n)$ as R .

Let γ_h be the h -th largest singular value of the matrix $RQ^{-1/2}$, ω_{a_h} the corresponding right singular vector, and ω_{b_h} the corresponding left singular vector. We find from Eq.(14) that γ_h for $H^* < h \leq H$ is of order $O(n^{-1/2})$. Hence, combining with Eq.(13), we get

$$\omega_{b_h} R Q^\rho = \omega_{b_h} R + O(n^{-1}) \quad \text{for } H^* < h \leq H, \quad (15)$$

where $-\infty < \rho \in \mathbb{R} < \infty$ is an arbitrary constant. The SB estimator, defined as the expectation value over the SB posterior distribution, is given by the following theorem:

Theorem 1 *Let $L = M$ in the MIP version or $L = N$ in the MOP version, and $L'_h = \max(L, n\gamma_h^2)$. The SB estimator of the map of an LNN is given by*

$$\hat{B}\hat{A} = \sum_{h=1}^H (1 - L'^{-1}_h L) \omega_{b_h} \omega_{b_h}^t R Q^{-1} + O(n^{-1}). \quad (16)$$

(The proof is given in Appendix A.)

Because the independence between A and B makes the posterior distribution localized, the following lemma holds.

Lemma 1 *The predictive distribution in the SB approaches can be written as follows:*

$$p(y|x, X^n, Y^n) = \left((2\pi)^N |\hat{V}|\right)^{-1/2} \cdot \exp\left(-\frac{(y - \hat{V} \hat{B} \hat{A} x)^t \hat{V}^{-1} (y - \hat{V} \hat{B} \hat{A} x)}{2}\right) + O(n^{-3/2}), \quad (17)$$

where $\hat{V} = I_N + O(n^{-1})$, and $|\cdot|$ denotes the determinant of a matrix.

(Proof) The predictive distribution is written as follows:

$$p(y|x, X^n, Y^n) = \langle p(y|x, A||B) \rangle_{p(A|X^n, Y^n||B)} \cdot \langle q(y|x) \exp\left(y^t (\hat{B}\hat{A} - B^*A^*)x\right) \rangle_{p(A|X^n, Y^n||B)} \quad (18)$$

where $\langle \cdot \rangle_p$ denotes the expectation value over a distribution p . Since $(\hat{B}\hat{A} - B^*A^*) = O(n^{-1/2})$ in the SB approaches, we can expand the predictive distribution as follows:

$$p(y|x, X^n, Y^n) \propto q(y|x) \left\langle 1 + y^t (\hat{B}\hat{A} - B^*A^*)x + \frac{y^t v v^t y}{2n} \right\rangle_{p(A|X^n, Y^n||B)} + O(n^{-3/2}), \quad (19)$$

where $v = \sqrt{n}(\hat{B}\hat{A} - B^*A^*)x$ is an N -dimensional vector of order $O(1)$. Calculating the expectation value and expanding the logarithm of Eq.(19), we arrive at Lemma 1. (Q.E.D.)

Comparing Eq.(16) with the ML estimator

$$\hat{B}\hat{A}_{MLE} = \sum_{h=1}^H \omega_{b_h} \omega_{b_h}^t R Q^{-1} \quad (20)$$

[Baldi and Hornik, 1995], we find that the SB estimator of each component is asymptotically equivalent to a positive-part JS type shrinkage estimator. Moreover, by virtue of Lemma 1, we can substitute the model at the SB estimator for the predictive distribution with asymptotically insignificant impact on generalization performance. Therefore, we conclude that the SB approach is asymptotically equivalent to the shrinkage estimation. Note that the variance of the prior distribution, Eq.(12), asymptotically has no effect upon prediction and hence upon generalization performance, as far as it is a positive, finite constant. We call L the degree of shrinkage. Remember that we can modify all the theorems in this paper for the ML estimation only by letting $L = 0$.

5.2 Generalization Error

Using the singular value decomposition of the true map B^*A^* , we can transform arbitrary A^* and B^* without change of the map into a matrix with its orthogonal row vectors and another matrix with its orthogonal column vectors, respectively. Accordingly, we assume the above orthogonalities without loss of generality. Then, Lemma 1 implies that the KL divergence, Eq.(9), with a set of n training samples is

given by

$$G(X^n, Y^n) = \left\langle \frac{\|(B^*A^* - \hat{B}\hat{A})x\|^2}{2} \right\rangle_{q(x)} + O(n^{-3/2}) \\ = \sum_{h=1}^H G_h(X^n, Y^n) + O(n^{-3/2}), \quad (21)$$

where

$$G_h(X^n, Y^n) = \frac{1}{2} \text{tr} \left((b_h^* a_h^{*t} - \hat{b}_h \hat{a}_h^t)^t (b_h^* a_h^{*t} - \hat{b}_h \hat{a}_h^t) \right) \quad (22)$$

is the contribution of the h -th component. Here $\text{tr}(\cdot)$ denotes the trace of a matrix. We denote by $\mathcal{W}_d(m, \Sigma, \Lambda)$ the d -dimensional Wishart distribution with m degrees of freedom, scale matrix Σ , and noncentrality matrix Λ , and abbreviate as $\mathcal{W}_d(m, \Sigma)$ the central Wishart distribution.

Theorem 2 *The generalization error of an LNN in the SB approaches can be asymptotically expanded as*

$$G(n) = \lambda n^{-1} + O(n^{-3/2}),$$

where the coefficient of the leading term, called the generalization coefficient in this paper, is given by

$$2\lambda = (H^*(M + N) - H^{*2}) \\ + \left\langle \sum_{h=1}^{H-H^*} \theta(\gamma_h'^2 > L) \left(1 - \frac{L}{\gamma_h'^2} \right)^2 \gamma_h'^2 \right\rangle_{q(\{\gamma_h'^2\})}. \quad (23)$$

Here $\theta(\cdot)$ is the indicator function of an event, $\gamma_h'^2$ is the h -th largest eigenvalue of a random matrix subject to $\mathcal{W}_{N-H^*}(M - H^*, I_{N-H^*})$, and $\langle \cdot \rangle_{q(\{\gamma_h'^2\})}$ denotes the expectation value over the distribution of the eigenvalues.

(Proof) According to Theorem 1, the difference between the SB and the ML estimators of a true component with a positive singular value is of order $O(n^{-1})$. Furthermore, the generalization error of the ML estimator of the component is the same as that of the regular models because of its identifiability. Hence, from Eq.(4), we obtain the first term of Eq.(23) as the contribution of the first H^* components. On the other hand, we find from Eq.(15) and Theorem 1 that for a redundant component, identifying $RQ^{-1/2}$ with R affects the SB estimator only of order $O(n^{-1})$, which, hence, does not affect the generalization coefficient. We say that U is the general diagonalized matrix of an $N \times M$ matrix T if T is singular value decomposed as $T = \Omega_b U \Omega_a$, where Ω_a and Ω_b are an $M \times M$ and an $N \times N$ orthogonal matrices, respectively. Let D be the general diagonalized matrix of R , and D' the $(N - H^*) \times (M - H^*)$ matrix created by removing the first H^* columns and rows from D . Then, the first H^* diagonal elements of D correspond to the positive true singular value components and D' consists only of noises. Therefore, D' is the general diagonalized matrix of $n^{-1/2} R'$, where R' is an $(N - H^*) \times (M - H^*)$ random matrix whose elements are independently subject to $\mathcal{N}_1(0, 1)$, so that $R' R'^t$ is subject to $\mathcal{W}_{N-H^*}(M - H^*, I_{N-H^*})$. The redundant components imitate $n^{-1/2} R'$. Hence, using Theorem 1 and Eq.(22), we obtain the second term of Eq.(23) as the contribution of the last $(H - H^*)$ components. Thus, we complete the proof of Theorem 2. (Q.E.D.)

5.3 Large Scale Approximation

In a similar fashion to the analysis of the ML estimation in [Fukumizu, 1999], the second term of Eq.(23) can be analytically calculated in the large scale limit when M , N , H , and H^* go to infinity in the same order. We define the following scalars: $\alpha = N'/M' = (N - H^*)/(M - H^*)$, $\beta = H'/N' = (H - H^*)/(N - H^*)$, and $\kappa = L/M' = L/(M - H^*)$. Let W be a random matrix subject to $\mathcal{W}_{N'}(M', I_{N'})$, and $\{u_1, \dots, u_{N'}\}$ the eigenvalues of $M'^{-1}W$. The measure of the empirical distribution of the eigenvalues is defined by

$$p(u)du = N'^{-1} \{ \delta(u_1) + \delta(u_2) + \dots + \delta(u_{N'}) \}, \quad (24)$$

where $\delta(u)$ denotes the Dirac measure at u . In the large scale limit, the measure, Eq.(24), converges almost everywhere to

$$p(u)du = \frac{\sqrt{(u - u_m)(u_M - u)}}{2\pi\alpha u} \theta(u_m < u < u_M) du, \quad (25)$$

where $u_m = (\sqrt{\alpha} - 1)^2$ and $u_M = (\sqrt{\alpha} + 1)^2$ [Watcher, 1978]. Calculating moments of Eq.(25), we obtain the following theorem:

Theorem 3 *The generalization coefficient of an LNN in the large scale limit is given by*

$$2\lambda = (H^*(M + N) - H^{*2}) + \frac{(M - H^*)(N - H^*)}{2\pi\alpha} \\ \{ J(s_t; 1) - 2\kappa J(s_t; 0) + \kappa^2 J(s_t; -1) \}, \quad (26)$$

where

$$J(s; 1) = 2\alpha(-s\sqrt{1-s^2} + \cos^{-1} s), \\ J(s; 0) = -2\sqrt{\alpha}\sqrt{1-s^2} + (1+\alpha)\cos^{-1} s \\ - (1-\alpha)\cos^{-1} \frac{\sqrt{\alpha}(1+\alpha)s + 2\alpha}{2\alpha s + \sqrt{\alpha}(1+\alpha)}, \\ J(s; -1) \\ = \begin{cases} 2\sqrt{\alpha} \frac{\sqrt{1-s^2}}{2\sqrt{\alpha}s+1+\alpha} - \cos^{-1} s + \frac{1+\alpha}{1-\alpha} \cos^{-1} \frac{\sqrt{\alpha}(1+\alpha)s+2\alpha}{2\alpha s + \sqrt{\alpha}(1+\alpha)} & (0 < \alpha < 1) \\ 2\sqrt{\frac{1-s}{1+s}} - \cos^{-1} s & (\alpha = 1) \end{cases}$$

and $s_t = \max((\kappa - (1 + \alpha))/2\sqrt{\alpha}, J^{-1}(2\pi\alpha\beta; 0))$. Here $J^{-1}(\cdot; k)$ denotes the inverse function of $J(s; k)$.

5.4 Delicate Situations

In ordinary asymptotic analysis, one considers only situations when the amplitude of each component of the true model is zero or *distinctly-positive*. Also Theorem 2 holds only in such situations. However, as mentioned in the last paragraph of Section 4, it is important to consider *delicate* situations when the true map B^*A^* has tiny but non-negligible singular values such that $0 < \sqrt{n}\gamma_h^* < \infty$. Theorem 1 still holds in such situations by replacing the second term of Eq.(16) with $o(n^{-1/2})$. We regard H^* as the number of *distinctly-positive* true singular values such that $\gamma_h^{*-1} = o(\sqrt{n})$. Without loss of generality, we assume that B^*A^* is a non-negative, general diagonal matrix with its diagonal elements arranged in non-increasing order. Let R'^* be the true submatrix created by removing the

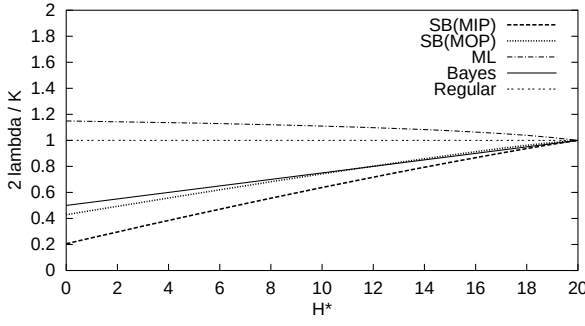


Figure 1: Generalization error.

first H^* columns and rows from B^*A^* . Then, D' , defined in the proof of Theorem 2, is the general diagonalized matrix of $n^{-1/2}R''$, where R'' is a random matrix such that $R''R''^t$ is subject to $\mathcal{W}_{N-H^*}(M-H^*, I_{N-H^*}, nR''^*R''^*)$. Therefore, we obtain the following theorem:

Theorem 4 *The generalization coefficient of an LNN in the general situations when the true map B^*A^* may have delicate singular values such that $0 < \sqrt{n}\gamma_h^* < \infty$ is given by*

$$2\lambda = (H^*(M+N) - H^{*2}) + \sum_{h=H^*+1}^H n\gamma_h^{*2} + \left\langle \sum_{h=1}^{H-H^*} \theta(\gamma_h''^2 > L) \left\{ \left(1 - \frac{L}{\gamma_h''^2}\right)^2 \gamma_h''^2 - 2 \left(1 - \frac{L}{\gamma_h''^2}\right) \gamma_h'' \omega_{b_h}'' \sqrt{n} R'^* \omega_{a_h}'' \right\} \right\rangle_{q(R'')}, \quad (27)$$

where γ_h'' , ω_{a_h}'' , and ω_{b_h}'' are the h -th largest singular value of R'' , the corresponding right singular vector, and the corresponding left singular vector, respectively, of which $\langle \cdot \rangle_{q(R'')}$ denotes the expectation value over the distribution.

6 Discussions

6.1 Comparison with the ML Estimation and with the Bayes Estimation

Figure 1 shows the generalization coefficients of an LNN with $M = 50$ input, $N = 30$ output, and $H = 20$ hidden units. The horizontal axis indicates the true rank H^* . The vertical axis indicates the coefficients normalized by the parameter dimension K , given by Eq.(4). The lines correspond to the generalization coefficients of the SB approaches, clarified in this paper, that of the ML estimation, clarified in [Fukumizu, 1999], that of the Bayes estimation, clarified in [Aoyagi and Watanabe, 2004], and that of the regular models, respectively.⁴ The results in Fig. 1 have been calculated in the large scale approximation, i.e., by using Theorem 3. We have also numerically calculated them by creating samples subject to the Wishart distribution and then using Theorem 2, and thus found that the both results almost coincide with each other so that we can hardly distinguish. We see in Fig. 1 that the SB approaches provide as good performance as the Bayes estimation, and that the MIP, moreover, has no greater generalization coefficient than the Bayes estimation for arbitrary H^* ,

⁴In the regular models, the normalized generalization coefficient is always equal to one, which leads to the penalty term of Akaike's information criterion [Akaike, 1974].

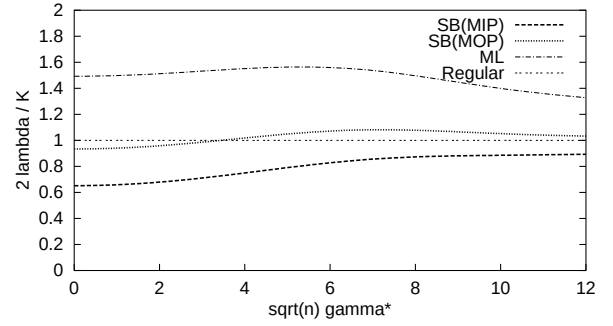


Figure 2: With *delicate* true components.

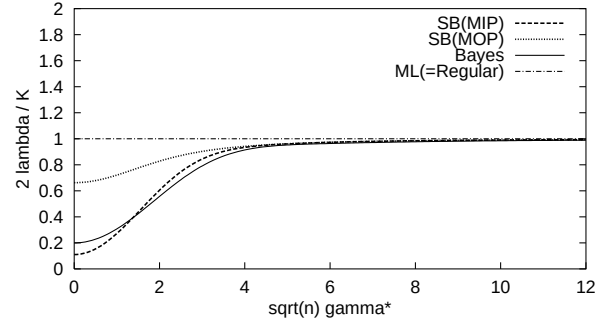


Figure 3: Single-output LNN.

which might seem to be inconsistent with the proved superiority of the Bayes estimation to any other learning method when we use the true prior distribution. This suspicion is cleared by consideration of *delicate* situations in the following.

Using Theorem 4, we can numerically calculate the SB, as well as the ML, generalization error in *delicate* situations when the true distribution is near the singularities. Figure 2 shows the coefficients of an LNN with $M = 50$ input, $N = 30$ output, and $H = 5$ hidden units on the assumption that the true map consists of $H^* = 1$ *distinctly-positive* component, three *delicate* components whose singular values are identical to each other, and the other one null component. The horizontal axis indicates $\sqrt{n}\gamma^*$, where $\gamma_h^* = \gamma^*$ for $h = 2, \dots, 4$. The Bayes generalization error in *delicate* situations was previously clarified [Watanabe and Amari, 2003], but unfortunately, only in single-output (SO) LNNs, i.e., $N = H = 1$.⁵ Figure 3 shows the coefficients of an SOLNN with $M = 5$ input units on the assumption that $H^* = 0$ and the true singular value of the one component, indicated by the horizontal axis, is *delicate*. We see in Fig. 3 that the SB approaches have a property similar to the Bayes estimation, suppression of overfitting by the entropy of the singularities. We also see that in some *delicate* situations, the MIP is worse than the Bayes estimation, which shows consistency with the superiority of the Bayes estimation. We conclude that in typical cases, the suppression by the singularities in the MIP is comparable to, or sometimes stronger than, that

⁵An SOLNN is regarded as a regular model at a view point of the ML estimation because the transform $b_1 a_1 \mapsto w \in \mathbb{R}^M$ makes the model linear and hence identifiable, and therefore, the ML generalization error is identical to that of the regular models. Nevertheless, an SOLNN has a property of unidentifiable models at a view point of the Bayesian learning methods, as shown in Fig. 3.

in the Bayes estimation.

It would be more fortunate if any of the SB approaches, which require much less computational costs than MCMC methods, would always provide comparable generalization performance to the Bayes estimation. However, the SB approaches have also a property similar to the ML estimation, acceleration of overfitting by selection of the largest singular values of a random matrix. Because of selection from a large number of random variables subject to non-compact support distribution, the $(H - H^*)$ largest eigenvalues of a random matrix subject to $\mathcal{W}_{N-H^*}(M - H^*, I_{N-H^*})$ are much greater than L when $(M - H^*) > (N - H^*) \gg (H - H^*)$. Therefore, the eigenvalues $\{\gamma_h^2\}$ in Theorem 2 go out of the effective range of shrinkage, and consequently, the SB approaches approximate the ML estimation in such atypical cases. Actually, the generalization coefficient of an LNN in the case that $M = N = 80$, $H = 1$, $H^* = 0$ is $2\lambda/K \sim 1.04$, which is greater than that of the regular models, though in the Bayes estimation, the generalization coefficient never exceeds that of the regular models [Watanabe, 2001].

6.2 Relation to Shrinkage Estimation

The relation between an EB approach in a linear model and the JS estimator, into which the SB estimator, Eq.(16), in an SOLNN is changed by letting $L = (M - 2)$, was discussed in [Efron and Morris, 1973]. Based on the EB approach, the JS estimator can be derived as the solution of an equation with respect to an unbiased estimator of the hyperparameter τ_1^{-2} , introduced in Section 3.2. The similarity between the JS and the SB estimators is natural because in an SOLNN, the transform $b_1 a_1 \mapsto w \in \mathbb{R}^M$ makes not only the model linear but also the prior distribution as Eq.(10).

We focus on SOLNNs in this paragraph. In Fig. 3, the SB approaches and the Bayes estimation seem to be superior to the ML estimation regardless of the true distribution. The following asymptotic expansion of the generalization coefficient with respect to $\sqrt{n}\gamma_1^*$ provides a clue when it occurs:

$$2\lambda = M - \xi(\sqrt{n}\gamma_1^*)^{-2} + o((\sqrt{n}\gamma_1^*)^{-2}), \quad (28)$$

where ξ is the coefficient of the leading term when γ_1^* increases to be *distinctly-positive*. The sign of ξ indicates the direction of approach to the line $2\lambda = M$, which corresponds to the generalization coefficient of the regular models. It was found that $\xi = (M - 1)(M - 3)$ in the Bayes estimation, which leads to the conjecture that the Bayes estimation would be superior to the ML estimation when $M \geq 4$ [Watanabe and Amari, 2003]. Similarly expanding Eq.(27), we get $\xi = M(M - 4)$ in the MIP, as well as $\xi = (2M - 5)$ in the MOP, which leads to the conjecture that the MIP when $M \geq 5$, as well as the MOP when $M \geq 3$, would be superior to the ML estimation. We have found that the numerical calculation by using Theorem 4 supports the conjecture above [Nakajima and Watanabe, 2005a]. We also find that $\xi = (M - 2)^2$ in the JS estimation, which is consistent with its proved superiority to the ML estimation when $M \geq 3$.

6.3 Relation to Variational Bayes Approach

The generalization error of the variational Bayes (VB) approach in LNNs has just been clarified [Nakajima and Watanabe, 2005b]. In the parameter subspace corresponding to the

redundant components, the VB posterior distribution extends with its variance of order 1 in the larger dimension parameter subspace either input one or the output one; while the SB posterior distribution extends with its variance of order 1 in the parameter space, not in the hyperparameter space, as we find from Eqs. (30) and (37) in Appendix A. Consequently, the VB approach is asymptotically equivalent to the MIP version of SB approach.

6.4 Future Works

A future work can be consideration of the effect of non-linearity of the activation function, $\psi(\cdot)$ in Eq.(1). We expect that the non-linearity would extend the range of basis selection and hence increase the generalization error.

7 Conclusions

We have introduced a subspace Bayes (SB) approach, an empirical Bayes approach where a part of the parameters are regarded as hyperparameters, and derived the solution of two versions of SB approach in three-layer linear neural networks. We have also clarified its generalization error and concluded that the SB approaches have a property similar to the Bayes estimation and provide as good performance as the Bayes estimation in typical cases.

Acknowledgments

The authors would like to thank the reviewers who gave us meaningful advice, which motivates us to add Section 4 and some comments in other sections. They also would like to thank Kazuo Ushida, Masahiro Nei, and Nobutaka Magome of Nikon Corporation for encouragement to research.

A Proof of Theorem 1

First, we will prove in the MIP version. Given an arbitrary map BA , we can have A with its orthogonal row vectors and B with its orthogonal column vectors by using the singular value decomposition. Just in that case, the prior probability, Eq.(12), is maximized. Accordingly, we assume without loss of generality that the optimum value of B consists of its orthogonal column vectors. Consequently, the marginal (conditional) likelihood, as well as the posterior distribution, factorizes as

$$\begin{aligned} Z(Y^n|X^n\|B) &= \prod_{h=1}^H Z(Y^n|X^n\|b_h), \\ p(A|X^n, Y^n\|B) &= \prod_{h=1}^H p(a_h|X^n, Y^n\|b_h). \end{aligned}$$

Then, substituting Eqs.(11) and (12) into Eq.(5), as well as Eq.(6), we can easily derive the marginal likelihood, as well as the posterior distribution, of each component as follows:

$$Z(Y^n|X^n\|b_h) \propto |S_h|^{-1/2} \exp\left(\frac{nb_h^t R S_h^{-1} R^t b_h}{2}\right), \quad (29)$$

$$\begin{aligned} p(a_h|X^n, Y^n\|b_h) \\ \propto \exp\left(-\left(a_h - S_h^{-1} R^t b_h\right)^t \frac{n S_h}{2} \left(a_h - S_h^{-1} R^t b_h\right)\right), \quad (30) \end{aligned}$$

where $S_h = (\|b_h\|^2 Q + n^{-1} I_M)$. Let $F'(Y^n|X^n\|b_h) = F(Y^n|X^n\|b_h) + \text{const.}$, where $F(Y^n|X^n\|b_h)$ is the stochastic complexity, i.e., the negative log marginal likelihood, of the h -th component. Then, we get

$$2F'(Y^n|X^n\|b_h) = -2 \log Z(Y^n|X^n\|b_h) + \text{const.} \\ = \log |S_h| - n b_h^t R S_h^{-1} R^t b_h \quad (31)$$

Hereafter, separately considering the components imitating the positive true ones and the redundant components, we will minimize Eq.(31). We abbreviate $F'(Y^n|X^n\|b_h)$ as $F'(b_h)$.

For a positive true component, $h \leq H^*$, the corresponding observed singular value γ_h of $RQ^{-1/2}$ is of order 1 with probability 1. Then, from Eq.(31), we get

$$2F'(b_h) = M \log \|b_h\|^2 - n \|b_h\|^{-2} b_h^t R Q^{-1} R^t b_h \\ + \|b_h\|^{-4} b_h^t R Q^{-2} R^t b_h + O(n^{-1}). \quad (32)$$

To minimize Eq.(32), the leading, second term dominates the determination of the direction cosine of b_h and leads to $b_h = \|b_h\|(\omega_{b_h} + O(n^{-1}))$. The first and the third terms determine the norm of b_h because the second term is independent of it. Thus, we get the optimal hyperparameter value as follows:

$$\hat{b}_h = \sqrt{\frac{\omega_{b_h}^t R Q^{-2} R^t \omega_{b_h}}{M}} \omega_{b_h} + O(n^{-1}). \quad (33)$$

Because the average of a_h over the posterior distribution, Eq.(30), is $\hat{a}_h = S_h^{-1} R^t b_h$, we obtain the SB estimator for the positive true component of the map BA as follows:

$$\hat{b}_h \hat{a}_h^t = \hat{\omega}_{b_h} \hat{\omega}_{b_h}^t R Q^{-1} + O(n^{-1}). \quad (34)$$

On the other hand, for a redundant component, $h > H^*$, Eq.(15) allows us to approximate Eq.(31) as follows:

$$2F'(b_h) = M \log (\|b_h\|^2 + n^{-1}) - \frac{n b_h^t R R^t b_h}{\|b_h\|^2 + n^{-1}} + O(n^{-1/2}). \quad (35)$$

Then, we find that the direction cosine of b_h , determined by the second term of Eq.(35), is approximated by ω_{b_h} with accuracy $O(n^{-1/2})$. After substituting $\gamma_h^2 \|b_h\|^2 (1 + O(n^{-1/2}))$ for $b_h^t R R^t b_h$, we get the following extreme condition by partial differentiation of Eq.(35) with respect to the norm of b_h :

$$0 = 2 \frac{\partial F'(b_h)}{\partial \|b_h\|^2} = \frac{M}{(\|b_h\|^2 + n^{-1})^2} \left(\|b_h\|^2 - \frac{n \gamma_h^2 - M}{n M} \right) \\ + O(\|b_h\|^{-2} n^{-1/2}). \quad (36)$$

We find that Eq.(36) has no solution if γ_h is less than $\sqrt{M/n}$. Therefore, using the fact that Eq.(35) diverges to infinity with arbitrary n if $\|b_h\| \rightarrow \infty$, we get the optimum hyperparameter value as follows:

$$\hat{b}_h = \sqrt{\frac{L'_h - M}{n M}} \omega_{b_h} + O(n^{-1}). \quad (37)$$

Thus, we obtain the SB estimator of the redundant component as follows:

$$\hat{b}_h \hat{a}_h^t = (1 - (L'_h)^{-1} M) \omega_{b_h} \omega_{b_h}^t R + O(n^{-1}). \quad (38)$$

Selecting the largest singular value components minimizes Eq.(31). Hence, combining Eqs.(34) and (38), and then using Eq.(15), we obtain the SB estimator in Theorem 1. We can also derive the SB estimator in the MOP version in exactly the same way. (Q.E.D.)

References

- [Akaike, 1974] H. Akaike. A new look at statistical model. *IEEE Trans. on Automatic Control*, 19:716–723, 1974.
- [Akaike, 1980] H. Akaike. Likelihood and bayes procedure. In *J. M. Bernald, Bayesian statistics*, pages 143–166, Valencia, Italy, 1980. University Press.
- [Aoyagi and Watanabe, 2004] M. Aoyagi and S. Watanabe. The generalization error of reduced rank regression in bayesian estimation. In *Proc. of ISITA*, pages 1068–1073, Parma, Italy, 2004.
- [Attias, 1999] H. Attias. Inferring parameters and structure of latent variable models by variational bayes. In *Proc. of UAI*, 1999.
- [Baldi and Hornik, 1995] P. F. Baldi and K. Hornik. Learning in linear neural networks: a survey. *IEEE Trans. on Neural Networks*, 6:837–858, 1995.
- [Efron and Morris, 1973] B. Efron and C. Morris. Stein's estimation rule and its competitors—an empirical bayes approach. *J. of Am. Stat. Assoc.*, 68:117–130, 1973.
- [Fukumizu, 1999] K. Fukumizu. Generalization error of linear neural networks in unidentifiable cases. In *Proc. of ALT*, pages 51–62. Springer, 1999.
- [Ghahramani and Beal, 2000] Z. Ghahramani and M. J. Beal. Graphical models and variational methods. In *Advanced Mean Field Methods*. MIT Press, 2000.
- [Hinton and van Camp, 1993] G. E. Hinton and D. van Camp. Keeping neural networks simple by minimizing the description length of the weights. In *Proc. of COLT*, 1993.
- [James and Stein, 1961] W. James and C. Stein. Estimation with quadratic loss. In *Proc. of the 4th Berkeley Symp. on Math. Stat. and Prob.*, pages 361–379, Berkeley:University of California, 1961.
- [Kass and Steffey, 1989] R. E. Kass and D. Steffey. Approximate bayesian inference in conditionally independent hierarchical models (parametric empirical bayes models). *J. of the Am. Stat. Assoc.*, 84:717–726, 1989.
- [MacKay, 1995] D. J. C. MacKay. Developments in probabilistic modeling with neural networks—ensemble learning. In *Proc. of the 3rd Ann. Symp. on Neural Networks*, pages 191–198, 1995.
- [Nakajima and Watanabe, 2005a] S. Nakajima and S. Watanabe. Generalization performance of subspace bayes approach in linear neural networks. *Submitted to IEICE Trans.*, 2005.
- [Nakajima and Watanabe, 2005b] S. Nakajima and S. Watanabe. Variational bayes solution of linear neural networks and its generalization error and training error. *Submitted to ICML*, 2005.
- [Reinsel and Velu, 1998] G. C. Reinsel and R. P. Velu. *Multivariate Reduced-Rank Regression*. Springer, 1998.
- [Watanabe and Amari, 2003] S. Watanabe and S. Amari. Learning coefficients of layered models when the true distribution mismatches the singularities. *Neural Computation*, 15:1013–1033, 2003.
- [Watanabe, 2001] S. Watanabe. Algebraic information geometry for learning machines with singularities. *Advances in NIPS*, 13:329–336, 2001.
- [Watcher, 1978] K. W. Watcher. The strong limits of random matrix spectra for sample matrices of independent elements. *Ann. Prob.*, 6:1–18, 1978.
- [Yamazaki and Watanabe, 2003] K. Yamazaki and S. Watanabe. Singularities in mixture models and upper bounds of stochastic complexity. *Neural Networks*, 16(7):1029–1038, 2003.

Phase Transitions within Grammatical Inference

Nicolas Pernot, Antoine Cornuéjols & Michèle Sebag

Laboratoire de Recherche en Informatique, CNRS UMR 8623

Bât.490, Université de Paris-Sud, Orsay

91405 Orsay Cedex (France)

antoine.sebag@lri.fr

Abstract

It is now well-known that the feasibility of inductive learning is ruled by statistical properties linking the empirical risk minimization principle and the “capacity” of the hypothesis space. The discovery, a few years ago, of a phase transition phenomenon in inductive logic programming proves that other fundamental characteristics of the learning problems may similarly affect the very possibility of learning under very general conditions.

Our work examines the case of grammatical inference. We show that while there is no phase transition when considering the whole hypothesis space, there is a much more severe “gap” phenomenon affecting the effective search space of standard grammatical induction algorithms for deterministic finite automata (DFA). Focusing on the search heuristics of the RPNI and RED-BLUE algorithms, we show that they overcome this problem to some extent, but that they are subject to overgeneralization. The paper last suggests some directions for new generalization operators, suited to this Phase Transition phenomenon.

1 Introduction

It is now well-known that the feasibility of inductive learning is ruled by statistical properties linking the empirical risk minimization principle and the “capacity” of the hypothesis space [Vapnik, 1995]. While this powerful framework leads to a much deeper understanding of machine learning and to many theoretical and applicative breakthroughs, it basically involves only statistical information on the learning search space, e.g. the so-called VC-dimension. The dynamics of the learning search is not considered.

Independently, a new combinatoric paradigm has been studied in the Constraint Satisfaction community since the early 90s, motivated by computational complexity concerns [Cheeseman *et al.*, 1991]: where are the really hard problems? Indeed, the worst case complexity analysis poorly accounts for the fact that, despite an exponential worst-case complexity, empirically, the complexity is low for most CSP instances. These remarks led to developing the so-called

phase transition framework (PT) [Hogg *et al.*, 1996], which considers the satisfiability and the resolution complexity of CSP instances as random variables depending on order parameters of the problem instance (e.g. constraint density and tightness). This framework unveiled an interesting structure of the CSP landscape. Specifically, the landscape is divided into three regions: the YES region, corresponding to underconstrained problems, where the satisfiability probability is close to 1 and the average complexity is low; the NO region, corresponding to overconstrained problems, where the satisfiability probability is close to 0 and the average complexity is low too; last, a narrow region separating the YES and NO regions, referred to as *phase transition region*, where the satisfiability probability abruptly drops from 1 to 0 and which concentrates on average the computationally heaviest CSP instances.

The phase transition paradigm has been transported to relational machine learning and inductive logic programming (ILP) by [Giordana and Saitta, 2000], motivated by the fact that the covering test most used in ILP [Muggleton and Raedt, 1994] is equivalent to a CSP. As anticipated, a phase transition phenomenon appears in the framework of ILP: a wide YES (respectively NO) region includes all hypotheses which cover (resp. reject) all examples, and the hypotheses that can discriminate the examples lie in the narrow PT, where the average computational complexity of the covering test reaches its maximum.

Besides computational complexity, the PT phenomenon has far-reaching effects on the success of relational learning [Botta *et al.*, 2003]. For instance, a wide *Failure Region* is observed: for all target concepts/training sets in this region, no learning algorithms among the prominent ILP ones could find hypotheses better than random guessing [Botta *et al.*, 2003].

These negative results lead to a better understanding of the intrinsic limits of the existing ILP algorithms and search biases. Formally, consider a greedy specialization (top-down) search strategy: starting its exploration in the YES region, the system is almost bound to make random specialization choices, for all hypotheses in this region cover every example on average. The YES region constitutes a rugged plateau from a search perspective, and there is little chance that the algorithm ends in the right part of the PT region, where good hypotheses lie. A similar reasoning goes for algorithms that follow a greedy generalization strategy.

The phase transition paradigm thus provides another perspective on the pitfalls facing machine learning, focusing on the combinatoric search aspects while statistical learning focuses on the statistical aspects.

The main question studied in this paper is whether the PT phenomenon is limited to relational learning, or threatens the feasibility and tractability of other learning settings as well.

A learning setting with intermediate complexity between full relational learning and propositional learning is thus considered, that of grammatical inference (GI) [Pitt, 1989; Sakakibara, 1997]. Only the case of Finite-State Automata (FSA, section 2), will be considered through the paper. Specifically, the phase transition phenomenon will be investigated with respect to three distributions on the FSA space, incorporating gradually increasing knowledge on the syntactical and search biases of GI algorithms.

The first distribution incorporates no information on the algorithm and considers the whole space of FSA. Using a set of order parameters, the average coverage of automata is studied analytically and empirically.

The second one reflects the bias introduced by the generalization relations defined on the FSA space and exploited by GI algorithms. The vast majority of these algorithms first construct a least general generalization of the positive examples, or Prefix Tree Acceptor (PTA), and restrict the search to the generalizations of the PTA, or *generalization cone*¹.

The third distribution takes into account the heuristics used by GI algorithms, guiding the search trajectory in the generalization cone. Due to space limitations, the study is restricted to two prominent GI algorithms, namely RPNI [Oncina and Garcia, 1992] and RED-BLUE [Lang *et al.*, 1998].

This paper is organized as follows. Section 2 briefly introduces the domain of Grammatical Inference, the principles of the inference algorithms and defines the order parameters used in the rest of the paper. Section 3 investigates the existence and potential implications of phase transition phenomena in the whole FSA space (section 3.1) and in the generalization cone (section 3.2). Section 4 focuses on the actual landscape explored by GI algorithms, specifically considering the search trajectories of RPNI and RED-BLUE. Section 5 discusses the scope of the presented study and lays out some perspectives for future research.

2 Grammatical inference

After introducing general notations and definitions, this section briefly discusses the state of the art and introduces the order parameters used in the rest of the paper.

2.1 Notations and definitions

Grammatical inference is concerned with inferring grammars from positive (and possibly negative) examples. Only regular grammars are considered in this paper. They form the bottom class of the hierarchy of formal grammars as defined

¹More precisely, the Prefix Tree Acceptor is obtained by merging the states that share the same prefix in the Maximal Canonical Automaton (MCA), which represents the whole positive learning set as an automaton. A PTA is therefore a DFA with a tree-like structure.

by Chomsky, yet are sufficiently rich to express many interesting sequential structures. Their identification in the limit from positive examples only is known to be impossible, while it is feasible with a complete set of examples [Gold, 1967].

It is known that any regular language can be produced by a finite-state automaton (FSA), and that any FSA generates a regular language. In the remaining of the paper, we will mostly use the terminology of finite-state automata. A FSA is a 5-tuple $A = \langle \Sigma, Q, Q_0, F, \delta \rangle$ where Σ is a finite alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states, δ is the transition function defined from $Q \times \Sigma$ to 2^Q .

A positive example of a FSA is a string on Σ , produced by following any path in the graph linking one initial state q_0 to any accepting state.

A finite state-automaton (FSA) is deterministic (DFA) if Q_0 contains exactly one element q_0 and if $\forall q \in Q, \forall x \in \Sigma, \text{Card}(\delta(q, x)) \leq 1$. Otherwise it is non-deterministic (NFA). Every NFA can be translated into an equivalent DFA, but at the price of being possibly exponentially more complex in terms of number of states. Given any FSA A , there exists a minimum state DFA (also called *canonical DFA*) A such that $L(A) = L(A')$ (where $L(A)$ denotes the set of strings accepted by A). Without loss of generality, it can be assumed that the target automaton being learned is a canonical DFA.

A set S^+ is said to be *structurally complete* with respect to a DFA A if S^+ covers each transition of A and uses every element of the set of final states of A as an accepting state. Clearly, $L(PTA(S^+)) = S^+$.

Given a FSA A and a partition π on the set of states Q of A , the *quotient automaton* is obtained by merging the states of A that belong to the same block in partition π (see [Dupont *et al.*, 1994] for more details). Note that a quotient automaton of a DFA might be a NFA and vice versa. The set of all quotient automata obtained by systematically merging the states of a DFA A represents a lattice of FSAs. This lattice is ordered by the *grammar cover* relation $\preceq$. The transitive closure of $\preceq$ is denoted by $\ll$. We say that $A_{\pi_i} \ll A_{\pi_j}$ iff $L(A_{\pi_i}) \subseteq L(A_{\pi_j})$. Given a canonical DFA A and a set S^+ that is structurally complete with respect to A , the lattice derived from $PTA(S^+)$ is guaranteed to contain A .

From these assumptions, follows the paradigmatic approach of most grammatical inference algorithms (see, e.g., [Coste, 1999; Dupont *et al.*, 1994; Pitt, 1989; Sakakibara, 1997]), which equates generalization with state merging operations starting from the PTA.

2.2 Learning biases in grammatical inference

The core task of GI algorithms is thus to select iteratively a pair of states to be merged. The differences among algorithms is related to the choice of: (i) the search criterion (which merge is the best one); (ii) the search strategy (how is the search space explored); and (iii) the stopping criterion.

We shall consider here the setting of learning FSAs from positive and negative examples, and describe the algorithms studied in section 3. In this setting, the stopping criterion is determined from the negative examples: generalization proceeds as long as the candidate solutions remain correct, not

covering any negative example²

The RPNI algorithm [Oncina and Garcia, 1992] uses a depth first search strategy with some backtracking ability, favoring the pair of states which is closest to the start state, such that their generalization (FSA obtained by merging the two states and subsequently applying the determinisation operator) does not cover any negative example.

The RED-BLUE algorithm (also known as BLUE-FRIDGE) [Lang *et al.*, 1998] uses a beam search from a candidate list, selecting the pair of states after the *Evidence-Driven State Merging* (EDSM) criterion, i.e. such that their generalization involves a minimal number of final states. RED-BLUE thus also performs a search with limited backtracking, based on a more complex criterion and a wider search width than RPNI.

2.3 Order Parameters

Following the methodology introduced in [Giordana and Saitta, 2000], the PT phenomenon is investigated along so-called *order parameters* chosen in accordance with the parameters used in the *Abbadingo* challenge [Lang *et al.*, 1998]:

- The number Q of states in the DFA.
- The number B of output edges on each state.
- The number L of letters on each edge.
- The fraction a of accepting states, taken in $[0,1]$.
- The size $|\Sigma|$ of the alphabet considered.
- The length ℓ of the test examples. Also the maximal length ℓ of the learning examples in S^+ (as explained below).

The study first focuses on the intrinsic properties of the search space (section 3) using a *random sampling* strategy (all ℓ letters in the string being independently and uniformly drawn in Σ). In section 4, we examine the capacity of the studied learning algorithms to approximate a target automaton, based on *positive sampling*, where each training string is produced by following a path in the graph, randomly selecting an output edge in each step³.

3 Phase Transitions: the FSA space and the generalization cone

This section investigates the percentage of coverage of deterministic and non-deterministic Finite-State Automata, either uniformly selected (section 3.1), or selected in the subspace actually investigated by grammatical inference algorithms, that is, the generalization cone (section 3.2).

3.1 Phase Transition in the whole FSA space

The sampling mechanism on the whole deterministic FSA space (DFA) is defined as follows. Given the order parameter values $(Q, B, L, a, |\Sigma|)$:

²In this paper, after the standard Machine Learning terminology, a string is said to be covered by a FSA *iff* it belongs to the language thereof.

³The string is cut at the last accepting state met before arriving at length ℓ , if any; otherwise it is rejected.

- for every state q , (i) B output edges (q, q') are created, where q' is uniformly selected with no replacement among the Q states; (ii) $L \times B$ distinct letters are uniformly selected in Σ ; and (iii) these letters are evenly distributed among the B edges above.
- every state q is turned into an accepting state with probability a .

The sampling mechanism for NFA differs from the above in a single respect: two edges with same origin state are not required to carry distinct letters.

For each setting of the order parameters, 100 independent problem instances are constructed. For each considered FSA (the sampling mechanisms are detailed below), the coverage rate is measured as the percentage of covered examples among 1,000 examples (strings of length ℓ) uniformly sampled.

Fig. 1 shows the average coverage in the (a, B) plane, for $|\Sigma| = 2$, $L = 1$ and $\ell = 10$, where the accepting rate a varies in $[0, 1]$ and the branching factor B varies in $\{1, 2\}$. Each point reports the average coverage of a sample string s by a FSA (averaged over 100 FSA drawn with accepting rate a and branching factor B , tested on 1,000 strings s of length ℓ).

These empirical results are analytically explained from the simple equations below, giving the probability that a string of length ℓ be accepted by a FSA defined on an alphabet of size $|\Sigma|$, with a branching factor B and L letters on each edge, in the DFA and NFA cases (the number of states Q is irrelevant here).

$$P(\text{accept}) = \begin{cases} a \cdot \left(\frac{B \cdot L}{|\Sigma|}\right)^\ell & \text{for a DFA} \\ a \cdot [1 - (1 - \frac{L}{|\Sigma|})^B]^\ell & \text{for a NFA} \end{cases}$$

The coverage of the FSA decreases as a and B decrease. The slope is more abrupt in the DFA case than in the NFA case; still, there is clearly no phase transition here.

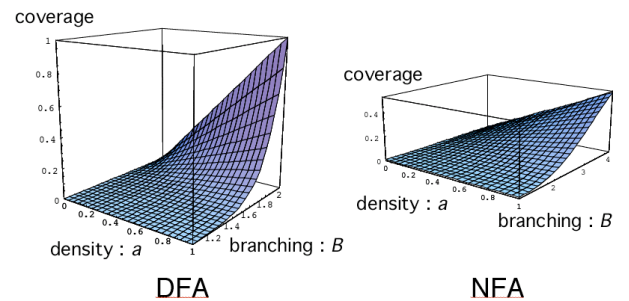


Figure 1: Coverage landscapes for Deterministic and Non-Deterministic FSA, for $|\Sigma|=2$, $L=1$ and $\ell=10$. The density of accepting states a and the branching factor B respectively vary in $[0, 1]$ and $\{1, 2\}$.

3.2 PT in the Generalization Cone

The coverage landscape displayed in Fig. 1 might suggest that grammatical inference takes place in a well-behaved search space. However, grammatical inference algorithms do

not explore the whole FSA space. Rather, as stated in section 2.1, the search is restricted to the generalization cone, the set of generalizations of the PTA formed from the set S^+ of the positive examples. The next step is thus to consider the search space actually explored by GI algorithms.

A new sampling mechanism is defined to explore the DFA generalization cone:

1. $|S^+|$ ($= 200$ in the experiments) examples of length ℓ are uniformly and independently sampled within the space of all strings of length $< \ell$, and the corresponding PTA is constructed;
2. N ($= 50$ in the experiments) PTAs are constructed in that way.
3. K ($= 20$ in the experiments) generalization paths, leading from each PTA to the most general FSA or Universal Acceptor (UA), are constructed;
In each generalization path ($A_0 = PTA, A_1, \dots, A_t = UA$), the i -th FSA A_i is constructed from A_{i-1} by merging two uniformly selected states in A_{i-1} , and subsequently applying the determinisation operator.
4. The generalization cone sample is made of all the FSAs in all generalization paths (circa 270,000 FSAs in the experiments).

The sampling mechanism on the non-deterministic generalisation cone differs from the above in a single respect: the determinisation operator is never applied.

Fig. 2 shows the behaviour of the coverage in the DFA generalisation cone for $|\Sigma| = 4$ and $\ell = 8$. Each DFA A is depicted as a point with coordinates (Q, c) , where Q is the number of states of A and c is its coverage (measured as in section 3). The coverage rate for each FSA in the sample is evaluated from the coverage rate on 1000 test strings of length ℓ .

Fig. 3 similarly shows the behaviour of the coverage in the NFA generalisation cone, with $|\Sigma| = 4$ and $\ell = 16$.

Fig 2, typical of all experimental results in the range of observation ($|\Sigma| = 2, 4, 8, 16$, and $\ell = 2, 4, 6, 8, 16, 17$), shows a clear-cut phase transition. Specifically, here, the coverage abruptly jumps from circa 13% to 54%; and this jump coincides with a gap in the number of states of the DFAs in the generalization cone: no DFA with a number of states in $[180, 420]$ was found. The gap is even more dramatic as the length of the training and test sequences ℓ is increased.

Interestingly, a much smoother picture appears in the non-deterministic case (fig. 3); although the coverage rapidly increases when the number of states decreases from 300 to 200, no gap can be seen, neither in the number of states nor in the coverage rate itself⁴.

In the following, we focus on the induction of DFAs.

4 Phase transition and search trajectories

The coverage landscape, for the DFAs, shows a hole in the generalization cone, with a density of hypotheses of coverage

⁴The difference with the DFA case is due to the determinisation process that forces further states merging when needed. A diffusion like analytical model was devised, that predicts the observed start of the gap with 15% precision.

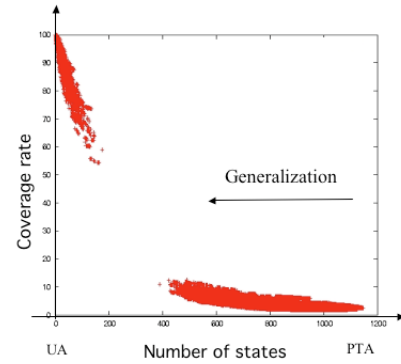


Figure 2: Coverage landscape in the DFA generalization cone ($|\Sigma| = 4$, $\ell = 8$). At the far right stand the 50 PTA sampled, with circa 1150 states each. The generalization cone of each PTA includes 1,000 generalization paths, leading from the PTA to the Universal Acceptor. Each point reports the coverage of a DFA, evaluated over a sample of 1,000 strings. This graph shows the existence of a large gap regarding both the number of states and the coverage of the DFAs that can be reached by generalization.

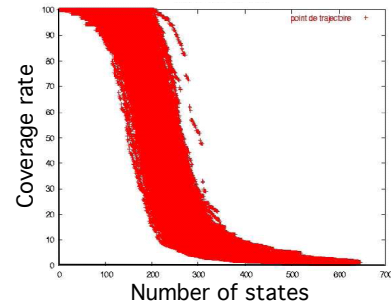


Figure 3: Coverage landscape in the NFA generalization cone, with same order parameters as in fig. 2.

in between a large interval (typically between less than 20% to approximately 60%) falling abruptly. Therefore, a random exploration of the generalization cone would face severe difficulties in finding a hypothesis in this region and would likely return hypotheses of poor performance if the target concept had a coverage rate in this “no man’s land” interval.

It is consequently of utmost importance to examine the search heuristics that are used in the classical grammatical inference systems. First, are they able to thwart the *a priori* very low density of hypotheses in the gap? Second, are they able to guide the search toward hypotheses of appropriate coverage rate, specially if this coverage falls in the gap?

The study will thus focus on two standard algorithms in grammatical inference, namely the RPNI and the RED-BLUE algorithms [Oncina and Garcia, 1992; Lang *et al.*, 1998].

4.1 Experimental setting

Previous experiments considered training sets made of positive randomly drawn strings sequences only. However, in order to assess the performance of learning algorithms, the hypothesis learned must now be compared to the target automaton. Therefore, another experimental setting is used in this section, with the sampling of target automata, and the construction of training and test sets. These data sets include positive and negative examples as most GI algorithms (and specifically RPNI and RED-BLUE) use negative examples in order to stop the generalization process.

In our first experiments, we tested whether heuristically guided inference algorithms can find good approximations of the target automata considering target automata with approximately (i) 50% coverage rate (as considered in the influential *Abbadingo* challenge, and in the middle of the “gap”), and (ii) 5% coverage rate.

For each target coverage rate, we used the experimental setting described in [Lang *et al.*, 1998] in order to retain a certain number of target automata with a mean size of Q states ($Q = 50$, in our experiments). For each automaton then, we generated N ($=20$) training sets of size $|S|$ ($= 100$) labeled according to the target automaton, with an equal number of positive and negative instances ($|S^+| = |S^-| = 50$) of length $\ell = 14$. The coverage rate was computed as before on 1000 uniformly drawn strings (with no intersection with the training set).

In a second set of experiments, we analyzed the learning performances of the algorithms with respect to test errors, both false positive and false negative.

In these experiments, we chose the type of target automata by setting the number of states Q and some predetermined structural properties⁵.

4.2 The heuristically guided search space

Due to space limitation, only the graph obtained for the RPNI algorithm is reported (see figure 4), with three typical learning trajectories. Similar results were obtained with the RED-BLUE algorithm.

One immediate result is that both the RPNI and the EDSM heuristics manage to densely probe the “gap”. This can explain why the gap phenomenon was not discovered before, and why the RED-BLUE algorithm for instance could solve some cases of the *Abbadingo* challenge where the target concepts have a coverage rate of approximately 50%. However, where RPNI tends to overspecialize the target automaton, RED-BLUE tends to overgeneralize it by 5% to 10%.

In order to test the capacity of the algorithms to return automata with a coverage rate close to the target coverage, we repeated these experiments with target automata of coverage rate of approximately 3%. The results (figure 5) shows that, in this case, RPNI ends up with automata of coverage 4 to 6 times greater than the target coverage. The effect is even more pronounced with RED-BLUE which returns automata of average coverage rate around 30%!

⁵The datasets and more detail are available at <http://www.lri.fr/~antoine/www/pt-gi/>

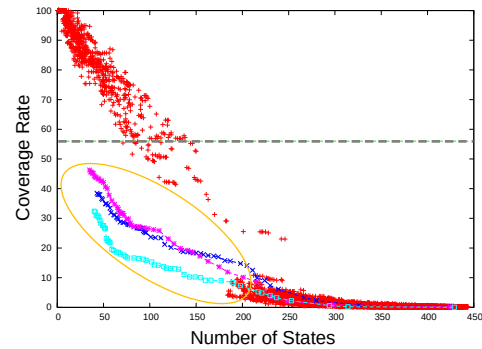


Figure 4: Three RPNI learning trajectories for a target concept of coverage=56%. Their extremity is outlined in the oval on the left. The dotted horizontal line corresponds to the coverage of the target concept. The cloud of points corresponds to random trajectories.

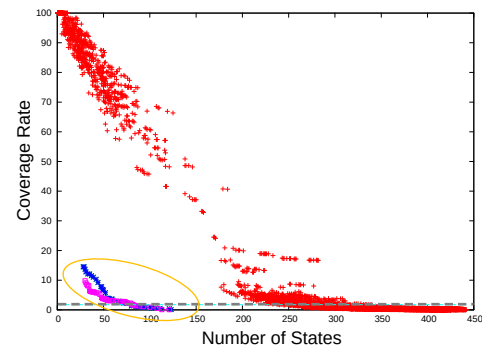


Figure 5: Same as in figure 4, except for the coverage of the target concept, here 3%.

4.3 Generalization error

Table 1, obtained for different sizes of the target automata and for training sets of structural completeness above 40%, confirms that both RPNI and RED-BLUE return overgeneralized hypotheses. On one hand, their average coverage is vastly greater than the coverage of the target automata, on the other hand, they tend to cover only part of the positive test instances, while they cover a large proportion of the negative test instances. This shows that the heuristics used in both RPNI and RED-BLUE may be inadequate for target concepts of low coverage.

5 Conclusion

This research has extended the Phase Transition-based methodology [Botta *et al.*, 2003] to the Grammatical Inference framework. Ample empirical evidence shows that the search landscape presents significant differences depending on the search operators that are considered.

Algo.	Q_c	$ucov_c$	Q_f	$ucov_f$	$pcov_f$	$ncov_f$
RB	15	5.97	10.38	33.81	60.93	34.69
RB	25	4.88	12.77	40.35	62.68	37.87
RB	50	4.2	14.23	45.38	66.14	42.23
RB	100	3.39	13.13	30.35	42.81	28.69
RPNI	15	5.95	5.14	22.9	57.51	26.99
RPNI	25	4.7	7.56	23.07	56.38	25.98
RPNI	50	3.87	14.08	23.45	51.89	24.42
RPNI	100	3.12	26.41	23.151	50.12	24.40

Table 1: Performances of RED-BLUE (RB) and RPNI for target DFA of sizes $Q = 15, 25, 50$ and 100 states. Q_f , $ucov_f$, $pcov_f$ and $ncov_f$ respectively denote the average size of the learned automata, their average coverage, the true positive and the false positive rates.

A first result is that random search appears to be more difficult in the DFA generalization cone than in the whole search space: a large gap was found, in terms of hypothesis coverage and size. This remark explains why sophisticated search biases are needed for grammatical inference algorithms in the problem range corresponding with the hole in the generalization cone.

A second finding regards the limitations of the search operators in RPNI and RED-BLUE, especially outside the region of the *Abbadingo* target concepts. Experiments with artificial learning problems built from target concepts with coverage less than 10% reveal that RPNI and RED-BLUE alike tend to learn overly general hypotheses; with respect to both the size (estimated by the number of states) and the coverage of the hypotheses, often larger by an order of magnitude than that of the target concept. What is even more worrying, is that this overgeneralization *does not imply* that the found hypotheses are complete: quite the contrary, the coverage of the positive examples remains below 65%, in all but one setting.

The presented study opens several perspectives for further research. First, it suggests that the learning search, and especially the stopping criterion, could be controlled using a hyper-parameter: the coverage rate of the target concept (possibly supplied by the expert, or estimated e.g. by cross-validation). In other words, the stopping criterion of the algorithms might be reconsidered. Secondly, more conservative generalisation operators will be investigated. Preliminary experiments done with e.g. reverted generalisation (same operator as in RPNI, applied on the reverted example strings) show that such operators can delay the determinisation cascade, and offer a finer control of the final coverage rate of the hypotheses.

Finally, the main claim of the paper is that the phase transition framework can be used to deliver indications regarding when and where specific search biases might fail – hopefully leading to understand and ultimately alleviate their limitations.

Acknowledgments

The last two authors are partially supported by the PASCAL Network of Excellence IST-2002-506 778.

References

- [Botta *et al.*, 2003] Marco Botta, Attilio Giordana, Lorenza Saitta, and Michèle Sebag. Relational learning as search in a critical region. *Journal of Machine Learning Research*, 4:431–463, 2003.
- [Cheeseman *et al.*, 1991] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the Really Hard Problems Are. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, IJCAI-91, Sydney, Australia*, pages 331–337, 1991.
- [Coste, 1999] Francois Coste. State merging inference of finite state classifiers. Technical report, Irisa, May 1999.
- [Dupont *et al.*, 1994] Pierre Dupont, Laurent Miclet, and Enrique Vidal. What Is the Search Space of the Regular Inference? In *Proceedings of the Second International Colloquium on Grammatical Inference and Applications, ICGI-94*, pages 25–37, 1994.
- [Giordana and Saitta, 2000] Attilio Giordana and Lorenza Saitta. Phase transitions in relational learning. *Machine Learning*, 41:217–251, 2000.
- [Gold, 1967] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.
- [Hogg *et al.*, 1996] Thomas Hogg, Bernard Huberman, and C. Williams. Phase transitions and the search problem. *Artificial Intelligence*, 81:1–15, 1996.
- [Lang *et al.*, 1998] Kevin Lang, Barak Pearlmutter, and Rodney Price. Results of the abbadingo one dfa learning competition and a new evidence driven state merging algorithm. In *Fourth International Colloquium on Grammatical Inference (ICGI-98)*, volume LNCS-1433, pages 1–12. Springer Verlag, 1998.
- [Muggleton and Raedt, 1994] Stephen Muggleton and Luc De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–679, 1994.
- [Oncina and Garcia, 1992] J. Oncina and P. Garcia. Inferring regular languages in polynomial update time. *Pattern Recognition and Image Analysis*, pages 49–61, 1992.
- [Pitt, 1989] Leonard Pitt. Inductive inference, dfas, and computational complexity. In *Proceedings of the Workshop on Analogical and Inductive Inference (AII-89)*, volume LNCS-397, pages 18–44. Springer Verlag, 1989.
- [Sakakibara, 1997] Yasubumi Sakakibara. Recent advances of grammatical inferences. *Theoretical Computer Science*, 185:15–45, 1997.
- [Vapnik, 1995] Vladimir Vapnik. *The Nature of Statistical Learning*. Springer, 1995.

Learning against opponents with bounded memory *

Rob Powers

Computer Science Department
Stanford University
Stanford, CA 94305
powers@cs.stanford.edu

Yoav Shoham

Computer Science Department
Stanford University
Stanford, CA 94305
shoham@cs.stanford.edu

Abstract

Recently, a number of authors have proposed criteria for evaluating learning algorithms in multi-agent systems. While well-justified, each of these has generally given little attention to one of the main challenges of a multi-agent setting: the capability of the other agents to adapt and learn as well. We propose extending existing criteria to apply to a class of adaptive opponents with bounded memory. We then show an algorithm that provably achieves an ϵ -best response against this richer class of opponents while simultaneously guaranteeing a minimum payoff against any opponent and performing well in self-play. This new algorithm also demonstrates strong performance in empirical tests against a variety of opponents in a wide range of environments.

1 Introduction

Recent work in multi-agent learning has put forth a number of proposals for judging algorithms [Bowling and Veloso, 2002; Conitzer and Sandholm, 2003; Powers and Shoham, 2005]. In addition to arguing the merits of their proposal, each researcher also demonstrated an algorithm meeting their criteria. Unfortunately, the algorithms and even the criteria themselves are in general applicable only within a very limited setting. In particular, there has been a focus on designing algorithms that behave well in the presence of stationary opponents, dodging the complexities that arise when the opponent may be adapting to the agent's past play.

The two criteria proposed in [Bowling and Veloso, 2002] require that the agent both converge to a stationary policy against some class of opponents and that the agent play a best response if the opponent converges to a stationary policy. If these criteria are satisfied by all the players, this results in a guarantee of ultimately repeatedly playing a Nash equilibria of the stage game. They then propose an algorithm that provably meets these criteria in two player normal form games with two actions per player. [Conitzer and Sandholm, 2003] adopt a restatement of these same criteria and prove

that their own algorithm achieves these criteria in arbitrary repeated games. Note however that neither of these algorithms makes any guarantee about the payoffs achieved by their algorithm against non-stationary opponents and can potentially be exploited arbitrarily by adaptive opponents, as shown in [Chang and Kaelbling, 2002].

In recent work, [Bowling, 2005] addresses this vulnerability by adding a requirement that the agent experience zero average regret. In this context, regret is traditionally defined as the maximum payoff that could have been achieved by playing any stationary policy against the opponent's entire history of actual moves minus the actual payoff the agent received. Several algorithms have been proven to achieve at most zero regret in the limit (see [Hart and Mas-Colell, 2000] and [Jafari *et al.*, 2001] for examples in both game theory and AI).

The work of [Fudenberg and Levine, 1995] on 'universal-consistency' is representative of this literature and also points out two limitations of the regret minimization approach as a whole. The first is the inability of no-regret strategies to capitalize on simple patterns in the opponent's play. They address this limitation with a proposal for the stronger concept of 'conditional consistency' and a new algorithm that achieves it in [Fudenberg and Levine, 1999]. The second limitation is that while a no-regret algorithm guarantees a minimum payoff against any possible opponent, it ignores the possibility that the sequence of moves played by the opponent is dependent on the agent's own moves. While this assumption is quite justified in games with a large number of players, it becomes a serious liability in repeated interactions with only a few players. While we are not aware of much work dealing explicitly with this limitation, [de Farias and Megiddo, 2004] address it in the design of their experts algorithm and the rational learning approach of [Kalai and Lehrer, 1993] can in principle handle adaptive algorithms of arbitrary complexity as long as they are assigned positive probability in the prior.

To see how the failure to consider adaptive opponents could hurt an algorithm's performance, let us consider a repeated version of the Prisoner's Dilemma game shown in Figure 1. Prisoner's Dilemma has been extensively studied [Axelrod, 1984] and numerous algorithms proposed that allow two agents to cooperate on the advantageous cooperation outcome without being exploited. The simplest but perhaps most effective of these is the Tit-for-Tat algorithm. Tit-for-Tat starts by cooperating and thereafter repeats whatever action

*This work was supported in part by Air Force Grant F30602-00-2-0598 and by NSF Grant IIS-0205633

the opponent played last. Note that any approach that considers only stationary opponents must always play *Defect*, since this is the unique best response to any stationary opponent and the only strategy that can ever result in no-regret performance. Against Tit-for-Tat this results in a payoff of 1, but the strategy of always playing *Cooperate* would yield a payoff of 3. Clearly, a no-regret policy is not the best response in this richer strategy space.

As another example of the advantages of considering adaptive opponents, consider playing the Stackelberg game of Figure 1 repeatedly. Notice that *Up* is a strictly dominated strategy, regardless of what the opponent chooses the row agent would prefer to play *Down*. However, if the opponent is learning, this would presumably prompt them to play *Left*, resulting in a payoff of 2 for the row agent. If it instead played the seemingly suboptimal action of *Up*, the opponent may learn to play *Right*, giving the row agent a higher payoff of 3. We can see that in this instance, teaching can play as much of a role in achieving a desirable outcome as learning. In both of these games some of the most successful strategies are those that have the ability to either cooperate with their opponents or manipulate their opponents as appropriate.

The weaknesses of this reliance on an assumption of stationarity were previously acknowledged in [Powers and Shoham, 2005] and we proposed the following three criteria:¹

Targeted Optimality: *Against any member of the target set of opponents, the algorithm achieves within ϵ of the expected value of the best response to the actual opponent.*

Compatibility: *During self-play, the algorithm achieves at least within ϵ of the payoff of some Nash equilibrium that is not Pareto dominated by another Nash equilibrium.*

Safety: *Against any opponent, the algorithm always receives at least within ϵ of the security value for the game.*

One of the key aspects of the proposal is the use of a parameterized target class of opponents against which to achieve optimal performance. While this could also address adaptive agents, the previous work only provides an algorithm for stationary opponents. In this work, we adopt the same criteria and analyze how to develop algorithms that behave well against opponents that can adapt to their past experience.

2 Environment

Within this paper, we will focus on the class of two-player repeated games with average reward. In this setting the two players repeatedly play a simultaneous move normal form game, represented as a tuple, $G = (n, A, R_{1...n})$, where n is the number of players, $A = A_1 \times \dots \times A_n$, where A_i is the set of actions for player i , and $R_i : A \rightarrow \mathbb{R}$ is the reward function for agent i . After each round, the agents accumulate their reward from the joint outcome and get to observe the prior actions of the other agent. Each agent is assumed to be trying to maximize its average reward. Two example normal form games with the rewards organized into a table are shown in Figure 1. For our purposes we assume that the

full game structure and payoffs are known to both agents. Finally, although we shall generally refer to the other player as the opponent, we do not mean to imply an adversarial setting, but instead consider the full space of general-sum games.

	<i>Cooperate</i>	<i>Defect</i>		<i>Left</i>	<i>Right</i>
<i>Cooperate</i>	3, 3	0, 4	<i>Up</i>	1, 0	3, 2
<i>Defect</i>	4, 0	1, 1	<i>Down</i>	2, 1	4, 0

(a) Prisoner's Dilemma (b) Stackelberg Game

Figure 1: Example stage games. The row player's payoff is given first, with the column player's payoff following.

3 Adaptive Opponents

While the goal of this work is to expand the set of possible opponents against which we can achieve a best response, we will need to limit their capabilities in some way. If we consider opponents whose future behavior can depend arbitrarily on the entire history of play, we lose the ability to learn anything about them in a single repeated game, since we will only ever see a given history once. We will therefore assume a limit on the opponent's ability to condition on the history. We propose directly limiting the amount of history available, by requiring that the opponents play a conditional strategy where their actions can only depend on the most recent k periods of past history, $F_i : O_{-1} \times \dots \times O_{-k} \rightarrow \Delta A_i$, where O_{-t} is the outcome of the game t periods ago. We will assume that the opponents have a default past history they assume at the start of the game. Note that even this simple model allows us to capture many methods such as Tit-for-Tat that current approaches are unable to properly handle.

Let's now consider how we could apply the criteria given above to this set of opponents. While the value of the best response to a given conditional strategy is well-defined, it would prove an unreasonable requirement for many possible strategies. Let's again consider the Prisoner's Dilemma game with an opponent that is either playing the grim strategy or always plays *Cooperate*. In the grim strategy the opponent initially starts playing *Cooperate* but switches to playing *Defect* indefinitely if its opponent ever plays *Defect* (a conditional strategy with history 1). Note that no possible learning strategy can achieve the value of the best response against both opponents, since it must play *Defect* at least once to distinguish them, at which point the option of always cooperating with the grim strategy will no longer exist. There are two possible approaches to remedying this problem. One way is to constrain the class of opponents we consider. Sufficient requirements for conditional strategies are that either the opponent only condition on the actions of the agent, not its own past actions, or that the policies the opponent plays assign non-zero probability to each action for every past history. An alternative approach would be to relax our best response target. Instead of requiring the agent achieve the best value possible by any strategy played from the start of the game, we can set the target to be the highest average value that can be achieved after any arbitrary initial sequence of moves to account for the need for exploration.

¹These three criteria were additionally required to hold for any choice of ϵ with probability at least $1 - \delta$ after a polynomial period of learning at the start of the game.

Even with these restrictions on our target, we can see that in order to guarantee an ϵ -best response with high probability we will require an exponential exploration period, since to find a good outcome an agent needs to sample from the exponential number of histories the opponent considers. Furthermore, if we allow the opponent to condition on its own actions, the number of observations required can become unbounded unless we add a requirement of a minimum probability of playing any given action.²

4 A Manipulative Algorithm

Although the MetaStrategy algorithm we introduced in [Powers and Shoham, 2005] is only explicitly designed for stationary opponents, we can use much of the intuition behind the approach to design a new algorithm for our class of adaptive opponents. The idea behind MetaStrategy is to start with a teaching strategy for an initial coordination/exploration phase and use its payoff and the opponent's play to determine which of three possible strategies to play. If its opponent is consistent with the target class it adopts a best response. If it achieves its target value by getting its opponent to adopt a best response to its teaching strategy it continues playing it, otherwise it selects a default strategy.³ The algorithm then plays according to this strategy as long as it exceeds its security level, reverting to a maximin policy if its payoff drops.

In figure 2, we show the implementation of this general approach for our target class. The MemBR strategy calculates a best response strategy against conditional strategies. This approach maintains counts of the opponent's actions after each history of length k , which it uses to calculate the cycle of agent actions with the highest expected reward out of all possible unique agent action sequences (those that don't contain a length k repeated subsequence). Given sufficient observations, this lets us guarantee that we achieve an ϵ -best response against any member of our target opponent set.⁴ We can calculate the probability the opponent's play is consistent with our target set by comparing the observed distribution of play for each history at separate times and measuring the deviation in action profiles. We continue to use the minimax strategy to achieve the security value guarantee. And for the self-play guarantee we can replace the Bully-Mixed strategy with a generous stochastic version of Godfather [Littman and Stone, 2001]. Godfather was motivated by the folk theorem for repeated games and selects some outcome in the game matrix with greater payoff for each agent than their security

²To see this, consider a conditional strategy that always plays *Defect* in Prisoner's Dilemma if its opponent played *Defect* the previous move, but plays *Cooperate* with δ probability if its opponent played *Cooperate* and it played *Defect*, and always plays *Cooperate* if both it and the agent played *Cooperate*. The agent will require a number of observations proportional to $1/\delta$ in order to distinguish this opponent from one that always plays *Defect*.

³In MetaStrategy this was Fictitious Play [Brown, 1951], but note that by instead using a no-regret policy, we could have achieved a stronger payoff guarantee against many opponents.

⁴This implementation is suitable for conditional strategies that only depend on the agent's actions. For general conditional strategies we need to consider the full space of deterministic conditional strategies to find a best response.

```

Set strategy = StochGodfather
for  $\tau_1$  time steps, Play strategy
for  $\tau_2$  time steps
  if ( $AvgValue < V_{Godfather} - \epsilon_1$ )
    With probability  $p$ ,
      set strategy = MemBR
  Play strategy
if opponentInTargetSet()
  for  $\tau_1$  time steps, Play MemBR
  if opponentInTargetSet()
    Set bestStrategy = MemBR
  else set bestStrategy = strategy
else if (strategy == StochGodfather
  AND  $AvgValue > V_{Godfather} - \epsilon_1$ )
  Set bestStrategy = StochGodfather
else set bestStrategy = MemBR
while not end of game
  if  $AvgValue < V_{security} - \epsilon_0$ 
    Play maximin strategy
  else
    Play bestStrategy

```

Figure 2: The Manipulator algorithm

values and plays its portion of the target outcome. If the opponent ever plays an action other than the matching action for the target outcome, the agent plays a strategy that forces the opponent to achieve no more than its security value until the opponent again plays its target action. For our purposes we've created a stochastic version of Godfather that selects a mixed strategy for the agent and a target action for the opponent such that the joint strategy gives the opponent a higher expected value than its security value and also denies it any advantageous deviations. This is necessary since we want our godfather algorithm to be implementable by a conditional strategy with history 1. Because of this constraint, we need to make sure the opponent can't achieve a net profit by deviating one turn and then playing the target action the next, incurring only one period of punishment. The parameters specified in the algorithm are a function of the desired values for ϵ and δ in the theoretical guarantees. (Our empirical results used $\tau_1 = 20000$, $\tau_2 = 60000$, $p = 0.005\%$, $\epsilon_0 = \epsilon_1 = 0.025$)

An additional advantage of reusing our existing framework is that we can apply the proof used for the MetaStrategy algorithm with only minor modifications to show Theorem 1. The main change is that we must require that the agent play fully mixed strategies during the beginning of the game in order to get sufficient observations to test whether the opponent's play was consistent with the target set of strategies. Given this constraint, the proof consists mainly of a long string of applications of the Hoeffding inequality [Hoeffding, 1956].

Theorem 1 *Our algorithm, Manipulator, satisfies the three properties stated in the introduction for the class of conditional strategies with bounded memory k , after a training period depending on $\frac{|A|^k}{\lambda}$, where λ is the minimum probability the opponent assigns to any action, or $\lambda = 1$ for opponents that condition only on the agent's actions.*

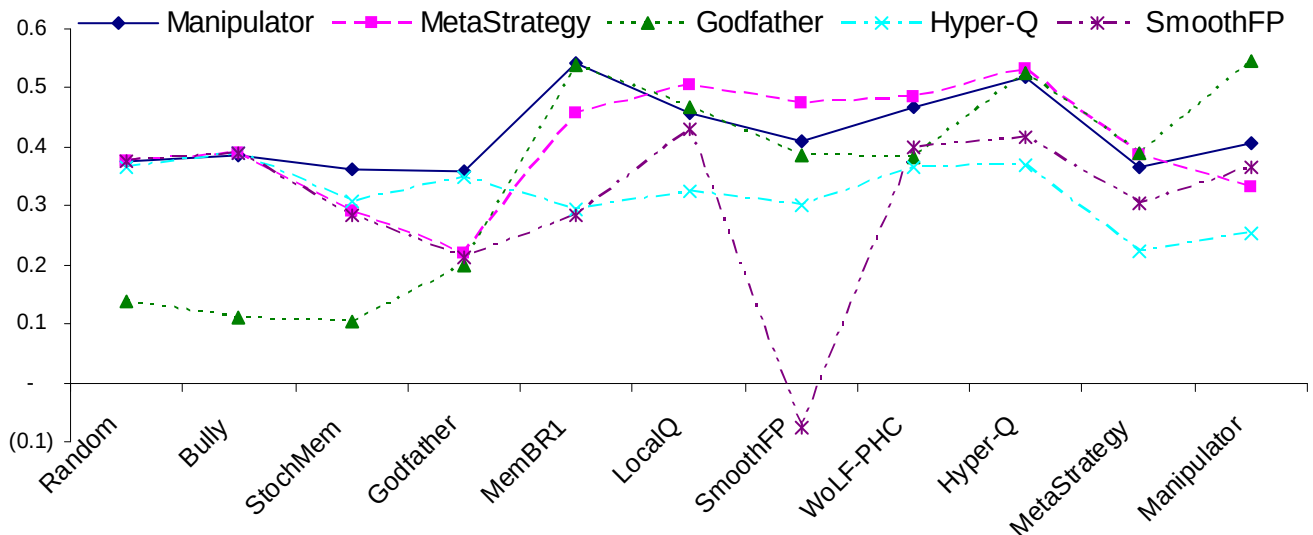


Figure 3: Average value for last 20K rounds (of 200K) across selected games in GAMUT. Game payoffs range from -1 to 1.

5 Experimental Results

In order to test the performance of our approach we’ve used the comprehensive testing environment detailed in [Powers and Shoham, 2005; Nudelman *et al.*, 2004]. Besides MetaStrategy, our set of opponents includes Bully and Godfather from [Littman and Stone, 2001], Hyper-Q [Tesauro, 2004], Local Q-learning [Watkins and Dayan, 1992], smooth fictitious play [Fudenberg and Levine, 1995], and WoLF-PHC [Bowling and Veloso, 2002]. We also include random stationary strategies (Random), random conditional strategies (StochMem), and MemBR1 which learns a best response to conditional strategies with history 1.

In Figure 3 we show the performance of the most successful of the distinct algorithms across a variety of normal form games. All of the adaptive algorithms fare well against the stationary opponents, Random and Bully, while Manipulator and to a lesser degree, Hyper-Q, fare the best against the bounded memory adaptive strategies, Godfather and StochMem. Manipulator and Godfather also have an advantage against opponents that learn a best response to conditional strategies, such as MemBR1 and Manipulator itself. For other approaches that fall outside the target sets of either MetaStrategy or Manipulator we can see that MetaStrategy has a slight advantage, mainly because of its ability to play pure strategies, while Manipulator is constrained to explore the opponent’s strategy space during the initial coordination period. However, an additional advantage of Manipulator in this setting is its ability to perform well in self-play, achieving the highest payoff of all algorithms tested, while MetaStrategy is limited to the space of stationary policies and misses more complex opportunities for cooperation in some games.

Let us now turn to the performance of our new algorithm in different types of games. Figure 4 shows the relative reward achieved by the most successful algorithms for a selection of games in GAMUT averaged across the set of opponents. We can see that Manipulator has the best performance in nearly every game. In games like Prisoner’s Dilemma and Hawk

And Dove, Godfather is able to perform better by manipulating more of the opponents into yielding, both by sending a clearer message, since it doesn’t have to do any exploration, and by waiting longer for its opponent to adapt. This stubbornness, however, proves its undoing in games where it is critical to adapt to the other opponent, such as the Dispersion Game. MetaStrategy’s strong performance in Shapley’s Game seems to stem from its default Fictitious Play strategy exploiting LocalQ and WoLF-PHC. However, we can see the advantage of Manipulator over MetaStrategy in games like Prisoner’s Dilemma and Travelers Dilemma which have equilibria in the space of repeated game strategies that Pareto-dominate any equilibria of the stage game.

6 Discussion

Although Manipulator demonstrates consistent performance across a wide variety of games, we are by no means claiming that it would be the best approach for all settings. In particular, it doesn’t fare nearly as well in the most adversarial games, like MatchingPennies, Rochambeau, and Shapley’s Game. This is not surprising since it will be unable to find any deals to offer with its Godfather component and its model-based assumption that its opponent is a conditional strategy offers no particular advantage against other adaptive opponents. An alternate approach we considered was adapting a model-free algorithm such as Q-learning [Watkins and Dayan, 1992] to the multi-agent setting, following in the footsteps of numerous previous researchers attempting to find effective multi-agent learning strategies (e.g. [Littman, 1994; Claus and Boutilier, 1998; Tesauro, 2004]). In traditional Q-learning the algorithm learns values for each action at each possible state of the world and then chooses the action that maximizes its expected reward. We propose two possible alternatives for dealing explicitly with adaptive opponents. One method is to incorporate the recent history into the state of the world and learn action values for each possible recent history. A second approach is to instead learn values over sequences

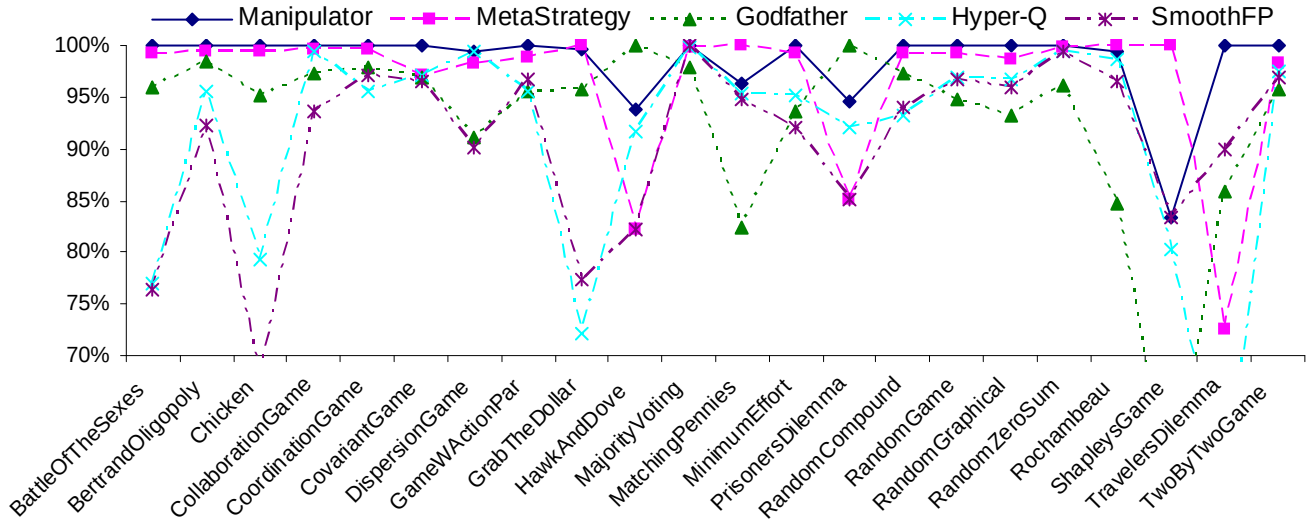


Figure 4: Percent of maximum value for last 20K rounds (of 200K) averaged across all opponents for selected games in GAMUT. The rewards were divided by the maximum reward achieved by any agent to make visual comparisons easier.

of actions. When we conducted tests for these two algorithms we found that the approach that conditioned on previous history demonstrated the ability to exploit some of the other adaptive approaches including: Local Q-learning, SmoothFP, and Hyper-Q, resulting in a higher average value in zero-sum games than any of the previous approaches. However, neither of these model-free methods performed as well in general sum games and both were dominated by Manipulator against each opponent when tested on the full set of games. Although these approaches also lack the theoretical guarantees of Manipulator, they profit from not requiring any advance knowledge about the game payoffs. Additionally, the ability to identify settings where individual approaches are particularly effective may lead to more powerful methods using portfolios of algorithms as suggested in [Leyton-Brown *et al.*, 2003].

Finally, it should be pointed out that the model we’ve put forth for modelling opponents with bounded capabilities is only one of many possible ones. A common approach used in the literature on bounded rationality [Neyman, 1985; Papadimitriou and Yannakakis, 1994] is to assume the agents can be modelled by finite automata with k states. Note that the automata model is more comprehensive than the set of conditional strategies since any conditional strategy opponent with bounded memory can be modelled by an automata with $|A|^k$ states if we allow stochastic outputs, but there exist automata that cannot be modelled by any function on a finite fixed history. In the case of automata with deterministic transitions, we can modify our Manipulator algorithm to handle this new class by implementing our version of Godfather as a DFA and replacing the best response function. Note that learning a best response to an opponent modelled by an unknown finite automata is equivalent to finding the best policy for an unknown POMDP, investigated in [Chrisman, 1992; Nikovski and Nourbakhsh, 2000]. While a difficult computation problem, we should be able to achieve the same theoretical properties for this alternate set of opponents given similar constraints to those we placed on the conditional strategies.

In particular, we need to consider how to handle finite automata with multiple ergodic sets of states and automata with arbitrarily small transition probabilities.

7 Conclusions and Future Work

We feel that explicitly addressing the issue of adaptive opponents is a critical element of learning in multi-agent systems. It is this very quality that seems to define the difference between the multi-agent setting and the single-agent one. Our algorithm approaches this setting by combining a teaching approach which manipulates adaptive opponents into playing to our agent’s advantage with a cooperative/learning approach that adapts itself to its best estimate of its opponent’s strategy. Our algorithm can be shown to achieve ϵ -optimal average reward against a non-trivial class of adaptive opponents while simultaneously guaranteeing a minimum payoff against any opponent and performing well in self-play, all with high probability. These results translate into good empirical performance in a wide variety of environments. There is clearly more work that can be done, however. As we discussed in the previous section, we’ve so far only analyzed one possible model for adaptive opponents with bounded memory and are still considering how best to incorporate other approaches that achieve better empirical performance against existing algorithms. Additionally, our approach still has significant restrictions on the set of environments it considers. We can immediately identify five limitations of our current approach:

1. Single Opponent: The criteria are only clearly defined for games with two players.
2. Single State: The criteria are only clearly defined for repeated games (rather than general stochastic games).
3. Average Reward: The criteria are defined for games in which the agent only cares about the average of its aggregated rewards (rather than a discounted sum).

4. Full Observability: The agent needs perfect observations of the opponent's actions from prior moves in the game.
5. Known Games: The algorithm needs to know all of the payoffs for each agent from the beginning of the game.

While some of these would only require minor modifications or transformations of the environment, others such as the discounted reward setting require a markedly different way of viewing the problem. We're currently working to test the limits of how much we can relax each of these restrictions in turn and hope our work here may serve as a first step towards a more widely applicable approach.

References

- [Axelrod, 1984] Robert Axelrod. *The Evolution of Cooperation*. Basic Books, New York, 1984.
- [Bowling and Veloso, 2002] Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.
- [Bowling, 2005] Michael Bowling. Convergence and no-regret in multiagent learning. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.
- [Brown, 1951] George Brown. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*. John Wiley and Sons, New York, 1951.
- [Chang and Kaelbling, 2002] Yu-Han Chang and Leslie Pack Kaelbling. Playing is believing: The role of beliefs in multi-agent learning. In *Advances in Neural Information Processing Systems 14*, pages 1483–1490, 2002.
- [Chrisman, 1992] Lonnie Chrisman. Reinforcement learning with perceptual aliasing: The perceptual distinctions approach. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 183–188, 1992.
- [Claus and Boutilier, 1998] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, 1998.
- [Conitzer and Sandholm, 2003] Vincent Conitzer and Thomas Sandholm. Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. In *Proceedings of the 20th International Conference on Machine Learning*, pages 83–90, 2003.
- [de Farias and Megiddo, 2004] Daniela Pucci de Farias and Nimrod Megiddo. How to combine expert (or novice) advice when actions impact the environment. In *Advances in Neural Information Processing Systems 16*, 2004.
- [Fudenberg and Levine, 1995] Drew Fudenberg and David Levine. Universal consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065–1089, 1995.
- [Fudenberg and Levine, 1999] Drew Fudenberg and David Levine. Conditional universal consistency. *Games and Economic Behavior*, 29:104–130, 1999.
- [Hart and Mas-Colell, 2000] Sergiu Hart and Andreu Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68:1127–1150, 2000.
- [Hoeffding, 1956] Wassily Hoeffding. On the distribution of the number of successes in independent trials. *Annals of Mathematical Statistics*, 27:713–721, 1956.
- [Jafari et al., 2001] Amir Jafari, Amy Greenwald, David Gondek, and Gunes Ercal. On no-regret learning, fictitious play, and nash equilibrium. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 226–223, 2001.
- [Kalai and Lehrer, 1993] Ehud Kalai and Ehud Lehrer. Rational learning leads to nash equilibrium. *Econometrica*, 61(5):1019–1045, 1993.
- [Leyton-Brown et al., 2003] Kevin Leyton-Brown, Eugene Nudelman, Galen Andrew, Jim McFadden, and Yoav Shoham. A portfolio approach to algorithm selection. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.
- [Littman and Stone, 2001] Michael Littman and Peter Stone. Implicit negotiation in repeated games. In *Proceedings of The Eighth International Workshop on Agent Theories, Architectures, and Languages*, pages 393–404, 2001.
- [Littman, 1994] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163, 1994.
- [Neyman, 1985] Abraham Neyman. Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. *Economic Letters*, pages 227–229, 1985.
- [Nikovski and Nourbakhsh, 2000] Daniel Nikovski and Illah Nourbakhsh. Learning probabilistic models for decision-theoretic navigation of mobile robots. In *Proceedings of the International Conference on Machine Learning*, pages 266–274, 2000.
- [Nudelman et al., 2004] Eugene Nudelman, Jenn Wortman, Kevin Leyton-Brown, and Yoav Shoham. Run the gamut: A comprehensive approach to evaluating game-theoretic algorithms. *AAMAS*, 2004.
- [Papadimitriou and Yannakakis, 1994] Christos H. Papadimitriou and Mihalis Yannakakis. On complexity as bounded rationality. In *STOC-94*, pages 726–733, 1994.
- [Powers and Shoham, 2005] Rob Powers and Yoav Shoham. New criteria and a new algorithm for learning in multi-agent systems. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.
- [Tesauro, 2004] Gerald Tesauro. Extending q-learning to general adaptive multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 16, 2004.
- [Watkins and Dayan, 1992] Chris Watkins and Peter Dayan. Technical note: Q-learning. *Machine Learning*, 8(3/4):279–292, May 1992.

ROCCER: an Algorithm for Rule Learning Based on ROC Analysis

Ronaldo C. Prati

Institute of Mathematics and Computer Science
University of São Paulo
São Carlos (SP), Brazil
prati@icmc.usp.br

Peter A. Flach

Department of Computer Science
University of Bristol
Bristol, United Kingdom
Peter.Flach@bristol.ac.uk

Abstract

We introduce a rule selection algorithm called ROCCER, which operates by selecting classification rules from a larger set of rules – for instance found by Apriori – using ROC analysis. Experimental comparison with rule induction algorithms shows that ROCCER tends to produce considerably smaller rule sets with compatible Area Under the ROC Curve (AUC) values. The individual rules that compose the rule set also have higher support and stronger association indexes.

1 Introduction

Classification rule learning can be defined as the process of, given a set of training examples, finding a set of rules that can be used for classification or prediction. Almost all classification rule learning algorithms belong to one of two families, namely separate-and-conquer and divide-and-conquer algorithms. The two families share a number of characteristics, most notably the assumption that the example space contains large continuous regions of constant class membership. The major differences are outlined below.

In the separate-and-conquer family of classification rule learning algorithms [Fürnkranz, 1999], the search procedure is generally an iterative greedy set-covering algorithm that on each iteration finds the best rule (according to a search criterion) and removes the covered examples. The process is repeated on the remaining examples until all examples have been covered or some stopping criterion has been met. In order to build a classifier, the rules found in each iteration are gathered to form either an ordered rule list (a decision list) or an unordered rule set. In the former case the classification is given by the first rule in the list that fires, while in the latter case the predictions of rules that fire are combined to predict a class.

This approach contrasts with the divide-and-conquer family of learning algorithms [Quinlan, 1993], where a global classifier is built following a top-down strategy by consecutive refinements of a partial theory. The result is generally expressed as a decision tree, which completely divides the instance space into non-overlapping axis-parallel hyperrectangles.

As decision tree induction necessarily builds complete disjoint models, in some complex domains with high-dimensional feature spaces these models can be quite complex. In such cases, individual rule learning algorithms may be preferable since they are capable of inducing overlapping and simpler rule sets [van den Eijkel, 2003]. However, one of the problems of set-covering rule learning is that rules are found in isolation, although they are used in the context of the others inside the classifier. This issue can pose several problems for a rule learning algorithm. First, from the learning perspective, fewer and fewer examples are available as covering progresses. In latter stages of induction, this may lead to fragmented training sets and rules with insufficient statistical support [Domingos, 1996]. Furthermore, each new rule is constructed in complete ignorance of the examples already covered by the previously induced rules. If a bad rule has been introduced in the rule set, there is no chance of finding a better rule for those examples (there is no backtracking).

In this work we present a new rule learning algorithm named ROCCER, aimed to overcome such problems. The main idea of the algorithm is to construct a convex hull in ROC space. We evaluate ROCCER on a broad set of benchmark domains from the UCI repository [Blake and Merz, 1998] and compare it with other rule induction methods. The paper is organized as follows. In Section 2 we discuss the background related to this work. Section 3 presents our proposed algorithm, and Section 4 contains the experimental evaluation. In Section 5 we make some concluding remarks.

2 Related work

Several approaches have been proposed in the literature to overcome the fragmentation problem. [Liu *et al.*, 1998] decouple the rule generation from the covering step. The basic idea is to use an association rule algorithm to gather all rules that predict the class attribute and also pass a minimum quality criterion into the rule set. Although this approach might overcome some of the problems of the separate-and-conquer approach, and its performance was reported to outperform standard rule learning algorithms in some domains, its main drawback is related to the number of generated rules. Often they considerably outnumber the examples, implying serious difficulties from a knowledge discovery point of view, since the understandability and usability of the generated model would decrease and the risk of overfit-

ting would increase. This idea has been extended in the AprioriC and AprioriSD algorithms [Javanoski and Lavrač, 2001; Kavšek *et al.*, 2003] by adding an additional filtering step to remove some of the redundant rules. However, AprioriC still tends to build large rule sets and AprioriSD has been developed mainly for subgroup discovery.

A different approach is to use weighted covering, which has been independently proposed by [Cohen and Singer, 1999] and [Weiss and Indurkha, 2000]. Instead of completely removing the examples covered by the best rule on each iteration, their weights are decreased, and in each iteration the covering algorithm concentrates on highly-weighted (i.e., infrequently covered) examples. [Lavrač *et al.*, 2004] also discuss the use of weighted covering in a subgroup discovery context. Alternative methods to remove redundant rules are based on pruning [Fürnkranz and Widmer, 1994; Cohen, 1995].

Some authors propose the use of confidence thresholds for classification. [Gamberger and Lavrač, 2000] include only rules with high confidence in the rule set. The classifier then refuses to classify a new instance if none of the rules cover it. [Ferri *et al.*, 2004] extends this idea by retraining a new classifier on the unclassified examples.

3 The ROCCER rule selection algorithm.

Our approach relies on using ROC analysis for selecting rules instead of using a classical covering algorithm. Roughly speaking, a ROC graph is a plot of the fraction of positive examples misclassified — false positive rate (fpr) — on the x axis against the fraction of positive examples correctly classified — true positive rate (tpr) — on the y axis. It is possible to plot in a ROC graph either a single rule, a classifier (formed by a rule set, or not) or even a partial classifier (formed by a subset of a rule set, for instance).

For a threshold-based classifier one can obtain several pairs of points (fpr_i, tpr_i) by varying the threshold. If we trace a line connecting those points we obtain a curve in the ROC space that represents the behaviour of the classifier over all possible choices of the respective threshold. In a rule learning context, [Fürnkranz and Flach, 2005] show that rule learning using a set covering approach can be seen as tracing a curve in ROC space. To see why, assume we have a empty rule list, represented by the point $(0,0)$ in ROC space. Adding a new rule R_j to the rule list implies a shift to the point (fpr_j, tpr_j) , where fpr_j and tpr_j is the tpr and fpr of the partial rule list (interpreted as a decision list) containing all rules already learnt including R_j . A curve can be traced by plotting all partial rule lists (fpr_j, tpr_j) , for j varying from 0 to the total number n of rules in the final rule list in the order they are learnt. A final default rule that always predicts the positive class can be added at the end, connecting the point (fpr_n, tpr_n) to the point $(1,1)$.

Our approach is based on this observation, and the fact that the points which represent the optimum thresholds lie on the upper convex hull of the ROC curve [Provost and Fawcett, 2001]. Rules come from a external larger set of rules (in our implementation, we use the Apriori association rule learning algorithm, fixing the head of the rules to each of the possible

class values, allowing us to deal with multi-class problems) and we perform a selection step based on the ROC curve. The basic idea is to only insert a rule in the rule list if the insertion leads to a point outside the current ROC convex hull (the current ROC convex hull is the upper convex hull of the rules that are already in the rule list). Otherwise the rule is discarded. For a better understanding of how our algorithm works, we first describe it using an example.

Rules are selected separately for each class, and are kept in an ordered rule list. Let's label the class we are selecting rules for **positive**; the label **negative** represents the (conjunction of) examples in the other class(es). First, we initialize the rule list with a default rule, $R_{default}$, which always predicts positive. The current ROC convex hull is formed then by 2 points, $(0,0)$ and $(1,1)$ which means “ignore the default rule (classify everything as negative) or use the default rule (classify everything as positive)”. We use fpr_{R_i} , tpr_{R_i} to refer to the rule R_i 's true and false positive rates, and fpr_i , tpr_i to refer to a point i in the ROC curve, representing the corresponding rule list's true and false positive rates. Suppose now we are inserting a new rule R_1 . As the actual convex hull is formed only by the line segment $(0,0) - (1,1)$, R_1 will only be inserted if the point formed by the rule's (fpr_{R_1}, tpr_{R_1}) is above the convex hull. Let's say R_1 is inserted. The current convex hull is then updated, and contains the points $(0,0), (fpr_1, tpr_1), (1,1)$, where $fpr_1 = fpr_{R_1}$ and $tpr_1 = tpr_{R_1}$. This process is depicted in Figure 1.

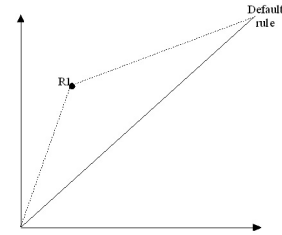


Figure 1: R_1 leads to a point outside the current convex hull (main diagonal) and is therefore inserted in the rule list.

Suppose we are now trying to insert a second rule R_2 . As we have said before, in the standard set-covering approach the learning of a new rule does not take into account the rules already learnt. In our approach, we try to overcome this issue using the ROC graph to analyze interactions among rules. We do this by comparing the rule we are trying to insert with the slopes of each of the line segments in the current convex hull in the ROC graph. In our example, if the slope of the point formed by the origin and (fpr_{R_2}, tpr_{R_2}) is above the line from the origin to the point (fpr_1, tpr_1) , we say that R_2 “improves in relation to” R_1 and insert R_2 in the rule list. This is similar to if R_2 had been learned before R_1 in the set-covering approach. By comparing with the rules which are already in the rule list, our algorithm provides a kind of backtracking. This insertion will, of course, produce changes to the other points in the ROC curve. The first non-trivial point in the ROC curve changes to (fpr_{R_2}, tpr_{R_2}) . If R_1 and R_2 do not have any overlap, the second point will be $(fpr_{R_2} + fpr_{R_1}, tpr_{R_2} + tpr_{R_1})$. If R_1 and R_2 do have some overlap, however, we should dis-

count the examples covered by both rules to calculate the second point.

If R_2 is not inserted at the first iteration, we proceed by comparing with the remaining line segments in the ROC convex hull. Before we compare with the next line segment, we update R_2 's fpr and tpr by removing the examples covered by R_1 (the examples which evaluate to true for R_1 's antecedent). In fact, we are “interpreting” R_2 as $\neg R_1 \wedge R_2$. If the updated position of R_2 is above the line from R_1 to $R_{default}$, R_2 is inserted in the rule list after R_1 . This process is depicted in Figure 2. Otherwise (since no further rules remain) R_2 is discarded. The pseudo-code of this algorithm is shown in Algorithm 1.

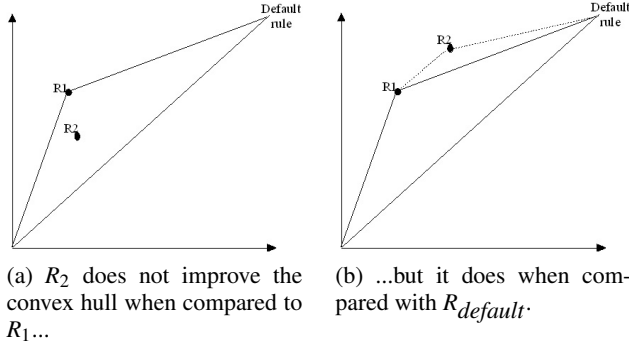


Figure 2: Finding the right point to insert R_2 .

Algorithm 1: The ROCCER algorithm.

Data: RS_{in} : A (large) rule set for a given class

Result: RS_{out} : A (smaller) rule list containing the selected rules

$RS_{out} = \{R_{default}\};$

foreach $Rule \in RS_{in}$ **do**
 TryToInsertRule($Rule$)

endfch

return RS_{out}

procedure TryToInsertRule($Rule$)

 RuleToCompare = first rule in RS_{out} ;

repeat

if $Rule$'s (fpr, tpr) is outside convex hull **then**

 Insert Rule into RS_{out} before RuleToCompare

else

 /* shift to a new origin */

 Remove all the examples from Rule which are covered by RuleToCompare;

 Actualize Rule's (fpr, tpr) ;

 RuleToCompare = next rule in RS_{out}

endif

until RuleToCompare $\neq R_{default}$;

if Rule is not inserted **then**

 Discard Rule

endif

Due to overlapping coverage among rules, this process

does not necessarily lead to a convex curve. The insertion of a new rule in the rule list may introduce concavities before or after the point of insertion. If concavities occur after an insertion, the inserted rule covers examples originally covered by subsequent rules, which decreases the latter's precision. In this case, we remove the rules where the concavity occurs. Alternatively, if the concavity occurs before the insertion point, both rules share a region where they misclassify some examples. In this case, it is unreasonable to use the partial rule list including the first rule but excluding the second, because the corresponding ROC point is under the convex hull. Therefore, we construct the disjunction of the two rules and treat it as a single rule.

In our implementation, rules presented to ROCCER are initially ordered, using the Euclidean distance to the point $(0, 1)$ in the ROC space. However, due to the possibility to remove a selected rule and the (implicit) backtracking, the order dependence in selecting rules is lower than a decision list in inducing rules. Experiments with another orderings would be an interesting issue for further work, though.

This concludes the description of the training phase. For classification, we also use a ROC-based method. Bayes' theorem states that the odds that a classifier correctly classifies an instance (posterior odds) is given by the likelihood ratio times the odds of the instance being of the predicted class (prior odds). In ROC space, the likelihood ratio can be interpreted as tpr/fpr . Recall that we selected rules separately for each class. Thus, we have a ROC convex hull for each class. To classify a new instance we also consider each class separately, and, for each class, we determine the first rule that fires in the respective ROC convex hull. This rule has an associated (tpr, fpr) in the ROC curve, which yields a likelihood ratio. The posterior odds is then converted back to a probability (for ranking), or we select the class with maximum posterior odds (for classification).

4 Experimental evaluation

In order to empirically evaluate our proposed approach, we performed an experimental evaluation using 16 data sets from UCI [Blake and Merz, 1998]. We used only data sets without missing values, as Apriori (the association rule algorithm we use to generate the rules for subsequent selection by ROCCER) can't handle them. Table 1 summarizes the data sets employed in this study. It shows, for each data set, the number of attributes (#Attrs), the number of examples (#Examples), and percentage of examples in the majority class (%MajClass) – although ROCCER can handle more than two classes, in order to calculate AUC values we restricted our experiments to two-class problems. For data sets having more than two classes, we chose the class with fewer examples as the positive class, and collapsed the remaining classes as the negative.

ROCCER's results were compared with those obtained by the following rule learning systems:

CN2 This algorithm is a classical implementation of the separate-and-conquer rule learning family. In its first version [Clark and Niblett, 1989] CN2 induces a decision list using entropy as a search heuristic. It has later

#	Data set	# Attrs	# Examples	%MajClass
1	Breast	10	683	65.00
2	Bupa	7	345	57.98
3	E.Coli	8	336	89.58
4	Flag	29	194	91.24
5	German	21	1000	70.00
6	Glass	10	214	92.07
7	Haberman	4	306	73.53
8	Heart	14	270	55.55
9	Ionosphere	34	351	64.10
10	Kr-vs-Kp	37	3196	52.22
11	Letter-a	17	20000	96.06
12	New-thyroid	6	215	83.72
13	Nursery	9	12960	97.45
14	Pima	9	768	65.10
15	Satimage	37	6435	90.27
16	Vehicle	19	846	76.48

Table 1: UCI data sets used in our experiments.

been modified to incorporate the induction of unordered rule sets and Laplace error correction as evaluation function [Clark and Boswell, 1991].

Ripper [Cohen, 1995] proposed Ripper in the *Incremental Reduced Error Pruning* (IREP) [Fürnkranz and Widmer, 1994] context. It has features such as error-based pruning and an MDL-based heuristic for determining how many rules should be learned.

Slipper This algorithm is a further improvement of Ripper which uses a weighted set-covering approach [Cohen and Singer, 1999].

C4.5 [Quinlan, 1993]’s C4.5 is almost a standard in empirical comparison of symbolic learning algorithms. It is a member of divide-and-conquer family. C4.5 uses information gain as quality measure to build a decision tree and a post-pruning step based on error reduction. We can consider each branch in a decision tree as a rule.

Ripper and Slipper were used with -a option to generate rules for both classes. CN2 was used in its two versions, ordered (CN2OR) and unordered (CN2). We also evaluated both pruned (C45) and non-pruned (C45NP) trees induced by C4.5. All other parameters were set to default values. In order to calculate the AUC values we estimated probabilities of each rule using Laplace correction. For the unordered version of CN2, probabilities were estimated using all fired rules. AUC values were estimated using the trapezoidal rule. We used [Borgelt and Kruse, 2002]’s implementation of Apriori to generate the large rule sets used by ROCCER. The parameters were set to 50% of confidence and 1/3 of the percentage of minority class as support. For ROCCER, the probabilities were estimated by the posterior odds (described in Section 3). We also compare with a bagging of all rules generated by Apriori.

We ran the experiments using 10-fold stratified cross-validation. The experiment is paired, *i.e.*, all inducers were given the same training and test files. The averaged AUC values (and respective standard deviations in brackets) are

shown in Table 2. We also perform a two-tailed Dunett multiple comparison with a control procedure¹ using ROCCER as control (the other results against ROCCER). Cells having AUC values statistically better than ROCCER are represented in dark gray while light gray is used to represent cells statistically worse than ROCCER, both with 95% confidence level.

Table 2 shows relatively few statistically significant differences. Comparing against C4.5, ROCCER achieved 4 wins and 1 loss. This is the same score if we compare ROCCER against Ripper. Against Slipper, the results are 5 wins and no losses. A comparison of ROCCER against C4.5 without pruning, CN2 and CN2 ordered yield 2 losses and no wins. We believe these two losses are due to the high degree of class skew in those two datasets (they are the most skewed in our study). In order to allow Apriori to find rules for both classes in these domains, the support parameter used in Apriori is very low. In these cases we have both a small number of rules generated for the minority class and a large number of rules generated for the majority class. Further improvements should be made in ROCCER to cope with such situations (for instance, it would be interesting to introduce different minimum support for each class). Taking into account all the rule learning algorithms, the score is 12 wins and 9 losses (all losses are concentrated in the two skewed domains, though). Comparing with all the generated rules, ROCCER produced 6 wins and no losses. We also compute AUC value on selecting k (the same number as ROCCER) random rules and rules with higher individual AUC values. Due to lack of space results are not shown in this paper, but they are in most of the cases significantly worst and never better than ROCCER. This indicates that ROCCER’s selection procedure is responsible for a gain of performance over all the presented rules.

The good results with both versions of CN2, and the relatively poor AUC figures for Ripper and Slipper, are worth noticing, and may be explained by the absence of pruning mechanisms. It has already been reported in the literature that non-pruned trees are better for probability prediction and thus produce higher AUC values [Provost and Domingos, 2003]. It might be expected that a similar phenomenon also would occur with algorithms from the separate-and-conquer family. Ripper and Slipper are – at least conceptually – similar to CN2 but incorporate, respectively, rule pruning and weighted coverage.

Table 3 presents the average size (in number of rules) of the rule sets for each algorithm. Size 0 means that the classifier is formed only by the default rule. The picture here is more clear. Apart from some exceptions, ROCCER produces smaller rule sets than C4.5 without pruning, CN2 (both ordered and unordered), and Slipper. On the other hand, Ripper produced (significantly) smaller rule sets in 7 out of 16 domains, and there were 8 draws and 1 win. A further investigation involving the data sets where most of the algorithms produced smaller rule sets than ROCCER (Breast, Heart, Ionosphere and Kr-vs-kp) might produce some in-

¹Multiple comparison is used to adjust the observed significance level for the fact that multiple comparisons are made. If we use a t-test, and as each comparison has up to 5% Type I error, then the Type I error rate over the entire group can be much higher than 5%.

#	ROCCER	C45	C45NP	CN2	CN2OR	Ripper	Slipper	All
1	98.63(1.88)	97.76(1.51)	98.39(1.30)	99.26(0.81)	99.13(0.92)	98.72(1.38)	99.24 (0.57)	99.07(0.87)
2	65.30(7.93)	62.14(9.91)	57.44(11.92)	62.74(8.85)	62.21(8.11)	69.10(7.78)	59.84 (6.44)	65.38(10.63)
3	90.31(11.56)	50.00(0.00)	90.06(7.75)	90.17(6.90)	85.15(11.38)	61.86(25.49)	74.78 (15.94)	16.50(10.43)
4	61.83(24.14)	50.00(0.00)	68.68(17.22)	53.22(24.12)	42.78(24.43)	45.28(14.93)	52.35 (7.44)	62.11(23.96)
5	72.08(6.02)	71.43(5.89)	67.71(4.12)	75.25(5.38)	70.90(4.70)	64.02(13.62)	71.32 (6.20)	73.37(4.84)
6	79.45(12.98)	50.00(0.00)	81.50(12.65)	73.74(15.40)	79.64(13.24)	49.75(0.79)	50.00 (2.36)	35.62(18.93)
7	66.41(11.54)	55.84(6.14)	64.33(13.58)	59.83(9.87)	59.28(10.13)	57.45(3.85)	50.40 (11.14)	66.52(5.94)
8	85.78(8.43)	84.81(6.57)	81.11(7.91)	83.61(6.89)	82.25(6.59)	84.89(7.68)	84.03 (6.36)	90.72(6.28)
9	94.18(4.49)	86.09(9.97)	90.91(6.03)	96.23(2.97)	92.18(7.54)	92.06(5.94)	93.95 (6.82)	90.14(5.32)
10	99.35(0.36)	99.85(0.20)	99.86(0.20)	99.85(0.16)	99.91(0.17)	99.85(0.21)	99.91 (0.09)	92.67(1.60)
11	96.08(0.52)	95.49(1.96)	99.33(0.46)	99.34(0.28)	99.44(0.63)	97.27(1.86)	98.82(0.44)	92.45(1.54)
12	98.40(1.70)	87.85(10.43)	97.50(3.39)	99.14(1.19)	98.43(2.58)	94.95(9.94)	99.12 (1.25)	89.97(7.75)
13	97.85(0.44)	99.42(0.14)	99.74(0.13)	100.00(0.00)	99.99(0.01)	99.43(0.26)	94.40(1.59)	97.79(0.65)
14	70.68(5.09)	72.07(4.42)	72.60(6.50)	70.96(4.62)	71.97(5.44)	68.07(9.46)	70.02 (5.97)	70.37(5.01)
15	89.39(2.38)	90.15(1.70)	91.31(1.32)	91.48(1.45)	91.48(0.90)	86.83(3.94)	89.06 (1.98)	79.62(4.95)
16	96.42(1.47)	94.76(3.00)	96.99(1.44)	97.38(2.05)	96.49(2.41)	95.01(2.22)	93.99 (3.13)	93.37(3.05)
Avg	85.13	77.98	84.84	84.51	83.2	79.03	80.08	75.98

Table 2: AUC values estimated with 10-fold cross-validation on the 16 UCI data sets described in Table 1, obtained with ROCCER, C4.5, C4.5 without pruning, CN2 unordered, CN2 ordered (*i.e.*, learning decision lists), Ripper, Slipper, and bagging all rules found by Apriori. Numbers between brackets indicate standard deviations; dark gray indicates significant wins over ROCCER, and light gray indicates significant losses against ROCCER.

sights for improvements to our approach.

We conclude from Tables 2 and 3 that ROCCER combines, in a sense, the best of both worlds: it achieves AUC values that are comparable to those of unpruned decision trees and CN2, but without the large number of rules induced by those systems. Finally, Table 4 presents statistics of the individual rules that comprise the rule sets, which demonstrates another advantage of the ROCCER approach. Support ranges from 0 to 100% and is a measure of the relative coverage of each rule. Weighted relative accuracy (WRAcc) ranges from 0 to 0.25 and assesses the significance of a rule, in terms of difference between the observed and expected numbers of true positives. The odds ratio ranges from 0 to ∞ and is a measure of strength of association. It can clearly be seen that the rules selected by ROCCER have considerably higher values for all measures. This means that the rules are more meaningful in isolation, without reference to the other rules in the rule set. Thus, ROCCER successfully overcomes one of the main drawbacks of the set-covering approach.

	Support (%)	WRAcc	Odds Ratio
ROCCER	13.67 (13.89)	0.0355 (0.018)	154.02 (337.33)
C4.5	3.73 (6.01)	0.0094 (0.013)	44.55 (77.04)
C4.5NP	1.19 (1.06)	0.0030 (0.003)	31.90 (56.68)
CN2	3.90 (2.52)	0.0110 (0.009)	98.73 (138.64)
CN2OR	3.10 (2.18)	0.0085 (0.007)	95.07 (192.94)
Ripper	5.96 (5.34)	0.0184 (0.012)	74.08 (103.88)
Slipper	1.92 (1.58)	0.0060 (0.006)	33.86 (50.41)
All	8.07 (5.06)	0.0114 (0.014)	67.39 (111.72)

Table 4: Support, weighted relative accuracy and odds ratio averaged over all learned rules.

A final word should be said regarding computational complexity. ROCCER is, of course, computationally more expensive than the other algorithms. In the worst case, the com-

plexity is $O(n^2)$, where n is the number of rules used as input. However, on average, the number of iterations is $\Omega(mn)$, where m is the number of rules selected by the algorithm. Due to lack of space we will not report runtime statistics for all data sets. For most datasets the runtime ranges from a few seconds to 10 minutes per fold (on a Pentium 4 2.4Ghz machine with 512MB of RAM). For these datasets, the number of rules used as input is up to 1,000. For some domains (Krvs-kp and Satimage) the number of rules generated by Apriori is very high (more than 40,000). The runtime in these cases is on average nearly 1.5 hours per fold.

5 Conclusion

We presented ROCCER, a rule selection algorithm based on ROC analysis. ROCCER operates by selecting rules from a larger set of rules by maintaining a ROC convex hull in the ROC space. Features of ROCCER's approach include implicit backtracking and discovery of pairs of related rules. Experimental results demonstrate AUC values that are compatible with the best probability predictors such as unpruned decision trees, achieved with considerably smaller rule sets. The rules that compose the rule sets induced by ROCCER have also higher values of support, weighted relative accuracy and odds ratio, and thus are more meaningful as individual rules.

Acknowledgments

This work is partially supported by the Brazilian Research Council CAPES (Process no. BEX1340/03-8), and was carried out while the first author was visiting the University of Bristol.

References

[Blake and Merz, 1998] C. L. Blake and C. J. Merz. UCI repository of machine learning databases,

#	ROCCER	C4.5	C4.5NP	CN2	CN2OR	Ripper	Slipper	All
1	48.4(2.32)	37.8(12.62)	104.2(9.58)	32.8(2.74)	24.3(2.71)	16.9(1.1)	36.7(10.78)	502.1(8.96)
2	3.9(0.99)	15(10.53)	143.3(13.12)	100.7(3.77)	83.6(5.87)	7.5(1.84)	29.3(8.67)	292.8(21.57)
3	6.7(0.82)	0(0)	62(10.46)	35.3(2.83)	33.6(3.13)	4.9(4.33)	15.2(6.73)	27.5(6.05)
4	1.7(2.26)	0(0)	35.7(7.41)	20.2(1.62)	13.4(2.12)	1.3(0.48)	1.6(3.53)	827.4(254.92)
5	23.7(6.75)	78.2(18.5)	388.6(19.07)	143.9(6.98)	106.8(4.37)	8.9(3.25)	37.4(12.42)	2886.1(577.3)
6	2.4(0.52)	0(0)	32.1(5.99)	21.6(1.26)	21.9(3.18)	0.2(0.42)	2(3.43)	26.7(12.1)
7	0.8(0.42)	5.6(5.72)	71.2(9.39)	75.9(3.9)	57(5.98)	3.5(0.97)	11.8(13.21)	21.2(1.4)
8	68.2(4.42)	13.2(4.49)	97.4(15.04)	42.8(2.49)	36.1(1.79)	7.7(1.25)	27.2(8.73)	1875.6(91.9)
9	67.1(5.38)	23.4(3.98)	128.4(14.1)	36.9(2.28)	22.9(1.29)	19.6(4.43)	38.8(10.43)	20855.9(2690.9)
10	43.6(7.97)	29.2(2.15)	37.4(3.5)	30.1(1.85)	28.1(1.73)	27.3(1.7)	61.2(9.87)	41790.8(565.72)
11	78.4(3.06)	183.8(15.22)	700.4(31.76)	126.3(3.56)	120.2(2.7)	71(6.58)	115.2(6.16)	649.1(20.55)
12	8.5(0.53)	4(0)	35.4(4.2)	18.7(0.67)	15.3(0.82)	10.9(3.21)	20.7(5.48)	39.1(4.72)
13	18.9(1.66)	114.6(3.1)	227.1(3.57)	112.4(2.27)	17.6(0.52)	44.1(7.32)	57.6(2.37)	100.0(4.29)
14	4(0.82)	49.4(20.27)	347.4(10)	169.7(10.08)	168.4(7.9)	8.7(2.21)	30.5(10.82)	10.7(3.62)
15	143.9(51.28)	531.1(84.63)	1767.2(111.93)	158.8(6.09)	199.8(6.99)	32(4.37)	52.3(13.9)	4451.2(712.9)
16	48.4(4.48)	89.5(12.12)	188.9(9.42)	49.8(3.29)	41.2(2.82)	23.7(4.06)	75.7(12.11)	431.4(53.85)
Avg	35.54	73.43	272.92	73.49	61.89	18.01	38.33	4674.23

Table 3: Average numbers of rules obtained with the experiments reported in Table 2.

1998. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- [Borgelt and Kruse, 2002] C. Borgelt and R. Kruse. Induction of association rules: A priori implementation. In *Proc. 15th Conf. on Computational Statistics*, 2002.
- [Clark and Boswell, 1991] P. Clark and R. Boswell. Rule induction with CN2: Some recent improvements. In *Proc. 5th European Conf. on Machine Learning*, volume 482 of *LNAI*, pages 151–163. Springer-Verlag, 1991.
- [Clark and Niblett, 1989] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3(4):261–283, 1989.
- [Cohen and Singer, 1999] W. Cohen and Y. Singer. A simple, fast and effective rule learner. In *Proc. 16th Nat. Conf. on Artificial Intelligence*, pages 335–342. AAAI/MIT Press, 1999.
- [Cohen, 1995] W. Cohen. Fast effective rule induction. In *Proc. 12th Int. Conf. on Machine Learning*, pages 115–123, 1995.
- [Domingos, 1996] P. Domingos. Unifying instance-based and rule-based induction. *Machine Learning*, 24(2):141–168, 1996.
- [Ferri et al., 2004] C. Ferri, P. Flach, and J. Hernández. Delegating classifiers. In *Proc. 21st Int. Conf. on Machine Learning*, pages 289–296, 2004.
- [Fürnkranz and Flach, 2005] J. Fürnkranz and P. Flach. ROC’n’rule learning – toward a better understanding of rule covering algorithms. *Machine Learning*, 58(1):39–77, 2005.
- [Fürnkranz and Widmer, 1994] J. Fürnkranz and G. Widmer. Incremental reduced error pruning. In *Proc. 11th Int. Conf. on Machine Learning*, pages 70–77. Morgan Kaufmann, 1994.
- [Fürnkranz, 1999] J. Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, February 1999.
- [Gamberger and Lavrač, 2000] D. Gamberger and N. Lavrač. Confirmation rule sets. In *Proc. 4th European Conf. on Principles of Data Mining and Knowledge Discovery*, volume 1910 of *LNAI*, pages 34–43. Springer-Verlag, 2000.
- [Javanoski and Lavrač, 2001] V. Javanoski and N. Lavrač. Classification rule learning with Apriori-C. In *Proc. 10th Portuguese Conf. on Artificial Intelligence*, volume 2258 of *LNAI*, pages 44–52, Porto, Portugal, 2001. Springer-Verlag.
- [Kavšek et al., 2003] B. Kavšek, N. Lavrač, and V. Javanoski. Apriori-SD: Adapting association rule learning to subgroup discovery. In *Proc. 5th Intelligent Data Analysis*, volume 2810 of *LNCS*, pages 230–241. Springer Verlag, 2003.
- [Lavrač et al., 2004] N. Lavrač, B. Kavšek, P. Flach, and L. Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.
- [Liu et al., 1998] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining*, pages 80–86, New York, USA, 1998.
- [Provost and Domingos, 2003] F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, 2003.
- [Provost and Fawcett, 2001] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.
- [Quinlan, 1993] J. R. Quinlan. *C4.5 Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.
- [van den Eijkel, 2003] G. van den Eijkel. Information-theoretic tree and rule induction. In *Intelligent Data Analysis*, pages 465–475. Springer-Verlag, 2003.
- [Weiss and Indurkha, 2000] S. M. Weiss and N. Indurkha. Lightweight rule induction. In *Proc. 17th Int. Conf. on Machine Learning*, pages 1135–1142, 2000.

Stepwise Nearest Neighbor Discriminant Analysis*

Xipeng Qiu and Lide Wu

Media Computing & Web Intelligence Lab
Department of Computer Science and Engineering
Fudan University, Shanghai, China
xpqiu,ldwu@fudan.edu.cn

Abstract

Linear Discriminant Analysis (LDA) is a popular feature extraction technique in statistical pattern recognition. However, it often suffers from the small sample size problem when dealing with the high dimensional data. Moreover, while LDA is guaranteed to find the best directions when each class has a Gaussian density with a common covariance matrix, it can fail if the class densities are more general. In this paper, a new nonparametric feature extraction method, stepwise nearest neighbor discriminant analysis(SNNDA), is proposed from the point of view of the nearest neighbor classification. SNNDA finds the important discriminant directions without assuming the class densities belong to any particular parametric family. It does not depend on the nonsingularity of the within-class scatter matrix either. Our experimental results demonstrate that SNNDA outperforms the existing variant LDA methods and the other state-of-art face recognition approaches on three datasets from ATT and FERET face databases.

1 Introduction

The curse of high-dimensionality is a major cause of the practical limitations of many pattern recognition technologies, such as text classification and object recognition. In the past several decades, many dimensionality reduction techniques have been proposed. Linear discriminant analysis (LDA) [Fukunaga, 1990] is one of the most popular supervised methods for linear dimensionality reduction. In many applications, LDA has been proven to be very powerful.

The purpose of LDA is to maximize the between-class scatter while simultaneously minimizing the within-class scatter. It can be formulated by Fisher Criterion:

$$J_F(W) = \frac{W^T S_b W}{W^T S_w W}, \quad (1)$$

where W is a linear transformation matrix, S_b is the between-class scatter matrix and S_w is the within-class scatter matrix.

*The support of NSF of China (69935010) and (60435020) is acknowledged.

A major drawback of LDA is that it often suffers from the small sample size problem when dealing with the high dimensional data. When there are not enough training samples, S_w may become singular, and it is difficult to compute the LDA vectors. For example, a 100×100 image in a face recognition system has 10000 dimensions, which requires more than 10000 training data to ensure that S_w is nonsingular. Several approaches[Liu *et al.*, 1992; Belhumeur *et al.*, 1997; Chen *et al.*, 2000; Yu and Yang, 2001] have been proposed to address this problem. A common problem with all these proposed variant LDA approaches is that they all lose some discriminative information in the high dimensional space.

Another disadvantage of LDA is that it assumes each class has a Gaussian density with a common covariance matrix. LDA guaranteed to find the best directions when the distributions are unimodal and separated by the scatter of class means. However, if the class distributions are multimodal and share the same mean, it fails to find the discriminant direction[Fukunaga, 1990]. Besides, the rank of S_b is $c - 1$, where c is the number of classes. So the number of extracted features is, at most, $c - 1$. However, unless a posteriori probability function are selected, $c - 1$ features are suboptimal in Bayes sense, although they are optimal with regard to Fisher criterion [Fukunaga, 1990].

In this paper, a new feature extraction method, stepwise nearest neighbor discriminant analysis(SNNDA), is proposed. SNNDA is a linear feature extraction method in order to optimize nearest neighbor classification (NN). Nearest neighbor classification [Duda *et al.*, 2001] is an efficient method for performing nonparametric classification and often used in the pattern classification field, especially in object recognition. Moreover, the NN classifier has a close relation with the Bayes classifier. However, when nearest neighbor classification is carried out in a high-dimensional feature space, the nearest neighbors of a point can be very far away, causing bias and degrading the performance of the rule [Hastie *et al.*, 2001]. Hastie and Tibshirani [Hastie and Tibshirani, 1996] proposed a discriminant adaptive nearest neighbor (DANN) metric to stretch the neighborhood in the directions in which the class probabilities don't change much, but their method also suffers from the small sample size problem.

SNNDA can be regarded as an extension of nonparametric discriminant analysis[Fukunaga and Mantock, 1983], but it

doesn't depend on the nonsingularity of the within-class scatter matrix. Moreover, SNNDA finds the important discriminant directions without assuming the class densities belong to any particular parametric family.

The rest of the paper is organized as follows: Section 2 gives the review and analysis of the current existing variant LDA methods. Then we describe stepwise nearest neighbor discriminant analysis in Section 3. Experimental evaluations of our method, existing variant LDA methods and the other state-of-art face recognition approaches are presented in Section 4. Finally, we give the conclusions in Section 5.

2 Review and Analysis of Variant LDA Methods

The purpose of LDA is to maximize the between-class scatter while simultaneously minimizing the within-class scatter.

The between-class scatter matrix S_b and the within-class scatter matrix S_w are defined as

$$S_b = \sum_{i=1}^c p_i (m_i - m)(m_i - m)^T \quad (2)$$

$$S_w = \sum_{i=1}^c p_i S_i, \quad (3)$$

where c is the number of classes; m_i and p_i are the mean vector and a priori probability of class i , respectively; $m = \sum_{i=1}^c p_i m_i$ is the total mean vector; S_i is the covariance matrix of class i .

LDA method tries to find a set of projection vectors $W \in R^{D \times d}$ maximizing the ratio of determinant of S_b to S_w ,

$$W = \arg \max_W \frac{|W^T S_b W|}{|W^T S_w W|}, \quad (4)$$

where D and d are the dimensionalities of the data before and after the transformation respectively.

From Eq.(4), the transformation matrix W must be constituted by the d eigenvectors of $S_w^{-1} S_b$ corresponding to its first d largest eigenvalues [Fukunaga, 1990].

However, when the small sample size problem occurs, S_w becomes singular and S_w^{-1} does not exist. Moreover, if the class distributions are multimodal or share the same mean (for example, the samples in (b),(c) and (d) of Figure 2), it can fail to find the discriminant direction[Fukunaga, 1990]. Many methods have been proposed for solving the above problems. In following subsections, we give more detailed review and analysis of these methods.

2.1 Methods Aimed at Singularity of S_w

In recent years, many researchers have noticed the problem about singularity of S_w and tried to overcome the computational difficulty with LDA.

To avoid the singularity of S_w , a two-stage PCA+LDA approach is used in [Belhumeur *et al.*, 1997]. PCA is first used to project the high dimensional face data into a low dimensional feature space. Then LDA is performed in the reduced PCA subspace, in which S_w is non-singular. But this method

is obviously suboptimal due to discarding much discriminative information.

Liu *et al.* [Liu *et al.*, 1992] modified Fisher's criterion by using the total scatter matrix $S_t = S_b + S_w$ as the denominator instead of S_w . It has been proven that the modified criterion is exactly equivalent to Fisher criterion. However, when S_w is singular, the modified criterion reaches the maximum value, namely 1, for any transformation W in the null space of S_w . Thus the transformation W cannot guarantee the maximum class separability $|W^T S_b W|$ is maximized. Besides, this method still needs to calculate an inverse matrix, which is time consuming. Chen *et al.* [Chen *et al.*, 2000] suggested that the null space spanned by the eigenvectors of S_w with zero eigenvalues contains the most discriminative information. A LDA method (called NLDA) in the null space of S_w was proposed. It chooses the projection vectors maximizing S_b with the constraint that S_w is zero. But this approach discards the discriminative information outside the null space of S_w . Figure 1(a) shows that the null space of S_w probably contains no discriminant information. Thus, it is obviously suboptimal because it maximizes the between-class scatter in the null space of S_w instead of the original input space. Besides, the performance of the NLDA drops significantly when $N - c$ is close to the dimension D , where N is the number of samples and c is the number of classes. The reason is that the dimensionality of the null space is too small in this situation and too much information is lost [Li *et al.*, 2003]. Yu *et al.* [Yu and Yang, 2001] proposed a direct LDA (DLDA) algorithm, which first removes the null space of S_b . They assume that no discriminative information exists in this space. Unfortunately, it be shown that this assumption is incorrect. Fig.1(b) demonstrates that the optimal discriminant vectors do not necessarily lie in the subspace spanned by the class centers.

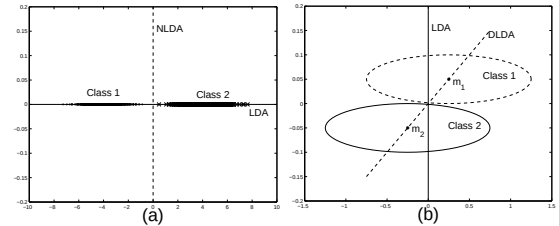


Figure 1: (a) shows that the discriminant vector (dashed line) of NLDA contains no discriminant information. (b) shows that the discriminant vector (dashed line) of DLDA is constrained to pass through the two class centers m_1 and m_2 . But according to the Fisher criteria, the optimal discriminant projection should be solid line (both in (a) and (b)).

2.2 Methods Aimed at Limitations of S_b

When the class conditional densities are multimodal, the class separability represented by S_b is poor. Especially in the case that each class shares the same mean, it fails to find the discriminant direction because there is no scatter of the class means[Fukunaga, 1990].

Notice the rank of S_b is $c - 1$, so the number of extracted features is, at most, $c - 1$. However, unless a posteriori probability function are selected, $c - 1$ features are suboptimal in Bayes sense, although they are optimal with regard to Fisher criterion [Fukunaga, 1990].

In fact, if classification is the ultimate goal, we need only estimate the class density well near the decision boundary [Hastie *et al.*, 2001].

Fukunaga and Mantock [Fukunaga and Mantock, 1983] presented a nonparametric discriminant analysis (NDA) in an attempt to overcome these limitations presented in LDA. In nonparametric discriminant analysis the between-class scatter S_b is of nonparametric nature. This scatter matrix is generally full rank, thus loosening the bound on extracted feature dimensionality. Also, the nonparametric structure of this matrix inherently leads to the extracted features that preserve relevant structures for classification. Bressan *et al.* [Bressan and Vitrià, 2003] explored the nexus between nonparametric discriminant analysis (NDA) and the nearest neighbors (NN) classifier and gave a slight modification of NDA which extends the two-class NDA to a multi-class version.

Although these nonparametric methods overcomes the limitations of S_b , they still depend on the singularity of S_w (or $\hat{S}_w$). The rank of $\hat{S}_w$ must be no more than $N - c$.

3 Stepwise Nearest Neighbor Discriminant Analysis

In this section, we propose a new feature extraction method, stepwise nearest neighbor discriminant analysis (SNDA). SNDA also uses nonparametric between-class and within-class scatter matrix. But it does not depend on singularity of within-class scatter matrix and improves the performance of NN classifier.

3.1 Nearest Neighbor Discriminant Analysis Criterion

Assuming a multi-class problem with classes $\omega_i (i = 1, \dots, c)$, we define the extra-class nearest neighbor for a sample $x \in \omega_i$ as

$$x^E = \{x' \notin \omega_i \mid \|x' - x\| \leq \|z - x\|, \forall z \notin \omega_i\}. \quad (5)$$

In the same fashion, the set of intra-class nearest neighbors are defined as

$$x^I = \{x' \in \omega_i \mid \|x' - x\| \leq \|z - x\|, \forall z \in \omega_i\}. \quad (6)$$

The nonparametric extra-class and intra-class differences are defined as

$$\Delta^E = x - x^E, \quad (7)$$

$$\Delta^I = x - x^I. \quad (8)$$

The nonparametric between-class and within-class scatter matrix are defined as

$$\hat{S}_b = \sum_{n=1}^N w_n (\Delta_n^E) (\Delta_n^E)^T, \quad (9)$$

$$\hat{S}_w = \sum_{n=1}^N w_n (\Delta_n^I) (\Delta_n^I)^T, \quad (10)$$

where w_n is the sample weight defined as

$$w_n = \frac{\|\Delta_n^I\|^\alpha}{\|\Delta_n^I\|^\alpha + \|\Delta_n^E\|^\alpha}, \quad (11)$$

where α is a control parameter between zero and infinity. This sample weight is introduced to deemphasize the samples in the class center and give emphases to the samples near to the other class. The sample that has a larger ratio between the nonparametric extra-class and intra-class differences is given an undesirable influence on the scatter matrix. The sample weights in Eq.(11) take values close to 0.5 near the classification boundaries and drop to zero as we move to class center. The control parameter α adjusts how fast this happens. In this paper, we set $\alpha = 6$.

From the Eq.(7) and (8), we can see that $\|\Delta_n^E\|$ represents the distance between the sample x_n and its nearest neighbor in the different classes, and $\|\Delta_n^I\|$ represents the distance between the sample x_n and its nearest neighbor in the same class. Given a training sample x_n , the accuracy of the nearest neighbor classification can be directly computed by examining the difference

$$\Theta_n = \|\Delta_n^E\|^2 - \|\Delta_n^I\|^2, \quad (12)$$

where Δ^E and Δ^I are nonparametric extra-class and intra-class differences and defined in Eq.(7) and (8).

If the difference Θ_n is more than zero, x_n will be correctly classified. Otherwise, x_n will be classified to the false class. The larger the difference Θ_n is, the more accurately the sample x_n is classified.

Assuming that we extract features by the $D \times d$ linear projection matrix W with a constraint that $W^T W$ is an identity matrix, the projected sample $x^{new} = W^T x$. The projected nonparametric extra-class and intra-class differences can be written as $\delta^E = W^T \Delta^E$ and $\delta^I = W^T \Delta^I$. So we expect to find the optimal W to make the difference $\|\delta_n^E\|^2 - \|\delta_n^I\|^2$ in the projected subspace as large as possible.

$$\hat{W} = \arg \max_W \sum_{n=1}^N w_n (\|\delta_n^E\|^2 - \|\delta_n^I\|^2). \quad (13)$$

This optimization problem can be interpreted as: find the linear transform that maximizes the distance between classes, while minimizing the expected distance among the samples of a single class.

Considering that,

$$\begin{aligned} & \sum_{n=1}^N w_n (\|\delta_n^E\|^2 - \|\delta_n^I\|^2) \\ &= \sum_{n=1}^N w_n (W^T \Delta_n^E)^T (W^T \Delta_n^E) - \sum_{n=1}^N w_n (W^T \Delta_n^I)^T (W^T \Delta_n^I) \\ &= \text{tr} \left(\sum_{n=1}^N w_n (W^T \Delta_n^E) (W^T \Delta_n^E)^T \right) \\ & \quad - \text{tr} \left(\sum_{n=1}^N w_n (W^T \Delta_n^I) (W^T \Delta_n^I)^T \right) \\ &= \text{tr} \left(W^T \left(\sum_{n=1}^N w_n \Delta_n^E (\Delta_n^E)^T \right) W \right) \end{aligned}$$

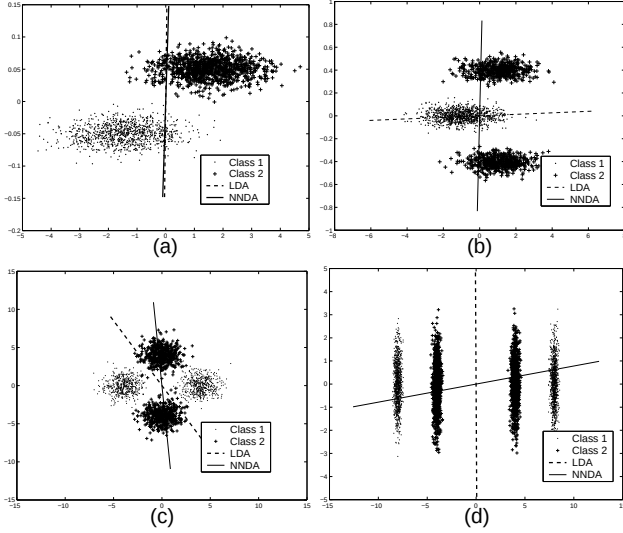


Figure 2: First projected directions of NNDA (solid) and LDA (dashed) projections, for four artificial datasets.

$$\begin{aligned}
& -tr(W^T (\sum_{n=1}^N w_n \Delta_n^I (\Delta_n^I)^T) W) \\
& = tr(W^T \hat{S}_b W) - tr(W^T \hat{S}_w W) \\
& = tr(W^T (\hat{S}_b - \hat{S}_w) W),
\end{aligned} \tag{14}$$

where $tr(\cdot)$ is the trace of matrix, $\hat{S}_b$ and $\hat{S}_w$ are the nonparametric between-class and within-class scatter matrix, as defined in Eq.(9) and (10).

So Eq.(13) is equivalent to

$$\widehat{W} = \arg \max_W tr(W^T (\hat{S}_b - \hat{S}_w) W). \tag{15}$$

We call Eq.(15) the *nearest neighbor discriminant analysis criterion (NNDA)*.

The projection matrix $\widehat{W}$ must be constituted by the d eigenvectors of $(\hat{S}_b - \hat{S}_w)$ corresponding to its first d largest eigenvalues.

Figure 2 gives comparisons between NNDA and LDA. When the class density is unimodal ((a)), NNDA is approximately equivalent to LDA. But in the cases that the class density is multimodal or that all the classes share the same mean ((b),(c) and (d)), NNDA outperforms LDA greatly.

3.2 Stepwise Dimensionality Reduction

In the analysis of the nearest neighbor discriminant analysis criterion, notice that we calculate nonparametric extra-class and intra-class differences (Δ^E and Δ^I) in original high dimensional space, then project them to the low dimensional space ($\delta^E = W^T \Delta^E$ and $\delta^I = W^T \Delta^I$), which does not exactly agree with the nonparametric extra-class and intra-class differences in projection subspace except for the orthonormal transformation case, so we have no warranty on distance preservation. A solution for this problem is to find the projection matrix $\widehat{W}$ by stepwise dimensionality reduction method. In each step, we re-calculate the nonparametric extra-class

and intra-class differences in its current dimensionality. Thus, we keep the consistency of the nonparametric extra-class and intra-class differences in the process of dimensionality reduction.

Figure 3 gives the algorithm of stepwise nearest neighbor discriminant analysis.

- Give D dimensional samples $\{x_1, \dots, x_N\}$, we expect to find d dimensional discriminant subspace.
- Suppose that we find the projection matrix $\widehat{W}$ via T steps, we reduce the dimensionality of samples to d_t in step t , and d_t meet the conditions: $d_{t-1} > d_t > d_{t+1}$, $d_0 = D$ and $d_T = d$.
- For $t = 1, \dots, T$
 1. Calculate the nonparametric between-class $\hat{S}_b^t$ and within-class scatter matrix $\hat{S}_w^t$ in the current d_{t-1} dimensionality,
 2. Calculate the projection matrix $\widehat{W}_t$, $\widehat{W}_t$ is $d_{t-1} \times d_t$ matrix.
 3. Project the samples by the projection matrix $\widehat{W}_t$, $x = \widehat{W}_t^T \times x$.
- The final transformation matrix $\widehat{W} = \prod_{t=1}^T \widehat{W}_t$.

Figure 3: Stepwise Nearest Neighbor Discriminant Analysis

3.3 Discussions

SNNDA has an advantage that there is no need to calculate the inverse matrix, so it is a more efficient and stable method. Moreover, though SNNDA optimizes the 1-NN classification, it is easy to extend it to the case of k -NN.

However, a drawback of SNNDA is the computational inefficiency in finding the neighbors when the original data space is high dimensionality. A improved method is that PCA is first used to reduce the dimension of data to $N - 1$ (the rank of the total scatter matrix) through removing the null space of the total scatter matrix. Then, SNNDA is performed in the transformed space. Yang *et al.* [Yang and Yang, 2003] shows that no discriminant information is lost in this transformed space.

4 Experiments

In this section, we apply our method to face recognition and compare it with the existing variant LDA methods and the other state-of-art face recognition approaches, such as PCA [Turk and Pentland, 1991], PCA+LDA [Belhumeur *et al.*, 1997], NLDA [Chen *et al.*, 2000], NDA [Bressan and Vitrià, 2003] and Bayesian [Moghaddam *et al.*, 2000] approaches. All the experiments are repeated 5 times independently and the average results are calculated.

4.1 Datasets

To evaluate the robustness of SNNDA, we perform the experiments on three datasets from the popular ATT face

database [Samaria and Harter, 1994] and FERET face database [Phillips *et al.*, 1998]. The descriptions of the three datasets are below:

ATT Dataset This dataset is the ATT face database (formerly ‘The ORL Database of Faces’), which contains 400 images (112×92) of 40 persons, 10 images per person. The images are taken at different times, varying lighting slightly, facial expressions (open/closed eyes, smiling/non-smiling) and facial details (glasses/no-glasses). Each image is linearly stretched to the full range of pixel values of $[0,255]$. Fig.4 shows some face examples in this database. The set of the 10 images for each person is randomly partitioned into a training subset of 5 images and a test set of the other 5. The training set is then used to learn basis components, and the test set for evaluate.



Figure 4: Face examples from ATT database

FERET Dataset 1 This dataset is a subset of the FERET database with 194 subjects only. Each subject has 3 images: (a) one taken under controlled lighting condition with a neutral expression; (b) one taken under the same lighting condition as above but with different facial expressions (mostly smiling); and (c) one taken under different lighting condition and mostly with a neutral expression. All images are pre-processed using zero-mean-unit-variance operation and manually registered using the eye positions. All the images are normalized by the eye locations and are cropped to the size of 75×65 . A mask template is used to remove the background and the hair. Histogram equalization is applied to the face images for photometric normalization. Two images for each person is randomly selected for training and the rest one is used for test.

FERET Dataset 2 This dataset is a different subset of the FERET database. All the 1195 people from the FERET Fa/Fb data set are used in the experiment. There are two face images for each person. This dataset has no overlap between the training set and the gallery/probe set according to the FERET protocol [Phillips *et al.*, 1998]. 500 people are randomly selected for training, and the remaining 695 people are used for testing. For each testing people, one face image is in the gallery and the other is for probe. All images are pre-processed by using the same method in FERET Dataset 1.

4.2 Experimental Results

Fig. 5 shows the rank-1 recognition rates with the different number of features on the three different datasets. It is shown that SNNDA outperforms the other methods. The recognition rate of SNNDA can reach almost 100% on ATT dataset.

The recognition rate of SNNDA have reached 100% on two FERET dataset surprisingly when the dimensionality of samples is about 20, while the other methods have poor performances in the same dimensionality. Moreover, SNNDA does not suffer from overfitting. Except SNNDA and PCA, the rank-1 recognition rates of the other methods have a descent when the dimensionality increases continuously.

Fig. 6 shows cumulative recognition rates on the three different datasets. From it, we can see that none of the cumulative recognition rates can reach 100% except SNNDA.

When dataset contains the changes of lighting condition (such as FERET Dataset 1), SNNDA also has obviously better performance than the others.

Different from ATT dataset and FERET dataset 1, where the class labels involved in training and testing are the same, the FERET dataset 2 has no overlap between the training set and the gallery/probe set according to the FERET protocol [Phillips *et al.*, 1998]. The ability of generalization from known subjects in the training set to unknown subjects in the gallery/probe set is needed for each method. Thus, the result on FERET dataset 2 is more convincing to evaluate the robust of each method. We can see that SNNDA also gives the best performance than the other methods on FERET dataset 2.

A major character, displayed by the experimental results, is that SNNDA always has a stable and high recognition rates on the three different datasets, while the other methods have unstable performances.

5 Conclusion

In this paper, we proposed a new feature extraction method, stepwise nearest neighbor discriminant analysis(SNNDA), which finds the important discriminant directions without assuming the class densities belong to any particular parametric family. It does not depend on the nonsingularity of the within-class scatter matrix either. Our experimental results on the three datasets from ATT and FERET face databases demonstrate that SNNDA outperforms the existing variant LDA methods and the other state-of-art face recognition approaches greatly. Moreover, SNNDA is very efficient, accurate and robust. In the further works, we will extend SNNDA to non-linear discriminant analysis with the kernel method. Another attempt is to extend SNNDA to the k -NN case.

References

- [Belhumeur *et al.*, 1997] P.N. Belhumeur, J. Hespanha, and D. Kiregeman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [Bressan and Vitrià, 2003] M. Bressan and J. Vitrià. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24:2743C2749, 2003.
- [Chen *et al.*, 2000] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33(10):1713–1726, 2000.

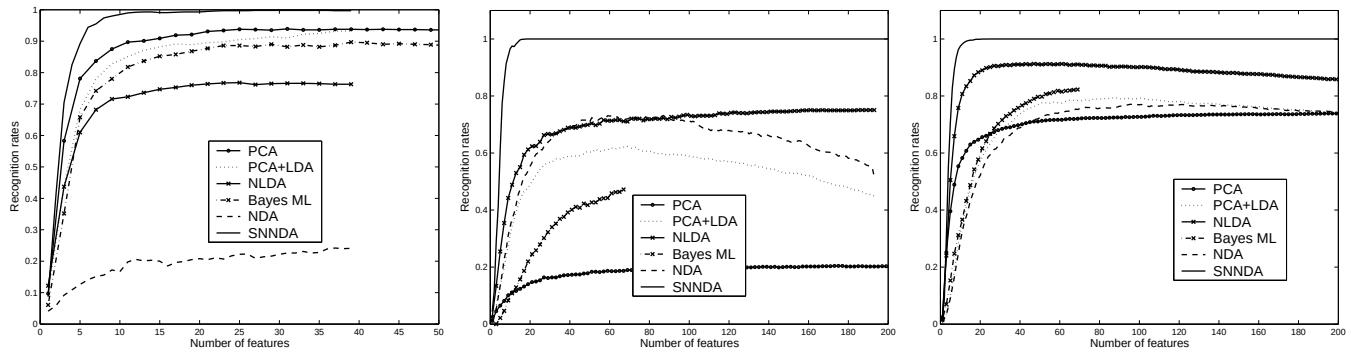


Figure 5: Rank-1 recognition rates with the different number of features on the three different datasets. (Left: ATT dataset; Middle: FERET dataset 1; Right: FERET dataset 2)

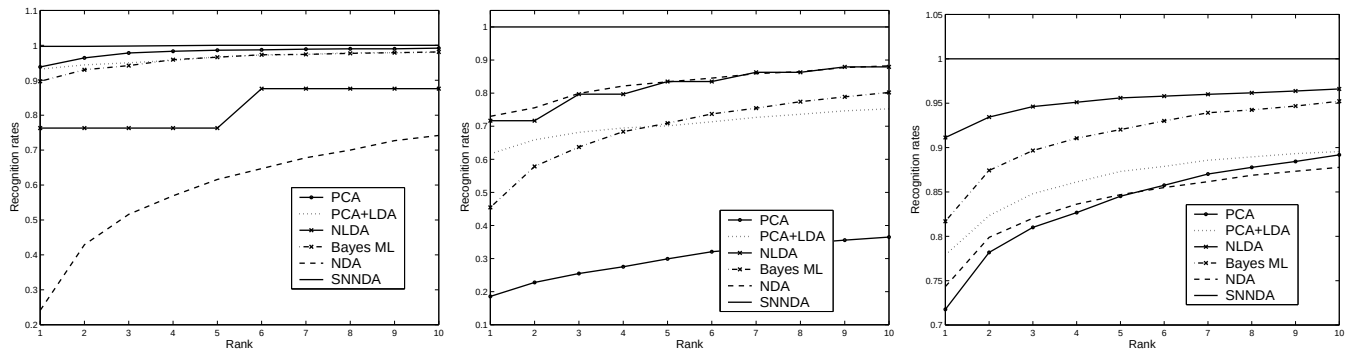


Figure 6: Cumulative recognition rates on the three different datasets. Left: ATT dataset (the number of features is 39; Middle: FERET dataset 1 (the number of features is 60); Right: FERET dataset 2 (the number of features is 60)

- [Duda *et al.*, 2001] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, New York, 2nd edition, 2001.
- [Fukunaga and Mantock, 1983] K. Fukunaga and J. Mantock. Nonparametric discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5:671C678, 1983.
- [Fukunaga, 1990] K. Fukunaga. *Introduction to statistical pattern recognition*. Academic Press, Boston, 2nd edition, 1990.
- [Hastie and Tibshirani, 1996] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:607C616, 1996.
- [Hastie *et al.*, 2001] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2001.
- [Li *et al.*, 2003] H.F. Li, T. Jiang, and K.S. Zhang. Efficient and robust feature extraction by maximum margin criterion. In *Proc. of Neural Information Processing Systems*, 2003.
- [Liu *et al.*, 1992] K. Liu, Y. Cheng, and J. Yang. A generalized optimal set of discriminant vectors. *Pattern Recognition*, 25(7):731C739, 1992.
- [Moghaddam *et al.*, 2000] B. Moghaddam, T. Jebara, and A. Pentland. Bayesian face recognition. *Pattern Recognition*, 33:1771–1782, 2000.
- [Phillips *et al.*, 1998] P.J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.
- [Samaria and Harter, 1994] Ferdinando Samaria and Andy Harter. Parameterisation of a stochastic model for human face identification. In *Proc. of 2nd IEEE Workshop on Applications of Computer Vision*, 1994.
- [Turk and Pentland, 1991] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [Yang and Yang, 2003] J. Yang and J.Y. Yang. Why can LDA be performed in PCA transformed space? *Pattern Recognition*, 36:563–566, 2003.
- [Yu and Yang, 2001] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34:2067–2070, 2001.

Using Predictive Representations to Improve Generalization in Reinforcement Learning

Eddie J. Rafols, Mark B. Ring, Richard S. Sutton, Brian Tanner

Department of Computing Science

University of Alberta

Edmonton, Alberta, Canada T6G 2E8

{erafols,markring,sutton,btanner}@cs.ualberta.ca

Abstract

The *predictive representations* hypothesis holds that particularly good generalization will result from representing the state of the world in terms of predictions about possible future experience. This hypothesis has been a central motivation behind recent research in, for example, PSRs and TD networks. In this paper we present the first explicit investigation of this hypothesis. We show in a reinforcement-learning example (a grid-world navigation task) that a predictive representation in tabular form can learn much faster than both the tabular explicit-state representation and a tabular history-based method.

1 Introduction

A *predictive* representation is one that describes the world in terms of predictions about future observations. The *predictive representations* hypothesis holds that such representations are particularly good for generalization. A good representation is one that captures regularities of the environment in a form useful to the learning agent; and in a reinforcement-learning task, something is “useful” if it increases the agent’s ability to receive rewards. Thus, representations generalize well when the regularities they capture allow an agent to learn more efficiently how to increase its cumulative reward.

Why do we believe that predictive representations might generalize particularly well? Good generalization tends to result when similar situations have similar representations. In interactive tasks, two situations are similar when executing an action sequence in one leads to about the same results as executing the same action sequence in the other, regardless of what the action sequence is. The more similar the results tend to be, the more similar the situations are, and, therefore, the more similar their predictive representations will be.

The predictive representations hypothesis makes an especially broad claim, and as a result it is especially resistant to testing or proof. In particular, questions regarding representation and generalization often involve the confounding issues of function approximation and representation acquisition — issues that we have developed specific measures to avoid. In reinforcement learning, for example, generalization is commonly achieved through function approximation: by learning

weightings from a set of features over the agent’s past and present perceptions. Good generalization occurs when the features are well chosen and are roughly independent of each other. Predictive representations, too, are generally used in conjunction with function approximation, where each prediction is treated as a feature.

Yet function approximation is itself a complex issue, requiring choices of method, optimization of learning parameters, etc., each of which can influence the results in its own way. To avoid such confusion and test the predictive-representations hypothesis most directly, we have controlled for the possible confounding effects of function approximation by developing a strictly tabular form of predictive representation. Furthermore, our focus is on how predictive representations affect generalization, not on how these representations are acquired, and so we assume they have already been acquired by some other process.

We illustrate the power of predictive representations in a grid-world navigation task where the agent’s action and observation space is chosen to be particularly impoverished, thus imposing a high degree of perceptual ambiguity. With only a small subset of the predictions necessary for a full representation of the environmental state, a reinforcement-learning agent can learn quickly and still achieve nearly optimal performance.

2 Predictive Representations

Several recently introduced methods for modeling dynamical systems have been inspired by the predictive-representations hypothesis. Among these are predictive state representations (PSRs) [Littman *et al.*, 2002] and temporal-difference networks (TD networks) [Sutton and Tanner, 2005]. Current research on predictive representations has focused principally on the acquisition of the representation (though their use for control is also beginning to be explored [James *et al.*, 2004; Izadi and Precup, 2003]).

PSRs are based on the concept of a core test—a sequence of actions followed by an observation—similar to the tests of Rivest and Schapire [1994]. An agent records the outcome of each core test as either a success or failure depending on whether the predicted observation matches the actual observation. The agent maintains the probability of success for each core test, and this value becomes a feature in the agent’s representation of the world. Knowledge of the world can be

expressed as a function (generally a linear combination) of these features. Core tests are continually acquired until the features form a sufficient statistic, which means that they capture all relevant, knowable information about the environment and can maintain such information from one time step to the next. (More will be said about sufficient statistics below.)

TD networks consist of two conceptually separate networks: a question network and an answer network. The question network poses questions about possible future observations. The answer network learns to predict answers to those questions. The questions are therefore analogous to the core tests of PSRs and the answer network is analogous to the function that computes the probabilities of core-test successes. However, the units of the question network can make predictions about not just the agent's observations, but also about the values of other network units, and these predictions of predictions allow the TD network to represent many possible core tests in a compact form. Furthermore, these predictions are generally, but not necessarily, action conditional.

The two approaches just outlined share the common aspiration of representing the world as a set of predictions about future observations. While research into predictive representations is still in its nascent phases, we attempt to make a prediction of our own about possible future observations. In this spirit we pose the following question: if a method does indeed prove successful at *acquiring* predictive representations, will these representations be good at generalization?

3 Tabular Predictive Representations

In order to focus exclusively on prediction as a basis for generalization, unclouded by issues involving representation acquisition, we assume the agent has already acquired the ability to make correct predictions. To eliminate issues involving function approximation, we consider deterministic tasks with a single, binary observation (though our method can also be applied to multiple binary observations, and we describe an extension in the future-work section for accommodating continuous observations or stochastic environments). To further focus our tests, we choose to look only at predictions that are contingent upon the agent's actions, though this restriction is not a requirement of predictive representations in general.

3.1 Identically Predictive Classes

We start with a set of binary tests resembling PSR core tests. Each test is of length n , meaning that it is a sequence of n actions followed by a single observation bit. We construct all possible tests of length 1 through n and produce a panel of test results in each state, one entry per test. Given N binary tests, the panel may take on 2^N distinct configurations of outcomes (assuming the environment is deterministic). If the agent has a actions available, then there are at most a^n distinct tests of length n , and the number of tests of length n or less is:

$$\begin{aligned} N &= \sum_{i=1}^n a^i \\ &= a^{n+1} - 1. \end{aligned}$$

If two states cannot be distinguished by any of the N tests, then the panel of results is identical in both states, and we

say these states are *identically predictive* for that value of n and belong to the same *identically predictive class* (similar to the states of Rivest and Schapire's Simple-Assignment Automata [1994], but constrained by n).

If for a given n an environment has c identically predictive classes, we arbitrarily number them 1 through c , and when an agent visits an environmental state, it observes the number for that state's identically predictive class. This number is therefore the agent's predictive representation in tabular form.

As an example, suppose that $a = 3$, and $n = 1$, then $N = 3$, and the panel consists of three tests (one for each action). Each test can result in an observation of 0 or 1, so there are 2^3 possible panel configurations and $c = 8$ (at most). Every state is aggregated into one of these eight classes, labeled 1 through 8, and the agent observes a 1, 2, 3, 4, 5, 6, 7, or 8 in each state that it visits.

In general, as n increases, both N and c increase, so there are fewer states per class on average and the agent's representation of its environment becomes more expressive.

However, not all 2^N panel configurations are necessarily represented in the environment: clearly, there can never be more classes c than there are environmental states S , so if $N > S$ then some tests must be equivalent (meaning there are no environmental states where the tests yield different results). Furthermore, as n increases, N increases exponentially, yet c tends to increase quite slowly in environments with even a moderate amount of regularity.

Eventually, increasing n no longer increases c , and the classes represent a sufficient statistic. At that point c may still be less than S if there are environmental states that cannot be distinguished by any test of any length.

3.2 Sarsa(0) with Identically Predictive Classes

All agents in this paper are trained using the reinforcement-learning algorithm known as episodic tabular Sarsa(0) [Sutton and Barto, 1998]. In the traditional Markov case—where the agent directly observes the environmental state—an action-value function is learned over the space of environmental states and actions. In this algorithm, the estimated value $Q(s, a)$ of each experienced state-action pair s, a is updated based on the immediate reward r and the estimated value of the next state-action pair; i.e.,

$$\Delta Q(s, a) = \alpha[r + Q(s', a') - Q(s, a)],$$

where α is a learning-rate parameter.

Episodic tabular Sarsa(0) is implemented over the predictive state space by mapping environmental states to their corresponding identically predictive classes, as described in the previous section. The function $C(\cdot)$ provides this mapping, and the resulting classes are then treated by the Sarsa agent as though they were environmental states:

$$\Delta Q(C(s), a) = \alpha[r + Q(C(s'), a') - Q(C(s), a)] \quad (1)$$

Because no distinction is made between the states within a class, the learning that occurs in one environmental state applies to all states mapped to the same class.

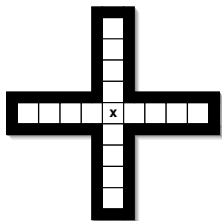


Figure 1: An agent in this world can have one of 4 orientations in each of the 17 grid squares. Its observation consists of one “nose touch” bit that senses only whether there is a wall directly in front of it. It chooses actions from {Rotate 90° Right, Rotate 90° Left, Advance}. There are therefore 68 distinct environmental states, but a predictive representation will identify (at most) 17 identically predictive classes.

3.3 Performance and Generalization

In episodic tasks the performance of a reinforcement-learning agent can be quantified by the total reward it receives per episode. Given infinite time, agents that observe the (Markov) state of the environment can achieve optimal performance.

In practice, however, learning an optimal policy may take arbitrarily long for an arbitrarily large state space, and these days one can rarely find enough time, let alone an infinite or even an arbitrary amount of it. It is therefore desirable in many cases to use methods that speed learning, even if the final solution is not absolutely optimal.

We anticipate a trade-off between the agent’s asymptotic performance and its speed of learning. As c increases, the classes more closely resemble the Markov state of the environment and the agent’s asymptotic performance approaches optimal. As c decreases, there are fewer distinct cases to learn about and learning speed improves, but there is a risk that the states comprising a class will disagree about the optimal action for that class. This disagreement can lead to sub-optimal action selection, and in extreme cases the conflict may be catastrophic, precluding the discovery of any reasonable policy. Therefore, a good representation will find low values of c while minimizing the amount of disagreement within classes. The predictive-representations hypothesis specifically holds that the classes formed through predictive representations will do just that.

3.4 Predictive Classes and Sufficient Statistics

An agent’s representation of the world is a sufficient statistic if it cannot be improved through any further experience with the world. In the case of predictive representations, a sufficient statistic implies that there are no additional predictions that can add knowledge or improve performance. On the other hand, adding more predictions to a sufficient statistic can never worsen asymptotic performance.

In the tabular case, if a sufficient statistic consists of c classes, no number of additional tests will increase c . (If by adding a test we were to end up with c' classes, where $c' > c$, then our original representation could not have been a sufficient statistic, since it did not represent the knowledge captured by the newest $c' - c$ class distinctions.) Nor will any

further tests ever reduce the number of classes or change the way states have been assigned to classes.

But an agent whose predictive representation is a sufficient statistic may still not distinguish all environmental states. This idea is illustrated in Figure 1 where there are 68 distinct environmental states (four orientations in each grid cell).

Given an agent that has three actions (Rotate 90° Right, Rotate 90° Left, Advance) and that observes only whether or not there is a wall directly in front of it, each arm of the cross will produce exactly the same sets of predictions. To the predictive agent, the arms are identical because there is no test that can distinguish a state in one arm from the corresponding states in the other three arms, even with infinite-length tests. Therefore, for this environment there are a maximum of 17 identically predictive classes (four orientations in four arm cells, plus only one in the center because all orientations in the center are identically predictive).

This example also illustrates how an agent’s learning speed can be improved through the use of predictive representations. If the task is to navigate to the cell marked X, an agent observing identically predictive class numbers will learn an optimal policy much faster than an agent observing environmental state, because the predictive-class agent has only a quarter of the situations to learn about and therefore requires far less experience to learn about all possible situations.

4 Tabular History-based Representations

The obvious competitors to predictive representations are history-based representations. Of these, the fixed-length or Markov- k approaches [Ring, 1994; McCallum., 1996; Mitchell, 2003] most clearly lend themselves to a tabular format. To promote a fair comparison between representations, we define a k -length history to be an observation followed by k action-observation pairs. In tabular format, each possible history is uniquely labeled.

Predictive representations and fixed-length history representations offer fundamentally different kinds of generalization. Specifically, most environmental states can be reached via multiple different paths, each corresponding to a different history; conversely, a single action sequence (history) may reach multiple environmental states by starting from different states. Therefore, the mapping between environmental states and fixed-length history sequences is many to many. But the mapping between environmental states and identically predictive classes is many to one.

Conceptually, the reason for the difference is that an agent can take many paths to arrive in a state, but once there, has only one set of possible futures; and the set of possible futures is absolutely fixed for each environmental state. For example, if $k = 1$ the agent has at least two ways of reaching every state in Figure 1—rotating right or rotating left. If $k = 2$ there are at least 4 different histories for each state. The number of fixed-length history representations that can lead to each environmental state increases exponentially with k , so as k increases, there are an exponential number of cases that the agent must learn about. (This problem does not automatically disappear by using function approximation methods.)

In contrast, the set of possible futures available from each

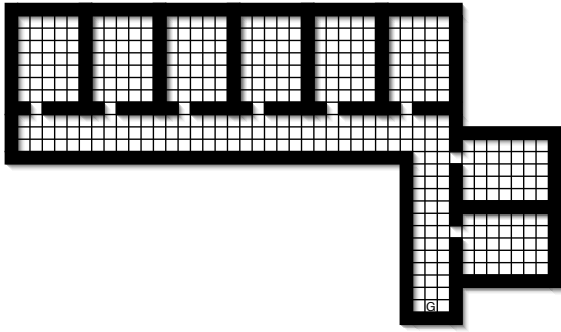


Figure 2: The “office layout” grid-world used for the navigation task. The agent starts an episode in one of the six top rooms, and finishes in the square marked **G**.

state is the same no matter how that state is reached, and as a result, the number of predictive classes is never more than the number of distinct environmental states. The nature of this mapping makes generalization in the predictive case far more intuitive, and to some extent the predictive-representations hypothesis is based on this clearer intuition.

5 Experiment Design

We examined the predictive-representations hypothesis by first designing a grid-world with a large degree of regularity (Figure 2). We then tested Sarsa(0) agents with four different methods of representation on a task in this environment to see how well each method was able to exploit the environment’s regularities.

The agent’s task was to navigate to the goal cell (marked **G**) from any randomly chosen square in one of the top six rooms. (The environment was designed to resemble a typical office layout, and the task can be likened to finding the quickest route to the staircase.) Representations that generalize well should allow their respective agents to exploit the regularities in the environment to improve their speed of learning.

As in Figure 1, the agent has a one-bit observation and three possible actions. The rewards for the task are +1 for reaching the goal state and −1 on all other timesteps. All transitions in the environment are deterministic and the environment has a total of 1696 states, 840 of which are possible start states. On average, there are 42.2 steps along the optimal path from start to goal. The task is formulated to be undiscounted and episodic; thus the agent is transported to a randomly chosen starting position upon reaching the goal.

In every case actions were chosen according to an ϵ -greedy policy; ϵ was set to 0.1, and α was set to 0.25, which are typical values for Sarsa agents in episodic tasks.

The four representational schemas tested were:

- Markov
- n -depth predictive classes
- k -step fixed-length histories
- random state aggregation

In the Markov case the agent directly observed its current environmental state, each state being represented by a unique

Agent		Unique Observations	% Environmental States
Markov	-	1696	100%
Predictive	n		
	2	67	3.9%
	3	185	10.7
	4	308	17.8
	5	416	24.1
	6	497	28.8
	7	568	32.9
Fixed-History	k		
	2	50	2.9%
	3	205	11.9
	4	790	45.7
	5	2,938	170.0
	6	10,660	616.9
Random	case		
	1	1,526	90%
	2	1,611	95
	3	1,679	99

Figure 3: The four representational schemas in their tested instantiations and the degree of state aggregation in each case. In the predictive case, the “unique observations” column represents the number of identically predictive classes, c . For fixed-length history, this column represents the number of unique histories that occurred during training.

label. In the predictive case the agent observed a label corresponding to the identically predictive class (as described in Section 3.2) for six different values of n . For each value of n , states were assigned to classes as described in Section 3.1. In the fixed-history case, the agent observed a label corresponding to the k -step history it had just experienced (Section 4) for five different values of k . To see whether our results in the predictive case were merely due to beneficial properties of state aggregation, we tried randomly assigning states to classes and then training according to Equation 1.

Figure 3 shows the vital stats for each of the representational methods tested. The amount of state aggregation that occurs in each method is shown in terms of the ratio of unique observations to environmental states.

6 Results

Performance results for the different representational schemas given in Figure 3 are graphed in Figure 4, with the exception of the random state aggregation method, which performed too poorly to be graphed meaningfully.

Each point in the graph represents the average number of steps per episode over the previous 10 episodes. The curves are averaged over 10,000 trials, each trial being 1,000 episodes. At the end of each trial, the agent’s action values are reset, and learning begins from scratch. Over the course of 1,000 episodes, the Markov case shows a smooth, steadily improving curve, which by the 1,000th episode is performing very close to optimal.

Though fixed-length history representations learn more slowly than the other methods, their learning speeds are

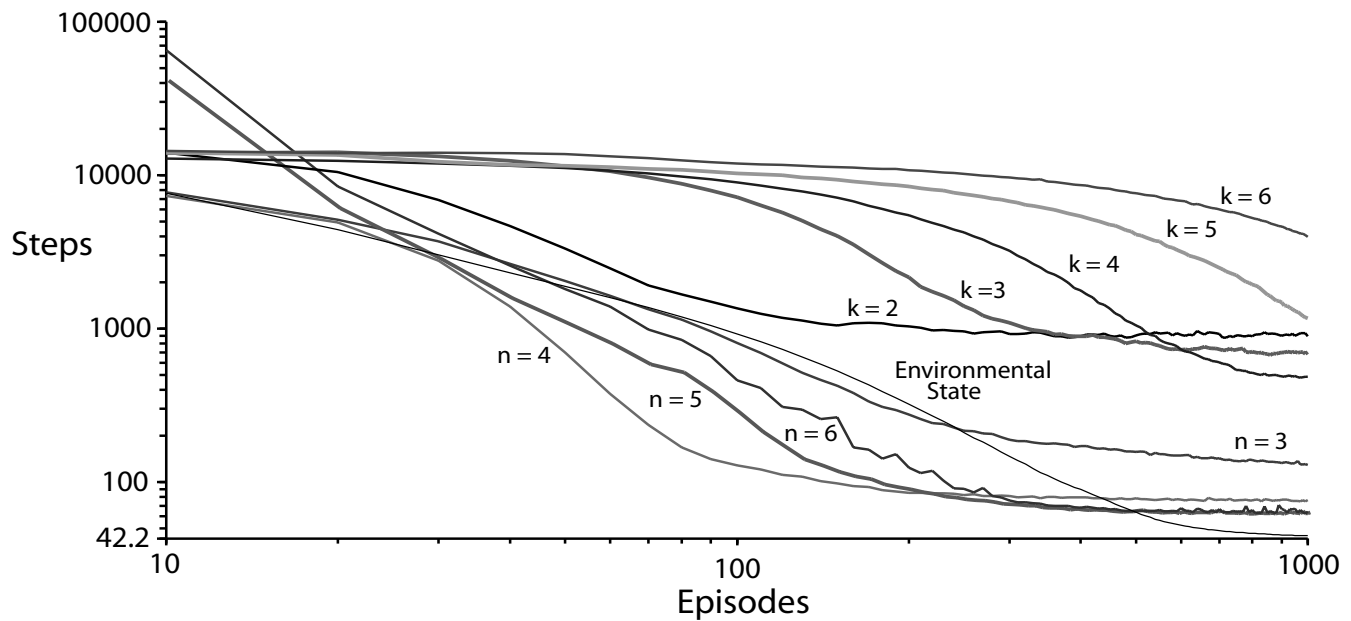


Figure 4: A comparison of three representational methods on the task of Figure 2. The representational schemas shown are: explicit environmental state, history-based representations for $k = 2$ through 6, and identically predictive classes for $n = 3$ through 6. The vertical axis shows average steps taken per episode over the past 100 episodes of training. Note that the axes are logarithmic and the vertical axis intersects the horizontal axis at the optimum.

fastest for small values of k and their final results are best for large values of k , demonstrating the trade-off between representational expressiveness and learning speed. The number of histories increases (exponentially) with k , which negatively impacts learning speed but positively impacts the final results of learning. Interestingly, on individual trials the curves for the history representation resemble a step function. Apparently, the ϵ -greedy action selection chooses poorly for many episodes until the agent suddenly learns some key piece of information that dramatically improves performance. As k grows, the agent takes longer to discover this piece of information but then converges to a better solution.

The results look promising for predictive representations. They allow both speedy learning and convergence to a good policy. In general, the results for the identically predictive representations are similar to those for the fixed-history representations in that convergence speed decreases and convergence quality increases as n increases. However, in contrast to the fixed-history representation, the number of identically predictive classes increases quite slowly with n and the generalization benefit of the predictive classes is clear. The representation effectively aggregates similar states, allowing the agent to converge to near-optimal solutions. This result closely matches our intuition and expectations. Only in the $n = 3$ case was there no improvement in learning speed, indicating there may be a maximum degree of state aggregation beyond which learning speed is not improved—also an expected and intuitive result from section 3.3. Results in the $n = 2$ and $n = 7$ cases match these trends but are not shown so as to keep the graph at least vaguely readable.

The case of random state aggregation is not shown in the

graphs because these agents performed so poorly. With only 10% state aggregation (90% of the states not aggregated), random state aggregation brought about catastrophic failure in over 80% of the agents tested, meaning that they show no progress in their training. To avoid catastrophic failure in 90% of the agents, the amount of state aggregation had to be 1% or less, meaning that only one state in a hundred shared a class with another state.

In contrast, there were no cases of catastrophic failure in any of the agents using predictive representations.

It might be argued that the predictive representations benefit unfairly from the preprocessing done to create the classes, which the history-based methods did not have, and which an actual learning agent may not reasonably be expected to acquire. Or it might be argued that history-based methods *could* somehow be augmented to combine all ways of reaching a state into equivalence classes as our tabular representations do for predictions. But these objections miss two fundamental aspects of this research and of predictive representations in general. First, it is not our intention here to prove the superiority of one particular learning algorithm to any other, but only to test out, in advance, whether the research direction of predictive representations looks promising. (The evidence collected clearly gives reason for optimism.) Second, acquisition of a predictive representation is always directed toward acquisition of a sufficient statistic. Once an agent acquires the predictions necessary for a sufficient statistic, then it can in principle create the class mappings used here. (In fact, those mappings would fall directly out of a TD network representation, being merely a subset.) It is not clear what the equivalent of a sufficient statistic might be for history-based

methods, nor whether it could be reasonably acquired.

Another objection is that performance could be improved even more by hand-coding a mapping of states to classes. But, don't be so sure! While it is obvious that an optimal policy could be encoded by just three classes (one for each action), finding a mapping that encourages learning is a completely different matter, and it is difficult to design a mapping by hand that (a) aggregates the states into a small enough number of classes to allow good generalization, (b) does not risk catastrophic failure by mapping two states into a class that makes learning impossible, and (c) is not essentially based on our intuitive notions of prediction.

7 Conclusions and Future Work

This paper makes an initial attempt to answer the question: if there were a learning system that could represent the world in terms of predictions about possible future experience, would that representation turn out to be useful for generalization in reinforcement-learning tasks? While this question is itself quite broad (fully answering it might require thorough testing in the presence of many confounding factors), our results are clear and lend weight to the possibility of a yes answer.

In comparison to other representational schemas, our tests suggest that predictive representations may generalize a state space very well, allowing faster learning without obvious risk of catastrophic failure through poor state aggregation.

But there is another significant conclusion to be reached. It is fairly common in reinforcement-learning tasks to provide agents with the environmental state in tabular form. Though it was our intention only to *examine* predictive representational methods, the particular tabular method we developed shows significant advantages over the Markov representation often seen in the reinforcement-learning literature. We suggest it may have unintended practical application to tasks with large state spaces when the environmental state is readily available. Mapping states into identically predictive classes may speed learning and still allow nearly optimal performance to be achieved. This certainly deserves further investigation.

We would also like to test predictive representations in non-deterministic tasks and in environments with continuous-valued observations. In the discrete, deterministic case we can make binary predictions of observations, but in environments with continuous-valued observations, the predictions would also be continuous, and in the non-deterministic tasks the predictions would be continuous-valued probabilities. One solution is to use tile coding to discretize the continuous values, reducing them to binary values that can be treated as described above. A second approach is to view the panel of test results as vectors and measure their similarity with a metric such as Euclidean distance. Vectors within a certain distance would belong to the same identically predictive class.

It should be noted that we have avoided the issue of poor or malicious placement of rewards. What would happen, for example, if the goal in Figure 1 were placed in the middle of one of the arms? The agent, not able to distinguish between one arm and the others, would actually be hindered by using a predictive representation. The solution is to incorporate re-

ward(s) into the predictive model of the world. The ability to predict rewards would increase the expressiveness of the representation and could assist the integration of predictive representations into reinforcement-learning algorithms.

Finally, there is an obvious enhancement to the classification method we employed in this paper. Since small values of n speed learning the most but large values of n converge to a better policy, it would make sense to gradually increase n , splitting up the identically predictive classes into smaller and smaller groups whenever learning begins to converge. Since all the states in each newly derived class would come strictly from the same parent class, the action values for the new classes could be initialized to those of the parent class, and all training done up to that point would be preserved.

Acknowledgments

We thank Anna Koop for her participation in our early discussions on this paper and for being a good sport. Thanks to Mark Lee for help with the simulation code. This research was supported in part by grants from iCORE, and NSERC, and by DARPA grant HR0011-04-1-0050.

References

- [Izadi and Precup, 2003] Masoumeh T. Izadi and Doina Precup. A planning algorithm for predictive state representations. In Georg Gottlob and Toby Walsh, editors, *IJCAI*, pages 1520–1521. Morgan Kaufmann, 2003.
- [James *et al.*, 2004] Michael R. James, Satinder Singh, and Michael L. Littman. Planning with predictive state representations. In *ICMLA-04, Proceedings of the International Conference on Machine Learning and Applications*, pages 304–311, 2004.
- [Littman *et al.*, 2002] Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.
- [McCallum., 1996] Andrew Kachites McCallum. *Reinforcement Learning with Selective Perception and Hidden State*. PhD thesis, Department of Computer Science, University of Rochester, Rochester, New York, August 1996.
- [Mitchell, 2003] Matthew W. Mitchell. Using markov-k memory for problems with hidden-state. In Hamid R. Arabnia and Elena B. Kozerenko, editors, *MLMTA*, pages 242–248. CSREA Press, 2003.
- [Ring, 1994] Mark B. Ring. *Continual Learning in Reinforcement Environments*. PhD thesis, University of Texas at Austin, Austin, Texas 78712, August 1994.
- [Rivest and Schapire, 1994] Ronald L. Rivest and Robert E. Schapire. Diversity-based inference of finite automata. *J. ACM*, 41(3):555–589, 1994.
- [Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.
- [Sutton and Tanner, 2005] Richard S. Sutton and Brian Tanner. Temporal-difference networks. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

InterActive Feature Selection

Hema Raghavan*

University of Massachusetts
140 Governor's Drive,
Amherst, MA-01002, USA
hema@cs.umass.edu

Omid Madani

Yahoo! Inc.
74 N Pasadena Ave, Pasadena,
CA 91103 USA
madani@yahoo-inc.com

Rosie Jones

Yahoo! Inc.
74 N Pasadena Ave, Pasadena,
CA 91103 USA
jonesr@yahoo-inc.com

Abstract

We study the effects of feature selection and human feedback on features in active learning settings. Our experiments on a variety of text categorization tasks indicate that there is significant potential in improving classifier performance by feature reweighting, beyond that achieved via selective sampling alone (standard active learning) if we have access to an *oracle* that can point to the important (most predictive) features. Consistent with previous findings, we find that feature selection based on the labeled training set has little effect. But our experiments on human subjects indicate that human feedback on feature relevance can identify a sufficient proportion (65%) of the most relevant features. Furthermore, these experiments show that feature labeling takes much less (about 1/5th) time than document labeling. We propose an algorithm that interleaves labeling features and documents which significantly accelerates active learning.

1 Introduction

A major bottleneck in machine learning applications is the lack of sufficient labeled data for adequate classifier performance, as manual labeling is often tedious and costly. Techniques such as active learning, semi-supervised learning, and transduction have been pursued with considerable success in reducing labeling requirements. In the standard active learning paradigm, learning proceeds sequentially, with the learning algorithm actively asking for the labels of instances from a teacher. The objective is to ask the teacher to label the most informative instances in order to reduce labeling costs and accelerate the learning. There has been very little work in supervised learning in which the user (teacher) is queried on aspects other than class assignment of instances. In experiments in this paper we study the benefits and costs of feature feedback via humans on active learning. To this end we pick document classification [Sebastiani, 2002] as the learning problem of choice because it represents a case of supervised learning which traditionally relies on example documents as input for training and where users have sufficient

prior knowledge on features which can be used to accelerate learning. For example, to find documents on the topic *cars* in traditional supervised learning the user would be required to provide sufficient examples of *cars* and *non-cars* documents. However, this is not the only way in which the information need of a user looking for documents on *cars* can be satisfied. In the *information retrieval* setting the user would be asked to issue a query, that is, state a few words (features) indicating her information need. Thereafter, feedback which may be at a term or at a document level may be incorporated. In fact, even in document classification, a user may use a keyword based search to locate the initial training examples. However, traditional supervised learning tends to ignore the prior knowledge that the user has, once a set of training examples have been obtained. In this work we try to find a marriage between approaches to incorporating user feedback from machine learning and information retrieval and show that active learning should be a dual process – at the term and at the document-level. This has applications in email filtering and news filtering where the user has some prior knowledge and a willingness to label some (as few as possible) documents in order to build a system that suits her needs. We show that humans have good intuition for important features in text classification tasks since features are typically words that are perceptible to the human and that this human prior knowledge can indeed accelerate learning.

In summary, our contributions are: (1) We demonstrate that access to a feature importance oracle can improve performance ($F1$) significantly over uncertainty sampling. (2) We show that even naive users can provide feedback on features with about 60% accuracy of the oracle. (3) We show that the relative manual costs of labeling features is about 1/5th that of document feedback. (4) We show a method of simultaneously soliciting class labels and feature feedback that improves classifier performance significantly.

We describe the experimental setup in Sec. 2 and show how feature selection using an oracle is useful to active learning in Sec. 3. In Sec. 4 we show that humans can indeed identify useful features and show how human-chosen features can be used to accelerate learning in Sec. 5. We relate our work to past work in Sec. 6 and outline directions for the future in section Sec. 7.

*This work was done in part when the author was at Yahoo! Inc.

2 Experimental setup

Our test bed for this paper comes from three domains:

(1) The 10 most frequent classes from the Reuters-21578 corpus (12902 documents). (2) The 20-Newsgroups corpus (20000 documents from 20 Usenet newsgroups). (3) The first 10 topics from the TDT-2001 corpus (67111 documents in 3 languages from broadcast and news-wire sources).

For all three corpora we consider each topic as a *one versus all* classification problem. We also pick two binary classification problems viz., *Baseball vs Hockey* and *Automobiles vs Motorcycles* from the 20-Newsgroups corpus. In all we have 42 classification problems.¹ All the non-english stories in the TDT corpus were machine translated into English. As features we use words, bigrams and trigrams obtained after stopping and stemming with the Porter stemmer in the Rainbow Toolkit [McCallum, 1996]

We use linear support vector machines (SVMs) and uncertainty sampling for active learning [Scholkopf and Smola, 2002; Lewis and Catlett, 1994]. SVMs are the state of art in text categorization, and have been found to be fairly robust even in the presence of many redundant and irrelevant features [Brank *et al.*, 2002; Rose *et al.*, 2002]. Uncertainty sampling [Lewis and Catlett, 1994] is a type of active learning in which the example that the user (teacher) is queried on is the unlabeled instance that the classifier is most uncertain about. When the classifier is an SVM, unlabeled instances closest to the margin are chosen as queries [Tong and Koller, 2002]. The active learner may have access to all or a subset of the unlabeled instances. This subset is called the pool and we use a pool size of 500 in this paper. The newly labeled instance is added to the set of labeled instances and the classifier is retrained. The user is queried a total of T times.

The *Deficiency* metric [Baram *et al.*, 2003] quantifies the performance of the querying function for a given active learning algorithm. Originally deficiency was defined in terms of accuracy. Accuracy is a reasonable measure of performance when the positive class is a sizeable portion of the total. Since this is not the case for all the classification problems we have chosen, we modify the definition of deficiency, and define it in terms of the $F1$ measure (harmonic mean of precision and recall [Rose *et al.*, 2002]). Using notation similar to the original paper [Baram *et al.*, 2003], let $\mathcal{U}$ be a random set of P labeled instances, $F1_t(RAND)$ be the average $F1$ achieved by an algorithm when it is trained on t randomly picked examples and $F1_t(ACT)$ be the average $F1$ obtained using t actively picked examples. Deficiency $\mathcal{D}$ is defined as:

$$\mathcal{D}_T = \frac{\sum_{t=init}^T (F1_M(RAND) - F1_t(ACT))}{\sum_{t=init}^T (F1_M(RAND) - F1_t(RAND))} \quad (1)$$

$F1_M(RAND)$ is the $F1$ obtained with a large number (M) of randomly picked examples. For this paper we take $M = 1000$ and $t = 2, 7, \dots, 42$. When $t = 2$ we have one positive and one negative example. $F1_t(\bullet)$ is the average $F1$ computed over 10 trials. In addition to deficiency we report $F1_t$ for some values of t . Intuitively, if C_{act} is the curve

obtained by plotting $F1_t(ACT)$, C_{rand} is the corresponding curve using random sampling and C_M is the straight line $F1_t = F1_M$ then deficiency is the ratio of the area between C_{act} and C_M and the area between C_{rand} and C_M . The lower the deficiency the better the active learning algorithm. We aim to minimize deficiency and maximize $F1$.

3 Oracle Feature Selection Experiments

The oracle in our experiments has access to the labels of all P documents in $\mathcal{U}$ and uses this information to return a list of the k most important features. We assume that the parameter k is input to the oracle. The oracle orders the k features in decreasing information gain order. Given a set of k features we can perform active learning as discussed in the previous section and plot C_{act} for each value of k .

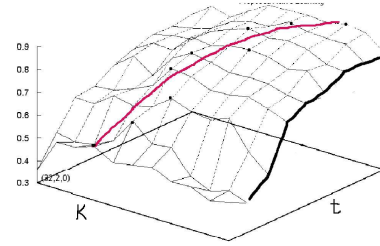


Figure 1: Average $F1_t(ACT)$ for different values of k . k is the number of features and t is the number of documents.

Figure 1 shows a plot of $F1_t(ACT)$ against number of features k and number of labeled training examples t , for the *Earnings* category in Reuters. The x , y and z axes denote k , t and $F1$ respectively. The number of labeled training examples t ranges from 2...42 in increments of 5. The number of features used for classification k has values from 32 to 33718 (all features). The dark dots represent the maximum $F1$ for each value of t , while the dark band represents the case when all features are used. This method of learning in one dimension is representative of traditional active learning. Clearly when the number of documents is few, performance is better when there is a smaller number of features. As the number of documents increases the number of features needed to achieve best accuracy increases. From the figure it is obvious that we can get a big boost in accuracy by starting with fewer features and then increasing the complexity of the model (the number of relevant features) as the number of labeled documents increase.

All 42 of our classification problems exhibit behavior like that in Figure 1 [Raghavan *et al.*, 2005]. We report the average deficiency, $F1_7$ ($F1$ score with 7 labeled examples) and $F1_{22}$ in Fig. 2 to illustrate this point. The column labeled *Act* shows performance using traditional active learning and all the features. The column labeled *Ora* shows performance obtained using a reduced subset of features using the Oracle.

Intuitively, with limited labeled data, there is little evidence to prefer one feature over another. Feature/dimension reduction (by the oracle) allows the learner to “focus” on dimensions that matter, rather than being “overwhelmed” with

¹ <http://www.daviddlewis.com/resources/testcollections/reuters21578/>, <http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html>, <http://www ldc.upenn.edu/Projects/TDT3/>

Class ↓	$\mathcal{D}_{42}$		$F1_7$		$F1_{22}$		$F1_{1000}$
	Act	Oracle	Act	Ora	Act	Ora	Act
Reuters	0.421	0.319	0.345	0.446	0.569	0.621	0.727
20-News.	0.602	0.344	0.072	0.222	0.21	0.29	0.446
TDT	0.735	0.656	0.186	0.290	0.282	0.407	0.751
Bas. vs Hock	0.710	0.447	0.587	0.701	0.785	0.828	0.963
Auto vs Mot.	0.676	0.321	0.431	0.724	0.758	0.860	0.899

Figure 2: Improvements in deficiency, $F1_7$ and $F1_{22}$ using an oracle to select the most important features. Remember that the objective is to minimize deficiency and maximize $F1$. For each of the three metrics, figures in bold are statistically significant improvements over Uncertainty sampling using all features (the corresponding columns denoted by Act). When 1000 documents are labeled ($F1_{1000}$) using the entire feature set leads to better $F1$ scores.

numerous dimensions right at the outset of learning. As the number of labeled examples increases, feature selection becomes less important, as the learning algorithm becomes more capable of finding the discriminating hyperplane (feature weights). We experimented with filter based methods for feature selection, which did not work very well (i.e., tiny or no improvements). This is expected given such limited training set sizes (see Fig. 3), and is consistent with most previous findings [Sebastiani, 2002]. Next we determine if humans can identify these *important features*.

4 Human Labeling

Consider our introductory example of a user who wants to find all documents that discuss *cars*. From a human perspective the words ‘car’, ‘auto’ etc may be important features in documents discussing this topic. Given a large number of documents labeled as on-topic and off-topic, and given a classifier trained on these documents, the classifier may also find these features to be most relevant. With little labeled data (say 2 labeled examples) the classifier may not be able to determine the discriminating features. While in general in machine learning the source of labels is not important to us, in active learning scenarios in which we expect the labels to come from humans we have valid questions to pose: (1) Can humans label features as well as documents? (2) If the labels people provide are noisy through being inconsistent, can we learn well enough? (3) Are features that are important to the classifier perceptible to a human?

Our concern in this paper is asking people to give feedback on features, or word n-grams, as well as entire documents. We may expect this to be more efficient, since documents contain redundancy, and results from our oracle experiments indicate great potential. On the other hand, we also know that synthetic examples composed of a combination of real features can be difficult to label [Baum and Lang, 1992].

4.1 Experiments and Results

In order to answer the above questions we conducted the following experiment. We picked 5 classification problems which we thought were perceptible to the average person on the street and also represented the broad spectrum of problems from our set of 42 classification problems. We took the two binary classification problems and from the remaining 40 one-versus-all problems we chose three (*earnings*, *hurricane Mitch* and *talk.politics.mideast*). For a given classification

problem we took the top 20 features as ranked by information gain on the entire labeled set. In this case we did not stem the data so that features remain as legitimate English words. We randomly mix these with features which are much lower in the ranked list. We show each user one feature at a time and give them two options – *relevant* and *not-relevant/don’t know*. A feature is relevant if it helps discriminate the positive or the negative class. We measure the time it takes the user to label each feature. We do not show the user all the features as a list, though this may be easier, as lists provide some context and serve as a summary. Hence our method provides an upper bound on the time it takes a user to judge a feature. We compare this with the time it takes a user to judge a document. We measure the precision and recall of the user’s ability to label features. We ask the user to first label the features and then documents, so that the feature labeling process receives no benefit due to the fact that the user has viewed relevant documents. In the learning process we have proposed, though, the user would be labeling documents and features simultaneously, so the user would indeed be influenced by the documents he reads. Hence our method is more stringent than the real case. We could in practice ask users to highlight terms as they read documents. Experiments in this direction have been conducted in information retrieval [Croft and Das, 1990].

Our users were six graduate students and two employees of a company, none of whom were authors of this paper. Of the graduate students, five were in computer science and one from public health. All our users were familiar with the use of computers. Five users understood the problem of document classification but none had worked with these corpora. One of our users was not a native speaker of English. The topics were distributed randomly, and without considering user expertise, so that each user got an average of 2-3 topics. There were overlapping topics between users such that each topic was labeled by 2-3 users on average. A feedback form asking the users some questions about the difficulty of the task was handed out at the end.

We evaluated user feature labeling by calculating their average precision and recall at identifying the top 20 features as ranked by an oracle using information gain on the entire labeled set. Fig. 3 shows these results. For comparison we have also provided the precision and recall (against the same oracle ranking of top 20 features) obtained using 50 labeled examples (picked using uncertainty sampling) denoted by @50.

Class Problem	Prec.		Rec.		Avg. Time (secs)	
	Hum.	@50	Hum.	@50	Feat.	Docs
Baseball..	0.42	0.3	0.7	0.3	2.83	12.6
Auto vs ...	0.54	0.25	0.81	0.25	3.56	19.84
Earnings	0.53	0.2	0.66	0.25	2.97	13
...mideast	0.68	0.35	0.55	0.35	2.38	12.93
...Mitch	0.716	0.65	0.56	0.65	2.38	13.19
Average	0.580	0.35	0.65	0.38	2.82	14.31

Figure 3: *Ability of users to identify important features. Precision and Recall against an oracle, of users (Hum.) and an active learner which has seen 50 documents(@50). Average labeling times for features and documents are also shown. All numbers are averaged over users.*

Precision and Recall of the humans is high, supporting our hypothesis that features that a classifier finds to be relevant after seeing a large number of labeled instances are obvious to a human after seeing little or no labeled data (the latter case being true of our experiments). Additionally the Precision and Recall @50 is significantly lower than that of humans, indicating that a classifier like an SVM needs to see much more data before it can find the discriminatory features.

The last column of Fig. 3 shows time taken for labeling features and documents. On average humans require about 5 times longer to label documents than to label features. Note that features may be even easier to label if they are shown in context – as lists, with relevant passages etc. There are several other metrics and points of discussion such as user expertise, time taken to label relevant and non-relevant features and so on, which we reserve for future work. One important consideration though, is that document length influences document labeling time. We found the two to be correlated by $r = 0.289$ which indicates a small increase in time for a large increase in length. The standard deviations for precision and recall are 0.14 and 0.15 respectively. Different users vary significantly in precision, recall and the total number of features labeled relevant. Based on feedback on a post-labeling survey we are inclined to believe that this is due to individual caution exercised during the labeling process.

Some of the highlights of the post-labeling survey are as follows. On average users found the ease of labeling features to be 3.8 (where 0 is most difficult and 5 is very easy) and documents 4.2. In general users with poor prior knowledge found the feature labeling process very hard, although as we will show, their labels were extremely useful to the classifier. The average expertise (5=expert) was 2.4, indicating that most users felt they had little domain knowledge for the tasks they were assigned. We now proceed to see how to use features labeled as relevant by our naive users in active learning.

5 A Human in the Loop

We saw in Sec. 3 that feature selection coupled with uncertainty sampling gives us big gains in performance when there are few labeled examples. In Sec. 4 we saw that humans can discern discriminative features with reasonable accuracy. We now describe our approach of applying term and document level feedback simultaneously in active learning.

5.1 InterActive Learning Algorithm

Let documents be represented as vectors $X_i = x_{i1} \dots x_{i|F|}$, where $|F|$ is the total number of features. At each iteration the active learner not only queries the user on an uncertain document, but also presents a list of f features and asks the user to label features which she considers relevant. The features to be displayed to the user are the top f features obtained by ordering the features by information gain. To obtain the information gain values with t labeled instances we trained a classifier on these t labeled instances. Then to compute information gain, we used the 5 top ranked (farthest from the margin) documents from the unlabeled set in addition to the t labeled documents. Using the unlabeled data for term level feedback is very common in information retrieval and is called pseudo-relevance feedback [Salton, 1968].

The user labels some of the f features which he considers discriminative features. Let $\vec{s} = s_1 \dots s_{|F|}$ be a vector containing weights of relevant features. If a feature number i that is presented to the user is labeled as relevant then we set $s_i = a$, otherwise $s_i = b$, where a and b are parameters of the system. The vector $\vec{s}$ is noisier than the real case because in addition to mistakes made by the user we lose out on those features that the user might have considered relevant, had he been presented that feature when we were collecting relevance judgments for features. In a real life scenario this might correspond to the lazy user who labels few features as relevant and leaves some features unlabeled in addition to making mistakes. If a user had labeled a feature as relevant in some past iteration we don't show the user that feature again.

We incorporate the vector $\vec{s}$ as follows. For each X_i in the labeled and unlabeled sets we multiply x_{ij} by s_j to get X'_{ij} . In other words we scale all relevant features by a and non-relevant features by b . We set $a = 10$ and $b = 1$.²

By scaling the important features by a we are forcing the classifier to assign higher weights to these features. We demonstrate this with the following example. Consider a linear SVM, $|F| = 2$ and 2 data points $X_1 = (1, 2)$ and $X_2 = (2, 1)$ with labels $+1$ and -1 respectively. An SVM trained on this input learns a classifier with $w = (-0.599, +0.599)$. Thus both features are equally discriminative. If feature 1 is considered more discriminative by a user, then by our method $X'_1 = (10, 2)$ and $X'_2 = (20, 1)$ and $w' = (0.043, -0.0043)$, thus assigning higher weight to f_1 . Now, this is a “soft” version of the feature selection mechanism of Sec. 3. But in that case the Oracle knew the ideal set of features; we can view those set of experiments as a special case where $b = 0$. We expect that human labels are noisy and we do not want to zero-out potentially relevant features.

5.2 Experiments and Results

To make our experiments repeatable (to compute average performance and for convenience) we simulate user interaction as follows. For each classification problem we maintain a list of features that a user might have considered relevant had he

²We picked our algorithm's parameters based on a preliminary test on 3 topics (baseball, earnings, and acquisitions) using the oracle features of Sec. 3.

been presented that feature. For these lists we used the judgments obtained in Sec. 4. Thus for each of the 5 classification problems we had 2-3 such lists, one per user who judged that topic. For the 10 TDT topics we have topic descriptions as provided by the LDC, and we simulated explicit human feedback on feature relevance as follows: The topic descriptions contain names of people, places and organizations that are key players in this topic in addition to other keywords. We used the words in these topic descriptions as the list of relevant features. Now, given these lists we can perform the simulated HIL (*Human in the Loop*) experiments for 15 classification problems. At each iteration f features are shown to the user. If the feature exists in the list of relevant features, we set the corresponding bit in $\vec{s}$ and proceed with the active learning as in Sec. 5.1. Fig. 4 shows the performance of the HIL experiments. As before we report deficiency, $F1_7$ and $F1_{22}$. As a baseline we also report results for the case when the top 20 features as obtained by the information gain oracle are input to the simulated HIL experiments (this represents what a user with 100% precision and recall would obtain by our method). The Oracle is (as expected) much better than plain uncertainty sampling, on all 3 measures, reinforcing our faith in the algorithm of Sec. 5.1. The performance of the HIL experiments is almost as good as the Oracle, indicating that user input (although noisy) can help improve performance significantly. The only relative poor performance for the HIL simulation is on the average $\mathcal{D}_{42}$ measure for the TDT categories, where we used all the words from the topic descriptions as a proxy for explicit human feedback on features. The plot on the right is of $F1_t(HIL)$ for *hurricane Mitch*. As a comparison $F1_t(ACT)$ is shown. The HIL values are much higher than for plain uncertainty sampling.

We also observed that relevant features were usually spotted in very early iterations. For the *Auto vs Motorcycles* problem, the user has been asked to label 75% (averaged over multiple iterations and multiple users) of the oracle features at some point or the other. The most informative words (as determined by the Oracle) – *car* and *bike* are asked of the user in very early iterations. The label for *car* is always (100% of the times) asked, and 70% of the time the label for this word is asked of the user in the first iteration itself. This is closely followed by the word *bike* which the user is queried on within the first 5 iterations 80% of the time. Most relevant features are queried within 10 iterations which makes us believe that we can stop feature level feedback in 10 iterations or so. When to stop asking questions on both features and documents and switch entirely to documents remains an area for future work.

6 Related Work

Our work is related to a number of areas including query learning, active learning, use of (prior) knowledge and feature selection in machine learning, term-relevance feedback in information retrieval, and human-computer interaction, from which we can cite only a few.

Our proposed method is an instance of query-based learning and an extension of standard (“pool-based”) active learning which focuses on selective sampling of instances (from

a pool of unlabeled data) alone [Cohn *et al.*, 1994]. Although query-based learning can be very powerful in theory [Angluin, 1992], arbitrary queries may be difficult to answer in practice [Baum and Lang, 1992], hence the popularity of pool-based methods, and the motivation for studying the effectiveness and ease of predictive feature identification by humans in our application area. That human prior knowledge can accelerate learning has been investigated by [Paz-zani and Kibler, 1992], but our work differs in techniques (they use prior knowledge to generate horn-clause rules) and applications. [Beineke *et al.*, 2004] uses human prior knowledge of co-occurrence of words to improve classification of product reviews. None of this work, however, considers the use of prior knowledge in the active learning setting. Our work is unique in the field of active learning as we extend the query model to include feature as well as document level feedback. Our study of the human factors (such as quality of feedback and costs) is also a major differentiating theme between our work and previous work in incorporating prior knowledge which did not address this issue, or might have assumed experts in machine learning taking a role in training the system [Schapire *et al.*, 2002; Wu and Srihari, 2004; Godbole *et al.*, 2004]. We only assume knowledge about the topic of interest. Our algorithmic techniques and the studied modes of interaction differ and are worth further comparison.

In both [Wu and Srihari, 2004; Schapire *et al.*, 2002], prior knowledge is given at the outset which leads to a “soft” labeling of the labeled or unlabeled data that is incorporated into training via modified boosting or SVM training. However, in our scheme the user is labeling documents and features simultaneously. We expect that our proposed interactive mode has an advantage over requesting prior knowledge from the outset, as it may be easier for the user to identify/recall relevant features while labeling documents in the collection and being presented with candidate features. The work of [Godbole *et al.*, 2004] puts more emphasis on system issues and focuses on multi-class training rather than a careful analysis of effects of feature selection and human efficacy. Their proposed method is attractive in that it treats features as single term documents that can be labeled by humans, but they also study labeling features before documents (and only in an “oracle” setting, *i.e.*, not using actual human annotators), and do not observe much improvements using their particular method over standard active learning in the single domain (Reuters) they test on.

7 Conclusions and Future Work

We showed experimentally that for learning with few labeled examples, good feature selection is extremely useful. As the number of examples increases, the vocabulary (feature set size) of the system also needs to increase. A teacher, who is not knowledgeable in machine learning, can help accelerate training the system in this early stage, by pointing out potentially important words. We also conducted a user study to see how well naive users performed as compared to a feature oracle. We used our users’ outputs in realistic *human in the loop* experiments and found significant increase in performance.

This paper raises the question of what questions (other than

Dataset	$\mathcal{D}_{42}$			$F1_7$			$F1_{22}$		
	Act	Oracle	HIL	Act	Oracle	HIL	Act	Oracle	HIL
Baseball	0.71	0.41	0.46	0.49	0.63	0.60	0.63	0.79	0.70
Earnings	0.90	0.64	0.64	0.61	0.79	0.73	0.80	0.85	0.86
Auto vs Motor	0.82	0.33	0.60	0.35	0.62	0.60	0.71	0.83	0.73
Hurr. Mitch	0.89	0.38	0.38	0.04	0.46	0.60	0.08	0.63	0.58
talk.politics.mideast	0.49	0.28	0.28	0.14	0.28	0.29	0.32	0.49	0.49
Avg TDT performance	0.86	0.77	0.89	0.09	0.21	0.24	0.18	0.32	0.22

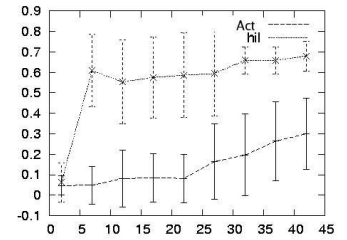


Figure 4: Improvement in deficiency due to human feature selection. Numbers for HIL are averaged over users. The graph shows Human Feature Selection for Hurricane Mitch with the x-axis being the number of labeled documents and y-axis $F1(HIL)$; the difference between these two curves is summarized by the deficiency score. The $F1_7$ and $F1_{22}$ scores show the points on the two curves where 7 and 22 documents have been labeled with active learning. The difference between no feature feedback and human-labeled features is greatest with few documents labeled, but persists up to 42 documents labeled.

questions about the labels of instances) an active learner can ask a user about the domain in order to learn as quickly as possible. In our case, the learner asked the teacher queries on the relevancy of words in addition to the labels of documents. Both types of questions were easy for the teacher to understand. Our subjects did indeed find marking words without context a little hard, and suggested that context might have helped. We intend to conduct a user study, to see what users can perceive easily, and to incorporate these into learning algorithms.

Acknowledgments

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor. We would also like to thank our users who voluntarily labeled data.

References

- [Angluin, 1992] D. Angluin. Comp. learning theory: survey and selected bibl. In *Proc. 24th Annu. ACM Sympos. Theory Comput.*, pages 351–369, 1992.
- [Baram et al., 2003] Y. Baram, R. El-Yaniv, and K. Luz. Online choice of active learning algorithms. In *Proc. of ICML-2003*, pages 19–26, 2003.
- [Baum and Lang, 1992] E. B. Baum and K. Lang. Query learning can work poorly when human oracle is used. In *Intern Joint Conf in Neural Networks*, 1992.
- [Beineke et al., 2004] Philip Beineke, Trevor Hastie, and Shivakumar Vaithyanathan. The sentimental factor: Improving review classification via human-provided information. In *Proc. of ACL’04, Main Volume*, pages 263–270, Barcelona, Spain, July 2004.
- [Brank et al., 2002] J. Brank, M. Grobelnik, N. Milic-Frayling, and D. Mladenic. Feature selection using linear support vector machines. Technical report, Microsoft Research, 2002.
- [Cohn et al., 1994] D. A. Cohn, L. Atlas, and R. E. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.
- [Croft and Das, 1990] W. B. Croft and R. Das. Experiments with query acquisition and use in document retrieval systems. In *SIGIR ’90*, 1990.
- [Godbole et al., 2004] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. In *PKDD*, pages 185–196, 2004.
- [Lewis and Catlett, 1994] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Proc. of ICML-94*, pages 148–156, 1994.
- [McCallum, 1996] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. <http://www.cs.cmu.edu/mccallum/bow>, 1996.
- [Pazzani and Kibler, 1992] M. J. Pazzani and D. Kibler. The role of prior knowledge in inductive learning. *Machine Learning*, 9, 54–97, 9, 1992.
- [Raghavan et al., 2005] Hema Raghavan, Omid Madani, and Rosie Jones. Interactive feature selection. Technical report, University of Massachusetts, Amherst, 2005.
- [Rose et al., 2002] T. G. Rose, M. Stevenson, and M. Whitehead. The reuters corpus vol. 1 - from yesterday’s news to tomorrow’s language resources. In *Inter. Conf. on Lang. Resources and Evaluation*, 2002.
- [Salton, 1968] G. Salton. *Automatic Information Organization and Retrieval*. McGraw Hill, 1968.
- [Schapire et al., 2002] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002.
- [Scholkopf and Smola, 2002] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.
- [Sebastiani, 2002] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 2002.
- [Tong and Koller, 2002] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, 2002.
- [Wu and Srihari, 2004] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proc. of KDD*, 2004.

Generative Modeling with Failure in PRISM

Taisuke Sato, Yoshitaka Kameya

Tokyo Institute of Technology / CREST, JST
2-12-1 Ōokayama Meguro-ku Tokyo Japan 152-8552

Neng-Fa Zhou

The City University of New York
2900 Bedford Avenue, Brooklyn, NY 11210-2889

Abstract

PRISM is a logic-based Turing-complete symbolic-statistical modeling language with a built-in parameter learning routine. In this paper, we enhance the modeling power of PRISM by allowing general PRISM programs to fail in the generation process of observable events. Introducing failure extends the class of definable distributions but needs a generalization of the semantics of PRISM programs. We propose a three valued probabilistic semantics and show how failure enables us to pursue constraint-based modeling of complex statistical phenomena.

1 Introduction

Modeling complex phenomena in the real world such as natural language understanding is a challenging task and PRISM is a symbolic-statistical modeling language proposed for that purpose [Sato and Kameya, 1997]. It is a relational probabilistic language for defining distributions over structured objects with arbitrary complexity. Parameters in the defined distribution (probability measure) are estimated efficiently from data using dynamic programming by a generic EM algorithm¹ called the *gEM* (*graphical EM*) algorithm. For example parameter estimation for PCFGs (probabilistic context free grammars) by PRISM is carried out with the same time complexity as the Inside-outside algorithm, a well-known EM algorithm for PCFGs [Sato and Kameya, 2001].²

The modeling power of PRISM comes from the fact that the user can use arbitrary definite clause programs to define distributions. There is no restriction on programs such as range-restrictedness³ or finite domain. Hence theoretically speaking, there is no limit on the complexity of modeling tar-

gets. Nevertheless EM learning⁴ is possible by the gEM algorithm for a program as long as it satisfies certain conditions. Consequently there is no or little barrier to the EM learning of new statistical models. PRISM thus offers a new and ideal tool to develop complex models that require both logical and statistical knowledge.

PRISM is primarily designed for *generative modeling* where a program models a process of generating observable events as a chain of probabilistic choices. In early versions of PRISM, programs are assumed to be *failure-free*, which means that a process never fails⁵ once a probabilistic choice is made and thus the probabilities of observable events sum to unity. Although typical statistical models such as Bayesian networks, HMMs (hidden Markov models) and PCFGs satisfy this condition, it makes constraint-based modeling, a widely used flexible methodology, impossible in which wrong solutions are filtered out by not satisfying *constraints*.⁶

The elimination of the failure-free condition was attempted in [Sato and Kameya, 2004] but rather limited to the failure of definite clause programs. In this paper we propose to allow *general programs*,⁷ a super class of definite clause programs, to fail. Although the introduction of general programs significantly increases the flexibility of PRISM modeling and expands the class of definable distributions, it comes with costs. First as the probabilities of observable events do not sum to unity, we have to deal with *log-linear models*⁸ [Abney, 1997; Cussens, 2001] which require intractable computation of a normalizing constant. We do not go into the detail of this problem here though.

Second we are faced with a semantic problem. Compared to definite clause programs where the least model semantics is unanimously accepted, there is no agreed-upon semantics

¹The EM algorithm is a class of statistical inference algorithms for ML (maximum likelihood) estimation which alternates expectation and maximization steps until convergence.

²The recent refinement of PRISM implementation [Zhou *et al.*, 2004] has made it possible to complete the parameter estimation for a PCFG containing 860 rules from 10,000 sentences in less than 10 minutes on a PC.

³Variables appearing in the clause head must also appear in the clause body. In particular unit clauses must be ground.

⁴Here and henceforth EM learning is used a synonym of ML estimation by the EM algorithm.

⁵A process fails if it does not lead to any observable output.

⁶Constraints are relations on variables such as equality, disequality and greater-than.

⁷A *general program* is a set of *general clauses* (non-Horn clauses) which may contain negation in their body.

⁸A distribution has the form $p(x) = Z^{-1} \exp(\sum_i \nu_i f_i(x))$ where ν_i is a coefficient, $f_i(x)$ a feature and Z a normalizing constant. In our setting Z is the sum of probabilities of all observable events.

for general programs,⁹ much less probabilistic semantics. But without such semantics, mathematically coherent modeling would be hardly possible. We fill the need by generalizing *distribution semantics*, the existing semantics for *positive* (= definite clause) PRISM programs [Sato, 1995] to a new three-valued semantics. It considers a ground atom as a random variable taking on true, false and undefined, and defines a distribution over the three-valued Herbrand interpretations of a given general PRISM program.

There remains one thing to solve however. To perform EM learning of a general PRISM program that may fail, we need a ‘failure program’ which positively describes how failure occurs in the original program. We show that the failure program can be synthesized mechanically by eliminating negation using FOC (first-order compiler). FOC is a deterministic program transformation algorithm that compiles negated goals into executable form with disunification constraints [Sato, 1989]. Once a failure program is obtained, or more generally negation is eliminated, efficient EM learning is possible by an appropriate modification to the gEM algorithm. Our contributions are summarized as follows.

- A new three-valued semantics for probabilistic logic programs with negation.
- A new approach to parameter learning of such programs through automated elimination of negation by FOC.

In the remainder of this paper we discuss each of the topics after reviewing PRISM briefly. Because our subject intersects a variety of areas from logic programming, statistics, program synthesis, to relational modeling, we are unable to give a detailed and formal description due to the space limitation. Necessarily the description is mostly based on examples and proofs are omitted. The reader is assumed to be familiar with logic programming [Doets, 1994] and EM learning [McLachlan and Krishnan, 1997]. Our notation follows Prolog conventions. So we use ‘,’ for conjunction and ‘;’ for disjunction. Variables begin with an upper case letter and variables in clause heads are implicitly universally quantified.

2 PRISM modeling

Informally PRISM is Prolog augmented with probabilistic built-ins and a parameter learning routine based on the gEM algorithm (the reader is referred to [Sato and Kameya, 2001] for a formal description of PRISM). The essence of PRISM modeling is to write a logic program containing random switches represented by the built-in predicate `msw(t, v)` where t , the name of a switch and v , a result of random choice are ground terms.¹⁰ The sample space for t viewed as a random variable is declared by the predicate `values/2`. The probability of each outcome of a switch, i.e. that of `msw(t, v)` being true, is called a *parameter*. Parameters can be set by the `set_sw/2` predicate or learned from sample data. The following PRISM program, which simulates a sequence of Bernoulli trials, defines a distribution of ground atoms of the form `ber(n, l)` where l is a list of outcomes of n coin tosses.

⁹We can see a variety of semantics in [Apt and Bol, 1994; Fitting, 2002].

¹⁰`msw` is an acronym of ‘multi-ary random switch’.

```
target(ber,2).      % observable event
values(coin,[heads,tails]). % outcomes
:- set_sw(coin,0.6+0.4). % parameters

ber(N,[R|Y]):-
    N>0,
    msw(coin,R), % probabilistic choice
    N1 is N-1,
    ber(N1,Y).   % recursion
ber(0,[]).
```

Figure 1: Bernoulli program

When this program is loaded, the parameters of the switch `coin` is set by `set_sw`: the probability of `heads` is 0.6 and that of `tails` is 0.4. There are three modes of execution in PRISM, namely *sampling execution*, *probability calculation*, and *learning*. In sampling execution, this program is executed like in Prolog except that the built-in `msw(coin, R)` binds probabilistically R to `heads` or `tails`. Probabilities are calculated by using the `prob/2` built-in. For example, the query `?- G = ber(2,[heads,tails]), prob(G,P)` calculates the probability P of having `tails` after having `heads`. Parameters are learned by using the built-in `learn/1` which takes a list of sample data as teacher data.

3 Three-valued distribution semantics

3.1 Two-valued distribution semantics

The Bernoulli program in Figure 1 is said to be *positive* because there is no negation in the program. For such positive PRISM programs, a formal semantics called *distribution semantics* is proposed [Sato, 1995] that defines a σ -additive probability measure¹¹ over the set of possible Herbrand interpretations from which a distribution of each predicate is derived. In the sequel ‘interpretation’ always means an Herbrand interpretation.

The distribution semantics is a probabilistic generalization of the standard least model semantics for definite clause programs. A positive PRISM program is written as $DB = R \cup F$, where R is a set of definite clauses and F a set of ground `msw` atoms to which a *basic distribution* P_{msw} is given. Sampling execution of DB w.r.t. a top-goal is considered as a partial exploration of the least model $M(R \cup F')$ where F' is a set of true `msw` atoms¹² sampled from P_{msw} . In this semantics, each ground atom is a binary random variable taking on **t** (true) or **f** (false). As there are a countably infinite number of ground atoms, we can use an infinite number of random variables in a program. This fact, for example, makes it possible to model PCFGs faithfully.

¹¹We use ‘distribution’ and ‘probability measure’ interchangeably for the sake of familiarity. In this paper the measurability of related functions are assumed without proofs.

¹²An atom whose predicate symbol is `msw` is called an `msw` atom.

3.2 Negation and a new three-valued semantics

The moment negation is introduced, we are faced with numerous problems concerning semantics and computation [Apt and Bol, 1994; Fitting, 2002]. The biggest problem on the semantic side is the lack of the canonical semantics: there exist many competing semantics for general programs. The primary computational problem is that whatever semantics we may adopt, it is not computable in general.

We adopt Fitting's three-valued semantics and extend it to a three-valued probabilistic semantics following [Sato and Kameya, 2001]. Fitting's three-valued semantics is a fixed point semantics similar to the least model semantics for positive programs but goals causing infinite computation are not considered false but considered undefined. It is simpler compared to other semantics such as stable model semantics [Gelfond and Lifshcitz, 1988] and well-founded semantics [Van Gelder *et al.*, 1991]. Also it has an advantage that it always gives a unique denotation to any general program.

Assume that our language $\mathcal{L}$ has countably many predicate and function symbols for each arity. Especially it contains the predicate symbol $\text{msw}/2$. A general PRISM program $DB = R \cup F$ is the union of a set R of general clauses and a set F of ground msw atoms such that no msw atom appears as a head in R . msw atoms are implicitly introduced by $\text{values}/2$ declarations in the program as their ground instantiations.

We associate a probability measure P_{msw} over the set of two-valued interpretations for F as follows. Let t be a switch name, $\Omega_t = \{v_1, \dots, v_k\}$ be a sample space for t declared by $\text{values}(t, [v_1, \dots, v_k])$, and $p_1, \dots, p_k$ are the parameters satisfying $\sum_{i=1}^k p_i = 1$. Let μ_t be a probability measure defined over Ω_t such that $\mu_t(v_i) = p_i$ ($1 \leq i \leq k$). We define an infinite product measure P'_{msw} of μ_t s over $\Omega'_{\text{msw}} = \prod_{t \in \text{SN}} \Omega_t$ where SN is the set of all switch names. Note that $\lambda \in \Omega'_{\text{msw}}$ is considered a function such that $\lambda(t) \in \Omega_t$.

Let Ω_F be the set of interpretations for F . We define a mapping $\varphi : \Omega'_{\text{msw}} \rightarrow \Omega_F$ such that $\varphi(\lambda)(A) = \text{true}$ if $A = \text{msw}(t, v)$ and $\lambda(t) = v$ for some $\text{msw}(t, v)$. Otherwise $\varphi(\lambda)(A) = \text{false}$. Finally we define P_{msw} as the probability measure over Ω_F induced by φ . By construction it holds for the sample space $\{v_1, \dots, v_k\}$ for t that $P_{\text{msw}}(\text{msw}(t, v_i)) = P'_{\text{msw}}(\{\lambda \mid \lambda(t) = v_i, \lambda \in \Omega'_{\text{msw}}\}) = p_i$ and $P_{\text{msw}}(\text{msw}(t, v_i), \text{msw}(t, v_j)) = 0$ if $i \neq j$.

A three-valued interpretation for $\mathcal{L}$ is a mapping which maps a ground atom in $\mathcal{L}$ to **t** (true), **f** (false) or **u** (undefined).¹³ We denote by $\Omega_{3\mathcal{L}}$ the set of three-valued interpretations for $\mathcal{L}$. Let $DB = R \cup F$ be a general PRISM program. For $\nu \in \Omega_F$, we denote by $\mathbf{M}_3(\nu)$ the least fixed point of the Fitting's three-valued direct mapping $\Phi_{\mathcal{P}}$ [Fitting, 2002] applied to the general program $\mathcal{P} = R \cup F_\nu$ where $F_\nu = \{A \mid \nu \models A, A \in F\}$. We consider $\mathbf{M}_3(\nu)$ as a three-valued interpretation for $\mathcal{L}$ where atoms not appearing in DB receive **u**. Then $\mathbf{M}_3(\cdot)$ is a mapping from Ω_F to $\Omega_{3\mathcal{L}}$. $\mathbf{M}_3(\cdot)$ induces a σ -additive probability measure P_{3DB} over $\Omega_{3\mathcal{L}}$. We define P_{3DB} as the denotation of DB in our new three-valued probabilistic semantics (denotational semantics). P_{3DB} is a

¹³The evaluation rule concerning **u** obeys Kleene's strong three valued logic. That is, $\neg \mathbf{u} = \mathbf{u}$, $v \vee \mathbf{u} = \mathbf{u} \vee v = \mathbf{u}$ if $v = \mathbf{f}$ or $\mathbf{u}$; otherwise $\mathbf{u} \vee v = \mathbf{t}$ if $v = \mathbf{t}$.

probabilistic extension of Fitting's three-valued semantics for general logic programs. It is also a generalization of the two-valued counter part P_{DB} defined in [Sato and Kameya, 2001] with a slight complication.¹⁴

4 Failure and log-linear models

Having established a probability measure P_{3DB} in Section 3, we can now talk about various probabilities such as failure probability on a firm mathematical basis. We show here how failure in generative modeling leads to log-linear models.

4.1 Coin flipping with an equality constraint

Suppose we have two biased coins $\text{coin}(a)$ and $\text{coin}(b)$. Also suppose we repeatedly flip them and record the outcome only when both coins agree on their outcome, i.e. both heads or both tails.¹⁵ The task is to estimate probabilities of each coin showing heads or tails from a list of records such as $[\text{agree}(\text{heads}), \text{agree}(\text{tails}), \dots]$.

We generatively model this experiment by a PRISM program R_{agree} shown below where $\text{msw}(\text{coin}(a), A)$ instantiates A to heads or tails with probabilities specified elsewhere. Assuming Prolog execution, $\text{agree}/1$ describes the process of flipping two coins that fails if their outcomes disagree. `success` means there is some output whereas `failure` means there is no output. We denote by $P_{\text{agree}}(\cdot)$ the defined distribution.

```

failure :- not(success).
success :- agree(_).

agree(A) :- % flip two coins and output
    msw(coin(a), A), % the result only when
    msw(coin(b), B), % both outcomes agree
    A=B. % (equality constraint).

```

Figure 2: A general PRISM program R_{agree}

We say a logic program is *terminating* if an arbitrary ground atom has a finite SLDNF tree [Doets, 1994]. Since $P_{\text{agree}} = R_{\text{agree}} \cup F'$ is terminating for an arbitrary set F' of msw atoms, no atom in the least model defined by the Fitting's $\Phi_{\mathcal{P}_{\text{agree}}}$ operator receives **u** (**u** implies an infinite computation). Consequently $P_{\text{agree}}(\text{success}) + P_{\text{agree}}(\text{failure}) = 1$ holds for any parameter setting of msw atoms. On the other hand if some ground atom A receives **u** in our semantics, we may not be able to compute $P_{3DB}(A)$ due to infinite computation. We therefore focus on terminating programs in the following so that no ground atom receives **u** and $P_{3DB}(A)$ is computable for every ground atom A .

¹⁴Suppose that DB is a definite PRISM program and let A be a ground atom. We have $P_{DB}(A = \mathbf{t}) = P_{3DB}(A = \mathbf{t})$ but not necessarily $P_{DB}(A = \mathbf{f}) = P_{3DB}(A = \mathbf{f})$. The reason is that when A causes only an infinite loop, it is false in the least model semantics whereas it is undefined in Fitting's three-valued semantics.

¹⁵This model is a very simplified model of linguistic constraints such as agreement in gender and number [Abney, 1997].

4.2 ML estimation for failure models

When we attempt ML estimation from observed data such as `[agree(heads), agree(tails), ...]`, we must recall that our observation is conditioned on success because we cannot observe failure. Therefore the distribution to be used is a log-linear model $P_{\text{agree}}(\cdot | \text{success}) (= P_{\text{agree}}(\cdot) / P_{\text{agree}}(\text{success}))$ in which the normalizing constant $P_{\text{agree}}(\text{success})$ is not necessarily unity.

Generally speaking ML estimation of parameters for log-linear models is computationally intractable. On the other hand if they are defined by generative processes with failure like $P_{\text{agree}}(\cdot | \text{success})$ in our example, a simpler estimation algorithm called the FAM algorithm is known [Cussens, 2001]. It is an EM algorithm regarding the number of occurrences of failure as a hidden variable. Recently, it is found that it is possible to combine the FAM's ability of dealing with failure and the gEM's dynamic programming approach and an amalgamated algorithm called the *fgEM algorithm* is proposed as a generic algorithm for EM learning of generative models with failure [Sato and Kameya, 2004]. It however requires a positive program that describes how failure occurs in the model. We next explain such a failure program can be automatically synthesized by FOC (first-order compiler).

5 FOC for probabilistic logic programs

5.1 Compiling universally quantified implications

FOC (first-order compiler) is a logic program synthesis algorithm which deterministically compiles negation, or more generally¹⁶ universally quantified implications of the form $\forall y (p(x, y) \rightarrow q(y, z))$ into definite clauses with disequality (disunification) constraints such as $\neg \exists x (s = t[x])$ [Sato, 1989]. Since what FOC does is deductive unfold/fold transformation, the compiled code is a logical consequence of the source program. More precisely we can prove from [Sato, 1989]

Theorem 1 *Let S be a source program consisting of clauses whose bodies include universally quantified implications. Suppose S is successfully compiled into S' by FOC and S' is terminating. Then for a ground atom A ,*

$\text{comp}(S) \vdash A$ (resp. $\neg A$) iff $\text{comp}(S') \vdash A$ (resp. $\neg A$).¹⁷

Figure 3 is a small example of negation elimination by FOC. As can be seen, ‘not’ in the source program is compiled away and a new function symbol `f0` and a new predicate symbol `closure_even0/2` are introduced during the compilation process. They are introduced to handle so called continuation. Usually the compilation adds disunification constraints but in the current case, they are replaced by a normal unification using sort information $x = 0 \vee \exists y (x = s(y))$ declared for FOC (not included in the source program). Since the compiled program is terminating, Theorem 1 applies.

5.2 Modifying FOC for msw

FOC was originally developed for non-probabilistic logic programs. When programs contain `msw` atoms and hence

¹⁶ $\neg A = (A \rightarrow \text{false})$

¹⁷ $\text{comp}(S)$ denotes the union of S and some additional formulas reflecting Prolog's top-down proof procedure [Doets, 1994].

```
% source program
even(0).
even(s(X)) :- not(even(X)).

% compiled program
even(0).
even(s(A)) :- closure_even0(A, f0).
closure_even0(s(A), _) :- even(A).
```

Figure 3: Compilation of `even` program

probabilistic however, compilation is still possible. Suppose we compile a universally quantified implication

$$\forall y (\text{msw}(s, t[y]) \rightarrow \phi[y])$$

into an executable formula where s and t are terms and ϕ is some formula such that t and ϕ possibly contain y . Although `msw/2` is a probabilistic predicate, our (three-valued) distribution semantics allows us to treat it as if it were a non-probabilistic predicate defined by some single ground atom, say `msw(s, t')`. We therefore modify FOC so that it complies¹⁸ the above formula into

$$\text{msw}(s, W), (\setminus + (W = t); W = t, \phi)$$

where W is a new variable. *We can prove this compilation preserves our three-valued semantics when the compiled program is terminating using Theorem 1 (proof omitted).* Figure 4 shows the compiled code produced by the modified FOC for failure in the R_{agree} program in Figure 2.

```
failure :- closure_success0(f0).
closure_success0(A) :- closure_agree0(A).
closure_agree0(_):-
    msw(coin(a), A),
    msw(coin(b), B),
    \+A=B.
```

Figure 4: The compiled program for R_{agree} (part)

Because this program is terminating for any sampled `msw` atoms, it exactly computes $P_{\text{agree}}(\text{failure})$.

In what follows we show two examples of generative modeling with failure. They are available at <http://sato-www.cs.titech.ac.jp/prism/>.

6 Dieting professor

In this section, we present a new type of HMMs: HMMs with constraints which may fail if transition paths or emitted symbols do not satisfy given constraints. Here is an example.

¹⁸We here assume that s is instantiated to some switch name at execution time.

Suppose there is a professor who takes lunch at one of two restaurants ‘r0’ and ‘r1’ everyday and probabilistically changes the restaurant he visits. As he is on a diet, he tries to satisfy a constraint that the total calories for lunch in a week are less than 4000 calories. He probabilistically orders pizza (900) or sandwich (400) at r0, and hamburger (500) or sandwich (500) at r1 (numbers are calories). He records what he has eaten in a week like [p, s, s, p, h, s, h] and preserves the record if and only if he has succeeded in satisfying the constraint. We have a list of the preserved records and we wish to estimate his behavioral probabilities from it.

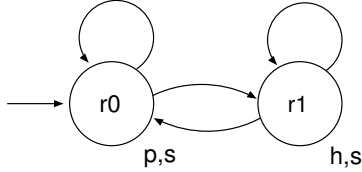


Figure 5: An HMM model for probabilistic choice of lunch

The behavior of the dieting professor is modeled by a constrained HMM described above and coded as a PRISM program below where failing to satisfy the calorie constraint on the seventh day causes failure thereby producing no output.

```

failure:- not(success).
success:- hmmf(L,r0,0,7).

hmmf(L,R,C,N):- N>0,
  msw(tr(R),R2), msw(lunch(R),D),
  ( R == r0, % pizza:900, sandwich:400
    ( D = p, C2 is C+900
      ; D = s, C2 is C+400 )
    ; R == r1, % hamburger:400, sandwich:500
    ( D = h, C2 is C+400
      ; D = s, C2 is C+500 ) ),
  L=[D|L2], N2 is N-1,
  hmmf(L2,R2,C2,N2).
hmmf([],_,C,0):- C < 4000.

```

Figure 6: Constrained HMM for the dieting professor

We verified the correctness of our modeling empirically by checking if the sum of probabilities of success and failure becomes unity. We used the ‘original value’ given in Table 1 as checking parameters.

```

?- prob(success,P_s),prob(failure,P_f),
   X is P_s+P_f.
X = 1.0
P_f = 0.348592596784
P_s = 0.651407403216

```

After checking that the sum is unity, we repeated learning experiments. In an experiment we generated 500 samples by the program with the same parameter values used in the checking, and then let the program estimate them using the

fgEM algorithm. Table 1 shows averages of 50 experiments where numbers should be read that for example, the probability of $msw(tr(r0),r0)$ is originally 0.7 and the average of estimated ones is 0.697. Estimated parameters look close enough to the original ones.

sw name	original value		average estimation	
tr(r0)	r0 (0.7)	r1 (0.3)	r0 (0.697)	r1 (0.303)
tr(r1)	r1 (0.7)	r0 (0.3)	r1 (0.701)	r0 (0.299)
lunch(r0)	p (0.4)	s (0.6)	p (0.399)	s (0.601)
lunch(r1)	h (0.5)	s (0.5)	h (0.499)	s (0.501)

Table 1: EM learning

We would like to emphasize that the computational strength of our approach is that the probability of failure is computed in a dynamic programming manner with no additional cost just like ordinary HMMs thanks to the tabling mechanism of PRISM [Zhou *et al.*, 2004].

7 Tic-tac-toe game

This example models the tic-tac-toe game by using negation and recursion. The purpose of this example is to illustrate ‘EM learning through negative goals’, which has never been attempted before to our knowledge. In tic-tac-toe, the ‘o’ player has to place three ‘o’s in a row, horizontally, vertically, or diagonally on the board of a 9 by 9 board before the opponent, the ‘x’ player, does so with ‘x’. The program in Figure 7 models the game. The basic idea of modeling is expressed by the following clause:

```
win(X):- your_opponent(X,Y),not(win(Y))
```

which states that unless your opponent wins, you win (or draw). In Figure 7, ‘*’ designates an

```

failure :- not(success).
success :-
  Ini_Board = [[*,*,*],[*,*,*],[*,*,*]],
  tic_tac_toe(o,Ini_Board).

tic_tac_toe(Turn,Board):-
  select_move(Turn,Board,Move),
  next_board(Turn,Board,Move,N_Board,S),
  ( S \== continue
    ; S == continue,
      opposite_turn(Turn,N_Turn),
      opponent_turn(N_Turn,N_Board) ).
opponent_turn(N_Turn,N_Board):-
  not(tic_tac_toe(N_Turn,N_Board)).

```

Figure 7: Tic-tac-toe model

empty cell. The intended meaning of success is ‘o’ wins or ‘o’ draws, so success fails if ‘x’ wins. `select_move(Turn,Board,Move)` probabilistically selects and returns the next move (Move), given Turn and Board,

after which `next_board`(Turn, Board, Move, NBoard, S) judges the next board (NBoard) and returns one of the outcomes (`o_win`, `draw`, `x_win`, `continue`) through S. When compiled by FOC, this program yields a terminating program, and hence the probabilities of success and failure are correctly computed by the compiled program.¹⁹

Negative recursion in logic programs causes well-known difficulties from the semantics to the implementation even in the non-probabilistic case. Nevertheless, this example shows that the combination of the three-valued probabilistic semantics and compilation by FOC of general PRISM programs offers a promising approach to dealing with negation in the probabilistic context.

8 Related work and conclusion

We only discuss related work combining negation and probability in the framework of logic programming due to the space limitation and other work related to stochastic logic programming, Bayesian networks and Markov random field is omitted. The most relevant one seems the formalism proposed by Ngo and Haddawy [Ngo and Haddawy, 1997]. In their approach negation is allowed in a program but restricted to non-probabilistic goals. Semantics is defined locally in relation to a given problem. So no global semantics such as a probability measure over the set of possible Herbrand interpretations is given. The problem of parameter learning is not discussed either. Poole allows negation in his ICL (independent choice logic) programs and probabilities can be computed through negative goals [Poole, 1997]. Stable model semantics is adopted as global semantics but the problem of learning is left open.

In this paper we first tackled the problem of defining declarative semantics for probabilistic general logic programs that contain negative goals, and have proposed a new three-valued semantics which is unconditionally definable for any programs unlike other approaches. We then tackled the second problem, namely parameter learning. We have shown how parameter learning is made possible by negation elimination through the modified FOC (first-order compiler) that preserves the new semantics. As a result, constraints that may cause failure can now be utilized as building blocks of even more complex symbolic-statistical models such as generative HPSG models which is to be reported elsewhere.

References

- [Abney, 1997] S. Abney. Stochastic attribute-value grammars. *Computational Linguistics*, 23(4):597–618, 1997.
- [Apt and Bol, 1994] K.R. Apt and R.N. Bol. Logic programming and negation: A survey. *Journal of Logic Programming*, 19/20:9–71, 1994.
- [Cussens, 2001] J. Cussens. Parameter estimation in stochastic logic programs. *Machine Learning*, 44(3):245–271, Sept. 2001.
- [Doets, 1994] K. Doets. *From Logic to Logic Programming*. The MIT Press, 1994.
- [Fitting, 2002] M. Fitting. Fixpoint Semantics for Logic Programming, A Survey. *Theoretical Computer Science*, 278(1-2):25–51, 2002.
- [Gelfond and Lifshcitz, 1988] M. Gelfond and V. Lifshcitz. The stable model semantics for logic programming. pages 1070–1080, 1988.
- [McLachlan and Krishnan, 1997] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley Interscience, 1997.
- [Ngo and Haddawy, 1997] L. Ngo and P. Haddawy. Answering queries from context-sensitive probabilistic knowledge bases. *Theoretical Computer Science*, 171:147–177, 1997.
- [Poole, 1997] D. Poole. The independent choice logic for modeling multiple agents under uncertainty. *Artificial Intelligence*, 94(1-2):7–56, 1997.
- [Sato and Kameya, 1997] T. Sato and Y. Kameya. PRISM: a language for symbolic-statistical modeling. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI’97)*, pages 1330–1335, 1997.
- [Sato and Kameya, 2001] T. Sato and Y. Kameya. Parameter learning of logic programs for symbolic-statistical modeling. *Journal of Artificial Intelligence Research*, 15:391–454, 2001.
- [Sato and Kameya, 2004] T. Sato and Y. Kameya. A dynamic programming approach to parameter learning of generative models with failure. In *Proceedings of ICML 2004 workshop on Learning Statistical Models from Relational Data (SRL2004)*, 2004.
- [Sato, 1989] T. Sato. First order compiler: A deterministic logic program synthesis algorithm. *Journal of Symbolic Computation*, 8:605–627, 1989.
- [Sato, 1995] T. Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming (ICLP’95)*, pages 715–729, 1995.
- [Van Gelder et al., 1991] A. Van Gelder, K.A. Ross, and J.S. Schlipf. The well-founded semantics for general logic programs. *The journal of ACM (JACM)*, 38(3):620–650, 1991.
- [Zhou et al., 2004] Neng-Fa Zhou, Y. Shen, and T. Sato. Semi-naive Evaluation in Linear Tabling. In *Proceedings of the Sixth ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP2004)*, pages 90–97, 2004.

¹⁹As this program merely simulates moves until ‘o’ wins or draws without recording them, it is not usable to train parameters from the recorded moves of a game. We therefore wrote another program that can generate a list of moves through negative goals and conducted an EM learning experiment and confirmed its success.

Evolino: Hybrid Neuroevolution / Optimal Linear Search for Sequence Learning

Jürgen Schmidhuber^{1,2}, Daan Wierstra², and Faustino Gomez²
{juergen, daan, tino}@idsia.ch

¹ TU Munich, Boltzmannstr. 3, 85748 Garching, München, Germany

² IDSIA, Galleria 2, 6928 Lugano, Switzerland

Abstract

Current Neural Network learning algorithms are limited in their ability to model non-linear dynamical systems. Most supervised gradient-based recurrent neural networks (RNNs) suffer from a vanishing error signal that prevents learning from inputs far in the past. Those that do not, still have problems when there are numerous local minima. We introduce a general framework for sequence learning, EVolution of recurrent systems with LINear outputs (Evolino). Evolino uses evolution to discover good RNN hidden node weights, while using methods such as linear regression or quadratic programming to compute optimal linear mappings from hidden state to output. Using the Long Short-Term Memory RNN Architecture, the method is tested in three very different problem domains: 1) context-sensitive languages, 2) multiple superimposed sine waves, and 3) the Mackey-Glass system. Evolino performs exceptionally well across all tasks, where other methods show notable deficiencies in some.

1 Introduction

Real world non-linear dynamical systems are black-box in nature: it is possible to observe their input/output behavior, but the internal mechanism that generates this behavior is often unknown. Modeling such systems to accurately predict their behavior is a huge challenge with potentially far-reaching impact on areas as broad as speech processing/recognition, financial forecasting, and engineering.

Artificial Neural Networks with feedback connections or Recurrent Neural Networks (RNNs; [Werbos, 1990; Robinson and Fallside, 1987; Williams and Zipser, 1989]) are an attractive formalism for non-linear modeling because of their ability, in principle, to approximate any dynamical system with arbitrary precision [Siegelmann and Sontag, 1991]. However, training RNNs with standard gradient descent algorithms is only practical when a short time window (less than 10 time-steps) is sufficient to predict the correct system output. For longer temporal dependencies, the gradient vanishes as the error signal is propagated back through time so that network weights are never adjusted correctly to account for events far in the past [Hochreiter *et al.*, 2001].

Echo State Networks (ESNs; [Jaeger, 2004a]) deal with temporal dependencies by simply ignoring the gradients associated with hidden neurons. Composed primarily of a large pool of neurons (typically hundreds or thousands) with fixed random weights, ESNs are trained by computing a set of weights analytically from the pool to the output units using fast, linear regression. The idea is that with so many random hidden units, the pool is capable of very rich dynamics that just need to be correctly “tapped” by adjusting the output weights. This simple approach is currently the title holder in the Mackey-Glass time-series benchmark, improving on the accuracy of all other methods by as much as three orders of magnitude [Jaeger, 2004a].

The drawback of ESNs, of course, is that the only truly computationally powerful, nonlinear part of the net does not learn at all. This means that on some seemingly simple tasks, such as generating multiple superimposed sine waves, the method fails. According to our experience, it is also not able to solve a simple context-sensitive grammar task [Gers and Schmidhuber, 2001]. Moreover, because ESNs use such a large number of processing units, they are prone to overfitting, i.e. poor generalization.

One method that adapts *all* weights and succeeds in using gradient information to learn long-term dependencies is Long Short-Term Memory (LSTM; [Hochreiter and Schmidhuber, 1997; Gers and Schmidhuber, 2001]). LSTM uses a specialized network architecture that includes linear *memory cells* that can sustain their activation indefinitely. The cells have input and output gates that learn to open and close at appropriate times either to let in new information from outside and change the state of the cell, or to let activation out to potentially affect other cells or the network’s output. The cell structure enables LSTM to use gradient descent to learn dependencies across almost arbitrarily long time spans. However, in cases where gradient information is of little use due to numerous local minima, LSTM becomes less competitive.

An alternative approach to training RNNs is neuroevolution [Yao, 1999]. Instead of using a single neural network, the space of network parameters is searched in parallel using the principle of natural selection. A population of *chromosomes* or strings encoding, for instance, network weight values and connectivity is evaluated on the problem, and each chromosome is awarded a *fitness* value that quantifies its relative performance. The more highly fit chromosomes are combined by exchanging substrings (*crossover*) and by randomly changing some values (*mutation*), producing new so-

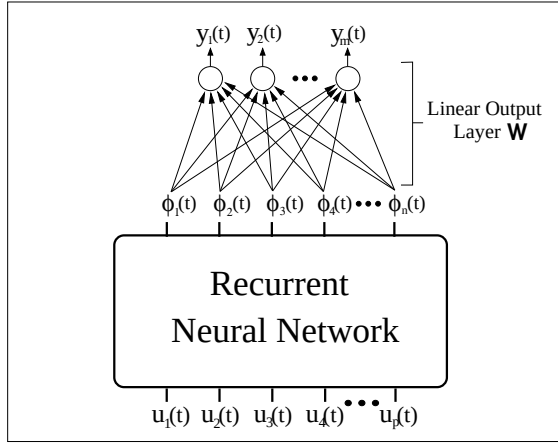


Figure 1: **Evolino network.** A recurrent neural network receives sequential inputs $u(t)$ and produce the vector $(\phi_1, \phi_2, \dots, \phi_n)$ at every time step t . These values are linearly combined with the weight matrix W to yield the network’s output vector $y(t)$. While the RNN is evolved, the output layer weights are computed using a fast, optimal method such as linear regression or quadratic programming.

lutions that hopefully improve upon the existing population. This approach has been very effective in solving continuous, partially observable reinforcement learning tasks where the gradient is not directly available, outperforming conventional methods (e.g. Q-learning, SARSA) on several difficult learning benchmarks [Moriarty and Miikkulainen, 1996; Gomez and Miikkulainen, 1999]. However, neuroevolution is rarely used for supervised learning tasks such as time series prediction because it has difficulty fine-tuning solution parameters (e.g. network weights), and because of the prevailing maxim that gradient information should be used when it is available.

In this paper, we present a novel framework called EVOlution of recurrent systems with LINear outputs (Evolino) that combines elements of the three aforementioned methods, to address the disadvantages of each, extending ideas proposed for feedforward networks of radial basis functions (RBFs) [Maillard and Gueriot, 1997]. Applied to the LSTM architecture, Evolino can solve tasks that ESNs cannot, and achieves higher accuracy in certain continuous function generation tasks than conventional gradient descent RNNs, including gradient-based LSTM.

Section 2 explains the basic concept of Evolino and describes in detail the specific implementation used in this paper. Section 3 presents our experiments using Evolino in three different domains: context-sensitive grammars, continuous function generation, and the Mackey-Glass time-series. Section 4 and 5 discuss the algorithm and the experimental results, and summarize our conclusions.

2 The Evolino Framework

Evolino is a general framework for supervised sequence learning that combines neuroevolution (i.e. the evolution of neural networks) and analytical linear methods that are optimal in some sense, such as linear regression or quadratic programming. The underlying principle of Evolino is that often a linear model can account for a large number of properties of a

problem. Properties that require non-linearity and recurrence are then dealt with by evolution.

Figure 1 illustrates the basic operation of an Evolino network. The output of the network at time t , $y(t) \in \mathbb{R}^m$, is computed by the following formulas:

$$y(t) = W\phi(t), \quad (1)$$

$$\phi(t) = f(u(t), u(t-1), \dots, u(0)), \quad (2)$$

where $\phi(t) \in \mathbb{R}^n$ is the output of a recurrent neural network $f(\cdot)$, and W is a weight matrix. Note that because the networks are recurrent, $f(\cdot)$ is indeed a function of the entire input history, $u(t), u(t-1), \dots, u(0)$. In the case of maximum margin classification problems [Vapnik, 1995] we may compute W by quadratic programming. In what follows, however, we focus on mean squared error minimization problems and compute W by linear regression.

In order to evolve an $f(\cdot)$ that minimizes the error between y and the correct output, d , of the system being modeled, Evolino does not specify a particular evolutionary algorithm, but rather only stipulates that networks be evaluated using the following two-phase procedure.

In the first phase, a training set of sequences obtained from the system, $\{u^i, d^i\}$, $i = 1..k$, each of length l^i , is presented to the network. For each sequence u^i , starting at time $t = 0$, each input pattern $u^i(t)$ is successively propagated through the recurrent network to produce a vector of activations $\phi^i(t)$ that is stored as a row in a $n \times \sum_i l^i$ matrix Φ . Associated with each $\phi^i(t)$, is a *target* row vector d^i in D containing the correct output values for each time step. Once all k sequences have been seen, the output weights W (the output layer in figure 1) are computed using linear regression from Φ to D . The column vectors in Φ (i.e. the values of each of the n outputs over the entire training set) form a non-orthogonal basis that is combined linearly by W to approximate D .

In the second phase, the training set is presented to the network again, but now the inputs are propagated through the recurrent network $f(\cdot)$ and the newly computed output connections to produce predictions $y(t)$. The error in the prediction or the *residual error* is then used as the fitness measure to be minimized by evolution.

Neuroevolution is normally applied to reinforcement learning tasks where correct network outputs (i.e. targets) are not known *a priori*. Here we use neuroevolution for supervised learning to circumvent the problems of gradient-based approaches. In order to obtain the precision required for time-series prediction, we do not try to evolve a network that makes predictions directly. Instead, the network outputs a set of vectors that form a basis for linear regression. The intuition is that finding a sufficiently good basis is easier than trying to find a network that models the system accurately on its own.

In this study, Evolino is instantiated using Enforced SubPopulations to evolve LSTM networks. The next sections describe ESP and LSTM, and the details of how they are combined within the Evolino framework.

2.1 Enforced Subpopulations

Enforced SubPopulations differs from standard neuroevolution methods in that instead of evolving complete networks, it *coevolves* separate subpopulations of network components or *neurons* (figure 2). Evolution in ESP proceeds as follows:

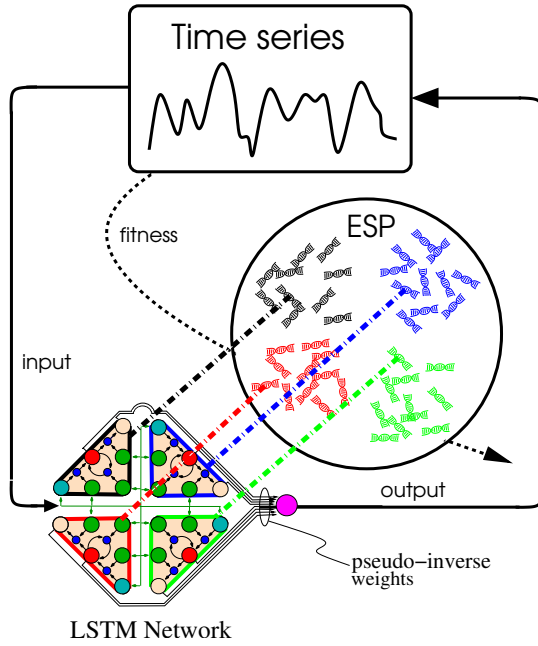


Figure 2: **Enforced SubPopulations (ESP)**. The population of neurons is segregated into subpopulations. Networks are formed by randomly selecting one neuron from each subpopulation. A neuron accumulates a fitness score by adding the fitness of each network in which it participated. The best neurons within each subpopulation are mated to form new neurons. The network shown here is an LSTM network with four memory cells (the triangular shapes).

1. Initialization: The number of hidden units H in the networks that will be evolved is specified and a subpopulation of n neuron chromosomes is created for each hidden unit. Each chromosome encodes a neuron's input, output, and recurrent connection weights with a string of random real numbers.
2. Evaluation: A neuron is selected at random from each of the H subpopulations, and combined to form a recurrent network. The network is evaluated on the task and awarded a fitness score. The score is added to the *cumulative fitness* of each neuron that participated in the network.
3. Recombination: For each subpopulation the neurons are ranked by fitness, and the top quartile is recombined using 1-point crossover and mutated using Cauchy distributed noise to create new neurons that replace the lowest-ranking half of the subpopulation.
4. Repeat the Evaluation–Recombination cycle until a sufficiently fit network is found.

ESP searches the space of networks indirectly by sampling the possible networks that can be constructed from the subpopulations of neurons. Network evaluations serve to provide a fitness statistic that is used to produce better neurons that can eventually be combined to form a successful network. This cooperative coevolutionary approach is an extension to Symbiotic, Adaptive Neuroevolution (SANE; [Moriarty and Miikkulainen, 1996]) which also evolves neurons, but in a single population. By using separate subpopulations, ESP accelerates the specialization of neurons into different sub-

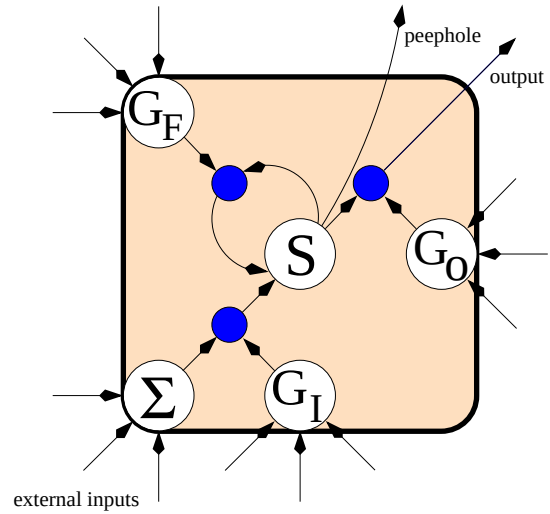


Figure 3: **Long Short-Term Memory**. The figure shows an LSTM memory cell. The cell has an internal state S together with a forget gate (G_F) that determines how much the state is attenuated at each time step. The input gate (G_I) controls access to the cell by the external inputs that are summed into the Σ unit, and the output gate (G_O) controls when and how much the cell fires. Small dark nodes represent the multiplication function.

functions needed to form good networks because members of different evolving sub-function types are prevented from mating. Subpopulations also reduce noise in the neuron fitness measure because each evolving neuron type is guaranteed to be represented in every network that is formed. This allows ESP to evolve recurrent networks, where SANE could not.

If the performance of ESP does not improve for a predetermined number of generations, a technique called *burst mutation* is used. The idea of burst mutation is to search the space of modifications to the best solution found so far. When burst mutation is activated, the best neuron in each subpopulation is saved, the other neurons are deleted, and new neurons are created for each subpopulation by adding Cauchy distributed noise to its saved neuron. Evolution then resumes, but now searching in a neighborhood around the previous best solution. Burst mutation injects new diversity into the subpopulations and allows ESP to continue evolving after the initial subpopulations have converged.

2.2 Long Short-Term Memory

LSTM is a recurrent neural network purposely designed to learn long-term dependencies via gradient descent. The unique feature of the LSTM architecture is the *memory cell* that is capable of maintaining its activation indefinitely (figure 3). Memory cells consist of a linear unit which holds the *state* of the cell, and three gates that can open or close over time. The input gate “protects” a neuron from its input: only when the gate is open, can inputs affect the internal state of the neuron. The output gate lets the state out to other parts of the network, and the forget gate enables the state to “leak” activity when it is no longer useful.

The state of cell i is computed by:

$$s_i(t) = net_i(t)g_i^{in}(t) + g_i^{forget}(t)s_i(t-1), \quad (3)$$

where g^{in} and g^{forget} are the activation of the input and forget gates, respectively, and net is the weighted sum of the external inputs (indicated by the Σ s in figure 3):

$$net_i(t) = h\left(\sum_j w_{ij}^{cell} c_j(t-1) + \sum_k w_{ik}^{cell} u_k(t)\right), \quad (4)$$

where h is usually the identity function, and c_j is the output of cell j :

$$c_j(t) = \tanh(g_j^{out}(t) s_j(t)). \quad (5)$$

where g^{out} is the output gate of cell j . The amount each gate g_i of memory cell i is open or closed at time t is calculated by:

$$g_i^{type}(t) = \sigma\left(\sum_j w_{ij}^{type} c_j(t-1) + \sum_k w_{ik}^{type} u_k(t)\right), \quad (6)$$

where $type$ can be *input*, *output*, or *forget*, and σ is the standard sigmoid function. The gates receive input from the output of other cells c_j , and from the external inputs to the network.

2.3 Combining ESP with LSTM in Evolino

We apply our general Evolino framework to the LSTM architecture, using ESP for evolution and regression for computing linear mappings from hidden state to outputs. ESP co-evolves subpopulations of memory cells instead of standard recurrent neurons (figure 2). Each chromosome is a string containing the external input weights and the input, output, and forget gate weights, for a total of $4 * (I + H)$ weights in each memory cell chromosome, where I is the number of external inputs and H is the number of memory cells in the network. There are four sets of $I + H$ weights because the three gates (equation 6) and the cell itself (equation 4) receive input from outside the cell and the other cells. ESP, as described in section 2.1, normally uses crossover to recombine neurons. However, for the present Evolino variant, where fine local search is desirable, ESP uses only mutation. The top quarter of the chromosomes in each subpopulation are duplicated and the copies are mutated by adding Cauchy noise to all of their weight values.

The linear regression method used to compute the output weights (W in equation 2) is the Moore-Penrose pseudo-inverse method, which is both fast and optimal in the sense that it minimizes the summed squared error [Penrose, 1955]—compare [Maillard and Gueriot, 1997] for an application to feedforward RBF nets. The vector $\phi(t)$ consists of both the cell outputs, c_i (equation 5), and their internal states, s_i (equation 3), so that the pseudo-inverse computes two connection weights for each memory cell. We refer to the connections from internal states to the output units as “output peephole” connections, since they peer into the interior of the cells.

For continuous function generation, *backprojection* (or *teacher forcing* in standard RNN terminology) is used where the predicted outputs are fed back as inputs in the next time step: $\phi(t) = f(u(t), y(t-1), u(t-1), \dots, y(0), u(0))$.

During training, the correct target values are backprojected, in effect “clamping” the network’s outputs to the right values. During testing, the network backprojects its own predictions. This technique is also used by ESNs, but whereas ESNs do not change the backprojection connection weights, Evolino evolves them, treating them like any other input to the network. In the experiments described below, backprojection

Training data	Gradient LSTM	Evolino LSTM
1..10	1..28	1..53
1..20	1..66	1..95
1..30	1..91	1..355
1..40	1..120	1..804

Table 1: **Generalization results for the $a^n b^n c^n$ language.** The table compares Evolino-based LSTM to Gradient-based LSTM. The left column shows the set of legal strings used to train each method. The other columns show the set of strings that each method was able to accept after training. The result for LSTM with gradient descent are from [Gers and Schmidhuber, 2001]. Averages of 20 runs.

was found useful for continuous function generation tasks, but interferes to some extent with performance in the discrete context-sensitive language task.

3 Experimental Results

Experiments were carried out on three test problems: context-sensitive languages, multiple superimposed sine waves, and the Mackey-Glass time series. The first two were chosen to highlight Evolino’s ability to perform well in both discrete and continuous domains. For a more detailed description of setups used in these two problems, and further experiments, we direct the reader to [Wierstra *et al.*, 2005]. The Mackey-Glass system was selected to compare Evolino with ESNs, the reference method on this widely used time series benchmark.

3.1 Context-Sensitive Grammars

Learning to recognize context-sensitive languages is a difficult and often intractable problem for standard RNNs because it can require unlimited memory. For instance, recognizing the language $a^n b^n c^n$ (i.e. strings where the number of *as*, *bs*, and *cs* is equal) entails counting the number of consecutive *as*, *bs*, and *cs*, and potentially having to remember these quantities until the whole string has been read. Gradient-based LSTM has previously been used to learn $a^n b^n c^n$, so here we compare the results in [Gers and Schmidhuber, 2001] to those of Evolino-based LSTM.

Four sets of 20 simulations were run each using a different training set of legal strings, $\{a^n b^n c^n\}$, $n = 1..N$, where N was 10, 20, 30, and 40. Symbol strings were presented to the networks, one symbol at a time. The networks had 4 input units, one for each possible symbol: *S* for start, *a*, *b*, and *c*. An input is set to 1.0 when the corresponding symbol is observed, and -1.0 when it is not present. At every time step, the network predicts what symbols could come next, *a*, *b*, *c*, and the termination symbol *T*, by activating its 4 output units. An output unit is considered to be “on” if its activation is greater than 0.0.

ESP evolved LSTM networks with 4 memory cells, weights randomly initialized to values between -0.1 and 0.1 . The Cauchy noise parameter α for both mutation and burst mutation was set to 0.00001, i.e. 50% of the mutations is kept within this bound. Evolution was terminated after 50 generations, after which the best network in each simulation was tested.

The results are summarized in Table 1. Evolino-based LSTM learns in approximately 3 minutes on average, but, more importantly, it is able to generalize substantially better than gradient-based LSTM.

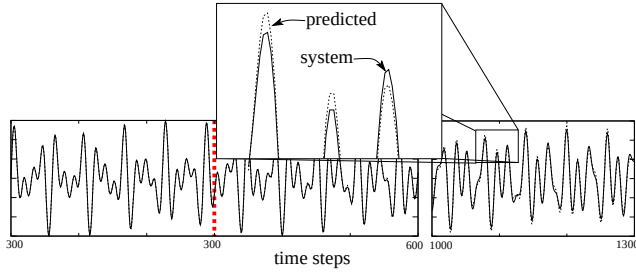


Figure 4: **Performance of Evolino on the triple superimposed sine wave task.** The plot shows the behavior of a typical network produced after 50 generations (3000 evaluations). The first 300 steps (the data-points left of the vertical dashed line) were used as training data, the rest must be predicted by the network during testing. Time-steps above 300 show the network predictions (dashed curve) during testing plotted against the correct system output (solid curve). The inset is a magnified detail that more clearly shows the two curves.

3.2 Multiple Superimposed Sine Waves

Jaeger [Jaeger, 2004b] reports that Echo State Networks are unable to learn functions composed of multiple superimposed oscillators. Specifically, functions like $\sin(0.2x) + \sin(0.311x)$, in which the individual sines have the same amplitude but their frequencies are not multiples of each other. ESNs have difficulty solving this problem because the dynamics of all the neurons in the ESN “pool” are coupled, whereas truly solving the task requires an internal representation of multiple attractors due to the non-periodic behavior of the function.

We evolved networks with 10 memory cells to predict the aforementioned double sine, $\sin(0.2x) + \sin(0.311x)$, and network with 15 cells for a more complex triple sine, $\sin(0.2x) + \sin(0.311x) + \sin(0.42x)$. Evolino used the same parameter settings as in the previous section, except that backprojection was used (see section 2.3). Networks for both tasks were evolved for 50 generations to predict the first 300 time steps of each function, and then tested on data points from time-steps 300..600.

The average summed squared error over the training set was 0.011 for the double sine and 0.2 for the triple sine. The average error over the test set was 0.044 and 1.58, respectively. These error levels are barely visible out to time-step 600. Figure 4 shows the behavior of one of the triple sine wave Evolino networks out to time-step 1300. The magnified inset illustrates how even beyond 3 times the length of the training set, the network still makes very accurate predictions.

3.3 Mackey-Glass Time-Series Prediction

The Mackey-Glass system (MGS; [Mackey and Glass, 1977]) is a standard benchmark for chaotic time series prediction. The system produces an irregular time series that is produced by the following differential equation: $\dot{y}(t) = \alpha y(t - \tau) / (1 + y(t - \tau)^\beta) - \gamma y(t)$, where the parameters are usually set to $\alpha = 0.2, \beta = 10, \gamma = 0.1$. The system is chaotic whenever the delay $\tau > 16.8$. We use the most common value for the delay $\tau = 17$.

Although the MGS can be modeled very accurately using feedforward networks with a time-window on the input, we compare Evolino to ESNs (currently the best method for

MGS) in this domain to show its capacity for making precise predictions. We used the same setup in our experiments as in [Jaeger, 2004a]. Networks were trained on the first 3000 time steps of the series using a “washout time” of 100 steps. During the washout time the vectors $\phi(t)$ are not collected for calculating the pseudo-inverse.

We evolved networks with 30 memory cells for 200 generations, and a Cauchy noise α of 10^{-7} . A bias input of 1.0 was added to the network, and the backprojection values were scaled by a factor of 0.1. For testing, the outputs were clamped to the correct targets for the first 3000 steps, after which the network backprojected its own prediction for the next 84 steps¹. The cell input (equation 4) was squashed with the $\tanh$ function. The average NRMSE_{84} for Evolino with 30 cells over the 15 runs was 1.9×10^{-3} compared to $10^{-4.2}$ for ESNs with 1000 neurons [Jaeger, 2004a]. The Evolino results are currently the second-best reported so far.

Figure 5 shows the performance of an Evolino network on the MG time-series with even fewer memory cells, after 50 generations. Because this network has fewer parameters, it is unable to achieve the same precision as with 30 neurons, but it demonstrates how Evolino can learn complex functions very quickly; in this case within approximately 3 minutes of CPU time.

4 Discussion

The real strength of the Evolino framework is its generality. Across different classes of sequence prediction problems, it was able to compete with the best known methods and convincingly outperform them in several cases. In particular, it generalized much better than gradient-based LSTM in the context-sensitive grammar task, and it solved the superimposed sine wave task, which ESNs cannot. These results suggest that Evolino could be widely applicable to modeling complex processes that have both discrete and continuous properties, such as speech.

Evolino avoids the problem of vanishing gradient and local minima normally associated with RNN training by searching the space of networks in parallel through evolution. Furthermore, by using LSTM memory cells, Evolino searches in a weight space that is already biased toward extracting, retaining, and relating discrete events that may be very far apart in time. And, by borrowing the idea of linear regression from ESNs, Evolino is capable of making very precise predictions in tasks like the Mackey-Glass benchmark.

Apart from its versatility, another advantage of Evolino over ESNs is that it produces more parsimonious solutions. ESNs have large pools of neurons that are more likely to overfit the data. Evolino networks can be made much smaller and, therefore, potentially more general, less susceptible to noise, and more easily comprehensible by, for instance, RNN rule extraction techniques.

Evolino is a template that can be instantiated by plugging in (1) alternative analytical methods for computing optimal linear mappings to the outputs, given the hidden state, (2) different neuroevolution algorithms, and (3) various recurrent network architectures. In particular, our implementation used

¹The normalized root mean square error (NRMSE_{84}) 84 steps after the end of the training sequence, is the standard comparison measure used for this problem.

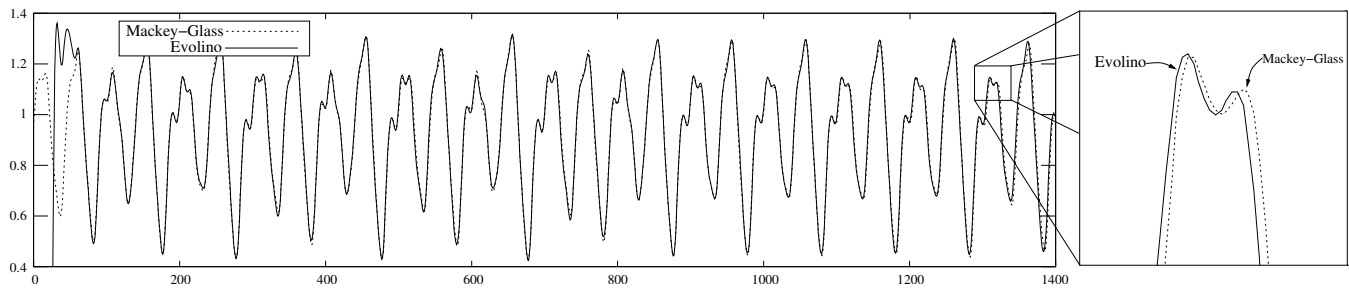


Figure 5: **Performance of Evolino on the Mackey-Glass time-series.** The plot shows both the Mackey-Glass system and the prediction made by a typical Evolino-based LSTM network evolved for 50 generations. The obvious difference between the system and the prediction during the first 100 steps is due to the washout time. The inset shows a magnification more clearly showing the deviation between the two curves.

mean squared error and linear regression, but we could as well use the maximum margin optimality criterion [Vapnik, 1995] and use quadratic programming to find optimal linear mappings from hidden state to sequence classifications, obtaining a hitherto unknown species of *sequential* support vector machines.

We could also use neuroevolution methods that evolve network topology as well, so that network complexity is also determined through genetic search. Other RNNs, such as higher-order networks could be used instead of LSTM. Generalizations to nonlinear readout mechanisms (e.g., nonlinear neural networks) with gradient-based search are obvious. We may also start training LSTM by Evolino, then fine-tune by traditional pure gradient search.

Future work will further explore this space of possible implementations to provide potentially even more powerful predictors, classifiers, and sequence generators.

5 Conclusion

We introduced EVolution of recurrent systems with LINear outputs (Evolino), a general framework that combines evolution of recurrent neural networks and analytical linear methods to solve sequence learning tasks. The implementation of Evolino in this paper combined the pseudo-inverse and Enforced Subpopulations algorithms to search a space of Long-Short Term Memory networks. This yielded a versatile method that can solve both tasks that require long-term memory of discrete events such as context-sensitive languages, and continuous time-series such as the Mackey-Glass benchmark and multiple superimposed sine waves.

Acknowledgments

This research was partially funded by CSEM Alpnach and the EU MindRaces project, FP6 511931.

References

- [Gers and Schmidhuber, 2001] F. A. Gers and J. Schmidhuber. LSTM recurrent networks learn simple context free and context sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001.
- [Gomez and Mikkulainen, 1999] Faustino Gomez and Risto Mikkulainen. Solving non-Markovian control tasks with neuroevolution. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Denver, CO, 1999. Morgan Kaufmann.
- [Hochreiter and Schmidhuber, 1997] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [Hochreiter *et al.*, 2001] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.
- [Jaeger, 2004a] H. Jaeger. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304:78–80, 2004.
- [Jaeger, 2004b] H. Jaeger. <http://www.faculty.iu-bremen.de/hjaeger/courses/seminarspring04/esnstandardslides.pdf>, 2004.
- [Mackey and Glass, 1977] M. C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197:287–289, 1977.
- [Maillard and Gueriot, 1997] E. P. Maillard and D. Gueriot. RBF neural network, basis functions and genetic algorithms. In *IEEE International Conference on Neural Networks*, pages 2187–2190, Piscataway, NJ, 1997. IEEE.
- [Moriarty and Mikkulainen, 1996] D. E. Moriarty and R. Mikkulainen. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32, 1996.
- [Penrose, 1955] R. Penrose. A generalized inverse for matrices. In *Proceedings of the Cambridge Philosophy Society*, volume 51, pages 406–413, 1955.
- [Robinson and Fallside, 1987] A. J. Robinson and F. Fallside. The utility driven dynamic error propagation network. Technical Report CUED/F-INFENG/TR.1, Cambridge University Engineering Department, 1987.
- [Siegelmann and Sontag, 1991] H. T. Siegelmann and E. D. Sontag. Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80, 1991.
- [Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.
- [Werbos, 1990] P. Werbos. Backpropagation through time: what does it do and how to do it. In *Proceedings of IEEE*, volume 78, pages 1550–1560, 1990.
- [Wierstra *et al.*, 2005] Daan Wierstra, Juergen Schmidhuber, and Faustino Gomez. Modeling non-linear dynamical systems with evolino. To appear in *Proceedings of the Genetic Evolutionary Computation Conference (GECCO-05)*. Springer-Verlag, Berlin; New York, 2005.
- [Williams and Zipser, 1989] R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent networks. *Neural Computation*, 1(2):270–280, 1989.
- [Yao, 1999] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

A Multi-Objective Multi-Modal Optimization Approach for Mining Stable Spatio-Temporal Patterns.

Michèle Sebag¹, Nicolas Tarrisson¹, Olivier Teytaud¹, Julien Lefevre² & Sylvain Baillet²

(1) TAO, CNRS – INRIA – Université Paris-Sud, F-91405 - Orsay

(2) LENA, CNRS, La Pitié Salpêtrière, F-75651 - Paris

Abstract

This paper, motivated by functional brain imaging applications, is interested in the discovery of stable spatio-temporal patterns. This problem is formalized as a multi-objective multi-modal optimization problem: on one hand, the target patterns must show a good stability in a wide spatio-temporal region (antagonistic objectives); on the other hand, experts are interested in finding all such patterns (global and local optima). The proposed algorithm, termed *4D-Miner*, is empirically validated on artificial and real-world datasets; it shows good performances and scalability, detecting target spatio-temporal patterns within minutes from 400+ Mo datasets.

1 Introduction

Spatio-temporal data mining is concerned with finding specific patterns in databases describing temporally situated spatial objects. Many approaches have been developed in signal processing and computer science to address such a goal, ranging from Fourier Transforms to Independent Component Analysis [Hyvarinen *et al.*, 2001], mixtures of models [Chudova *et al.*, 2003] or string kernel machines [Saunders *et al.*, 2004], to name a few. These approaches aim at particular pattern properties (e.g. independence, generativity) and/or focus on particular data characteristics (e.g. periodicity). However, in some application domains, the above properties are not relevant to extract the patterns of interest. The approach presented in this paper is motivated by such an application domain, functional brain imaging. Non invasive techniques, specifically magnetoencephalography (MEG) [Hämäläinen *et al.*, 1993], provide measures of the human brain activity with an unprecedented temporal resolution (time step 1 millisecond). This resolution enables researchers to investigate the timing of basic neural processes at the level of cell assemblies [Pantazis *et al.*, 2005]. A key task is to find out which brain areas are active and when, i.e. to detect spatio-temporal regions with highly correlated activities. It is emphasized that such regions, referred to as stable spatio-temporal patterns (STPs), are neither periodic nor necessarily independent. Currently, STPs are manually detected, which (besides being a tedious task) severely hampers the reproducibil-

ity of the results. This paper addresses the automatic detection of stable spatio-temporal patterns, i.e. maximal spatio-temporal regions with maximal correlation. This detection problem cannot be formalized as a multi-objective optimization (MOO) problem [Deb, 2001], because experts are interested in several active brain areas: an STP might be relevant though it corresponds to a smaller spatio-temporal region, with a lesser correlated activity than another STP. The proposed approach thus extends MOO along the lines of multi-modal optimization [Li *et al.*, 2002], defining a *multi-objective multi-modal optimization* framework (MoMOO). MoMOO is tackled by an evolutionary algorithm termed *4D-Miner*, using a diversity criterion to relax the multi-objective search (as opposed to diversity enforcing heuristics in MOO, e.g. [Corne *et al.*, 2000; Laumanns *et al.*, 2002]; more on this in section 2.3). To the best of our best knowledge, both the extension of evolutionary algorithms to MoMOO, and the use of multi-objective optimization within spatio-temporal data mining are new, although MOO attracts increasing attention in the domain of machine learning and knowledge discovery (see, e.g., [Ghosh and Nath, 2004; Francisci *et al.*, 2003]). Experimental validation on artificial and real-world datasets demonstrates the good scalability and performances of *4D-Miner*; sought STPs are found within minutes from medium sized datasets (450 Mo), on PC-Pentium 2.4 GHz. The paper is organized as follows. Section 2 formalizes the detection of stable spatio-temporal patterns as a multi-modal multi-objective optimization problem, and motivates the use of evolutionary algorithms [Bäck, 1995; Goldberg, 2002] for their detection. Section 3 gives an overview of *4D-Miner*. Section 4 describes the experiment goals and setting. Section 5 reports on the extensive validation of *4D-Miner* on artificial and real-world data. Section 6 discusses the approach with respect to relevant work, and the paper ends with perspectives for further research.

2 Position of the problem ; notations, criteria

2.1 Notations and definitions

Let N be the number of measure points. To the i -th measure point is attached a spatial position¹ $M_i = (x_i, y_i, z_i)$ and a

¹MEG measure points actually belong to a 2D shape (the surface of the skull). However, the approach equally handles 2D or 3D spatio-temporal data.

temporal activity $C_i(t), t \in [1, T]$. Let $I = [t_1, t_2] \subset [1, T]$

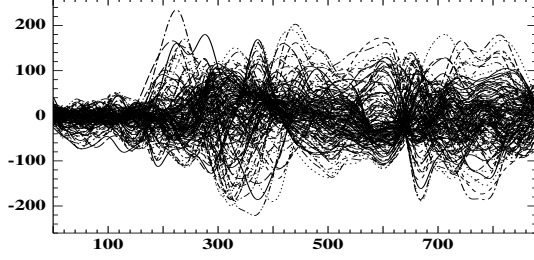


Figure 1: Magneto-Encephalography Data (N = 151, T = 875)

be a time interval, and let $\bar{C}_i^I$ denote the average activity of the i -th measure point during I . The I -alignment $\sigma_I(i, j)$ of measure points i and j over I is defined as:

$$\sigma_I(i, j) = \langle i, j \rangle_I \times \left(1 - \frac{|\bar{C}_i^I - \bar{C}_j^I|}{|\bar{C}_i^I|} \right), \quad \text{where}$$

$$\langle i, j \rangle_I = \frac{\sum_{t=t_1}^{t_2} C_i(t) \cdot C_j(t)}{\sqrt{\sum_{t=t_1}^{t_2} C_i(t)^2 \times \sum_{t=t_1}^{t_2} C_j(t)^2}}$$

As the sought patterns do not need to be spheric, ellipsoidal distances are considered. Only axis-parallel ellipsoids will be considered throughout the paper. For each weight vector $w = (a, b, c)$ ($w \in \mathbb{R}^{+3}$), distance d_w is defined on the measure points as:

$$d_w(i, j) = \sqrt{a(x_i - x_j)^2 + b(y_i - y_j)^2 + c(z_i - z_j)^2}$$

2.2 Multi-objective formalization

A pattern $X = \{i, I, w, r\}$ is characterized by a center point $i, i \in [1, N]$, a time interval I , an ellipsoidal distance d_w , and a radius $r > 0$. The spatial neighborhood $\mathcal{B}(i, w, r)$ is defined as the set of measure points j such that $d_w(i, j)$ is less than r .

The spatial amplitude of X , noted $a(X)$, is the number of measure points in $\mathcal{B}(i, w, r)$. The temporal amplitude of X , noted $\ell(X)$, is the number of time steps in interval I . The spatio-temporal alignment of X , noted $\sigma(X)$ is defined as the average, over all measure points in $\mathcal{B}(i, w, r)$, of their I -alignment with respect to the center i :

$$\sigma(X) = \frac{1}{a(X)} \sum_{j \in \mathcal{B}(i, w, r)} \sigma_I(i, j)$$

Interesting spatio-temporal patterns, according to the expert's first specifications, show maximal spatial and temporal amplitudes together with a maximal spatio-temporal alignment. Specifically, a solution pattern is such that i) increasing its spatial or temporal amplitude would cause the spatio-temporal alignment to decrease; ii) the only way of increasing its spatio-temporal alignment is by decreasing its spatial or temporal amplitude. It thus comes naturally to formalize the STP detection in terms of multi-objective optimization problem (see [Deb, 2001] and references therein; [de la Iglesia et al., 2003; Bhattacharyya, 2000] for related data mining approaches).

The three a, ℓ and σ criteria induce a partial order on the set of patterns, referred to as Pareto domination.

Definition 1. (Pareto domination)

Let $c_1, \dots, c_K$ be K real-valued criteria defined on domain Ω , and let X and Y belong to Ω .

X Pareto-dominates Y , noted $X \succ Y$, iff $c_k(X)$ is greater or equal $c_k(Y)$ for all $k = 1..K$, and the inequality is strict for at least one k_0 .

$$(X \succ Y) \iff \begin{cases} \forall k = 1..K, (c_k(X) \geq c_k(Y)) & \text{and} \\ \exists k_0 \text{ s.t. } (c_{k_0}(X) > c_{k_0}(Y)) \end{cases}$$

The set of non-dominated solutions after a set of criteria is classically referred to as *Pareto front* [Deb, 2001].

2.3 Multi-modal multi-objective formalization

However, the above criteria only partially account for the expert's expectations: a STP might have a lesser spatio-temporal alignment and amplitude than another one, and still be worthy, provided that it corresponds to another active brain area. Therefore, not all sought STPs belong to the Pareto front.

Multi-modal optimization, interested in finding global and local optima of the fitness landscape, has been extensively studied in the EC literature [Li et al., 2002]. Based on the multi-modal framework, a new framework extending multi-objective optimization and referred to as *multi-modal multi-objective optimization* (MoMOO), is thus proposed. Formally, let us first define a relaxed inclusion relationship, noted p -inclusion.

Definition 2. (p-inclusion)

Let A and B be two subsets of set Ω , and let p be a positive real number ($p \in [0, 1]$). A is p -included in B , noted $A \subset_p B$, iff $|A \cap B| > p \times |A|$.

Defining adequately the *support* of a candidate solution X (see below) multi-modal Pareto domination can be defined as follows:

Definition 3. (multi-modal Pareto domination)

With same notations as in Def. 1, X mo-Pareto dominates Y , noted $X \succ_{mo} Y$, iff the support of Y is p -included in that of X , and X Pareto-dominates Y .

$$X \succ_{mo} Y \iff [(Supp(Y) \subset_p Supp(X)) \text{ and } (X \succ Y)]$$

In the case of STPs, the support of $X = (i, I, w, r)$ is naturally defined as $Supp(X) = \mathcal{B}(i, d_w, r) \times I$, with $Supp(X) \subset [1, N] \times [1, T]$.

In contrast with [Corne et al., 2000; Laumanns et al., 2002] who use diversity-based heuristics for a better sampling of the Pareto front, diversity is thus used in MoMOO to redefine and extend the set of solutions.

2.4 Discussion

Functional brain imaging sets two specific requirements on spatio-temporal data mining. First, the expert's requirements are subject to change. Much background knowledge is involved in the manual extraction of stable spatio-temporal patterns, e.g. about the expected localization of the active brain areas. A flexible approach, accommodating further criteria

and allowing the user to customize the search (in particular, tuning the thresholds on the minimal spatial or temporal amplitudes, or spatio-temporal alignment) is thus highly desirable.

Second, the approach must be scalable with respect to the data size (number of measure points and temporal resolution). Although the real data size is currently limited, the computational cost must be controllable in order to efficiently adjust the user-supplied parameters; in other words, the mining algorithm must be an *any-time algorithm* [Zilberstein, 1996].

The approach proposed is therefore based on evolutionary algorithms (EAs); these are widely known as stochastic, population-based optimization algorithms [Bäck, 1995; Goldberg, 2002] that are highly flexible. In particular, EAs address multi-modal optimization [Li *et al.*, 2002] and they can be harnessed to sample the whole Pareto front associated to a set of optimization criteria, with a moderate overhead cost compared to the standard approach (i.e., optimizing a weighted sum of the criteria gives a single point of the Pareto front) [Deb, 2001]. Last, the computational resources needed by EAs can be controlled in a straightforward way through limiting the number of generations and/or the population size.

3 Overview of 4D-Miner

This section describes the 4D-Miner algorithm designed for the detection of stable spatio-temporal patterns.

3.1 Initialization

Following [Daida, 1999], special care is devoted to the initialization of this evolutionary algorithm. The extremities of the Pareto front, where STPs display a high correlation (respectively, a low correlation) on a small spatio-temporal region (resp. a wide region), do not fulfill the expert's expectations. Accordingly, some user-supplied thresholds are set on the minimal spatio-temporal amplitude and alignment.

In order to focus the search on relevant STPs, the initial population is generated using the initialization operator, sampling patterns $X = (i, w, I, r)$ as follows:

- Center i is uniformly drawn in $[1, N]$;
- Weight vector w is set to $(1, 1, 1)$ (initial neighborhoods are based on Euclidean distance);
- Interval $I = [t_1, t_2]$ is such that t_1 is drawn with uniform distribution in $[1, T]$; the length $t_2 - t_1$ of I_j is drawn according to a Gaussian distribution $\mathcal{N}(\min_\ell, \min_\ell/10)$, where $\min_\ell$ is a time length threshold².
- Radius r is deterministically computed from a user-supplied threshold $\min_\sigma$, corresponding to the minimal I -alignment desired.

$$r = \min_k \{d_w(i, k) \text{ s.t. } \sigma_{i,k}^I > \min_\sigma\}$$

All patterns X whose spatial amplitude $a(X)$ is less than a user-supplied threshold $\min_a$ are non-admissible; they will not be selected to generate new offspring. The user-supplied thresholds thus govern the proportion of usable individuals in the initial population.

²In case t_2 is greater than T , another interval I is sampled.

The computational complexity is in $\mathcal{O}(P \times N \times \min_\ell)$, where P is the population size, N is the number of measure points and $\min_\ell$ is the average length of the intervals.

3.2 Variation operators

From parent $X = (i, w, I, r)$, mutation generates an offspring by one among the following operators:

- replacing center i with another measure point in $\mathcal{B}(i, w, r)$;
- mutating w and r using self-adaptive Gaussian mutation [Bäck, 1995];
- incrementing or decrementing the bounds of interval I ;
- generating a brand new individual (using the initialization operator).

The crossover operator, starting from parent X , first selects the mate pattern $Y = (i', w', I', r')$ by tournament selection, Y minimizing the sum of the Euclidean distance between i and i' , and the distance between the center of I and I' among K patterns uniformly selected in the population, where K is set to $P/10$ in all the experiments. The offspring is generated by:

- replacing i with the center i' of the mate pattern Y ;
- replacing w (resp. r) using an arithmetic crossover of w and w' (respectively r and r');
- replacing I by the smallest interval containing I and I' .

An offspring is said admissible iff it satisfies the user-supplied thresholds mentioned in the initialization step.

3.3 Selection scheme

A Pareto archive is constructed with size L (set to $10 \times P$ in the experiments).

A steady-state scheme is used; at each step, an admissible parent X is selected among K uniformly drawn individuals in the population, by retaining the one which is dominated by the fewest individuals in the archive (Pareto tournament [Deb, 2001]). A single offspring Y is generated from X by applying a variation operator among the ones mentioned above.

Offspring Y is evaluated by computing criteria $a(Y)$, $\ell(Y)$, $\sigma(Y)$. Y is rejected if it is mo-Pareto dominated in the population; otherwise, it replaces a non-admissible individual in the population if any; if none it replaces an individual selected after anti-Pareto tournament (the individual out of K randomly selected ones in the population, that is dominated by the most individuals in the archive).

The archive is updated every P generations, replacing the mo-Pareto dominated individuals in the archive with individuals selected from the population after Pareto tournament.

4 Experimental setting and goal

This section presents the goal of the experiments, describes the artificial and real-world datasets used, and finally gives the parameters of the algorithm and the performance measures used to evaluate the algorithm.

4.1 Goals of experiments and datasets

The initial goal is to provide the expert with a usable STP detection algorithm. The real datasets have been collected from people observing a moving ball. Each dataset involves 151 measure points and the number of time steps is 875. As can be noted from Fig. 1, which shows a representative dataset, the amplitude of the activities widely vary along the time dimension. The other goal is to assess the scalability and the performances of *4D-Miner*, which is done using artificial datasets.

The artificial datasets are generated as follows. N measure points are located in uniformly selected locations in the 3D domain $[0, 1]^3$. Activities are initialized from random cumulative uniform noise, with $C_i(t+1) = C_i(t) + \epsilon$, and ϵ is drawn according to $U(0, \tau)$. Every target STP $S = (i, w, I, r, \delta)$ is defined from a center i , a weight vector w , a time interval I , a radius r , and a fading factor δ , used to bias the activities as detailed below. The activity C_S of the STP S is the average activity in the spatio-temporal region $\mathcal{B}(i, w, r) \times I$. Thereafter, activities are biased according to the target STPs: for each measure point j , for each STP S such that j is influenced by S ($d_w(i, j) < r$), the activity $C_j(t)$ is smoothed as

$$C_j(t) = (1 - e^{-\alpha_i(j,t)})C_j(t) + e^{-\alpha_i(j,t)}C_S$$

where $\alpha_i(j, t) = d_w(i, j) + \delta \times d(t, I)$

and $d(t, I)$ is the distance of t to interval I (0 if t belongs to I , otherwise the minimum distance of t to the bounds of I). The scalability of *4D-Miner* is assessed from its empirical

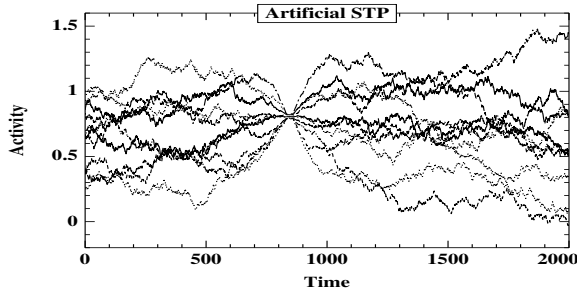


Figure 2: An artificial STP ($T = 2000$, $N = 2000$)

complexity depending on the number N of measure points and the number T of time steps.

The performances of *4D-Miner* are measured using the standard criteria in information retrieval, recall and precision. The recall is the fraction of target STPs that are identified by *4D-Miner*, i.e. p -included in an individual in the archive; the precision is the fraction of relevant individuals in the archive (p -included in a target STP). For each experimental setting (see below), the recall and precision are averaged over 21 independent runs. The number of target STPs is set to 10 in all experiments. Each STP influences a number of measure points varying in $[10, 20]$, during intervals uniformly selected in $[1, T]$ with length varying in $[15, 25]$, and δ uniformly selected in $[0, .05]$. Each problem instance thus includes STPs with various levels of difficulty; the detection is hindered as the spatio-temporal support (r and I) is comparatively low and the δ parameter increases (the regularity is not visible outside interval I , as in Fig 2).

4.2 Experimental setting

The experiments reported in the next section considers a population size $P = 200$, evolved along 8000 generations (8,000 fitness evaluations per run). A few preliminary runs were used to adjust the operator rates; the mutation and crossover rates are respectively set to .7 and .3.

The number N of measure points ranges in $\{500, 1000, 2000, 4000\}$. The number T of time steps ranges in $\{1000, 2000, 4000, 8000\}$.

The thresholds used in the initialization are: $\min_a = 5$ (minimum number of curves supporting a pattern); $\min_\ell = 5$ (minimum temporal amplitude of a pattern); $\min_\sigma = .1$ (minimum spatio-temporal alignment of two curves in a pattern).

For computational efficiency, the p -inclusion is computed as: X is p -included in Y if the center i of X belongs to the spatial support of Y , and there is an overlap between their time intervals.

The maximal size of the datasets ($T = 8000$, $N = 4000$) is 456 Mo. Computational runtimes are given on PC-Pentium IV, 2.4 GHz. *4D-Miner* is written in C++.

5 Experimental validation

This section reports on the experiments done using *4D-Miner*.

5.1 Experiments on real datasets

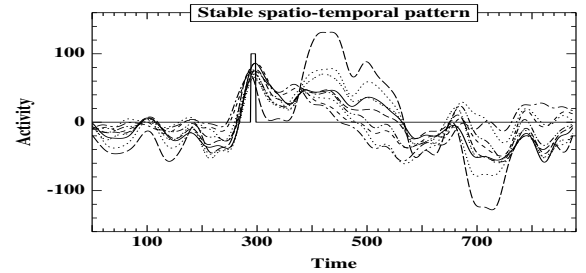


Figure 3: A stable spatio-temporal pattern, involving 8 measure points within interval $[289, 297]$, alignment .2930

Typical STPs found in the real datasets are shown in Figs. 3 and 4, showing all curves belonging to the STP plus the time-window of the pattern. The runtime is less than 1 minute (PC Pentium 1.4 GHz). As discussed in section 2.3, many relevant STPs are Pareto-dominated: typically the STP shown in Fig 3 is dominated by the one in Fig 4.

These patterns are considered satisfactory by the expert. All experiments confirm the importance of the user-defined thresholds, controlling the quality of the initial population. Due to the variability of the data, these threshold parameters are adjusted for each new experiment.

The coarse tuning of the parameters can be achieved based on the desired proportion of admissible individuals in the initial population. However, the fine-tuning of the parameters could not be automatized up to now, and it still requires running *4D-Miner* a few times. For this reason, the control of the computational cost through the population size and number of generations is one of the key features of the algorithm.

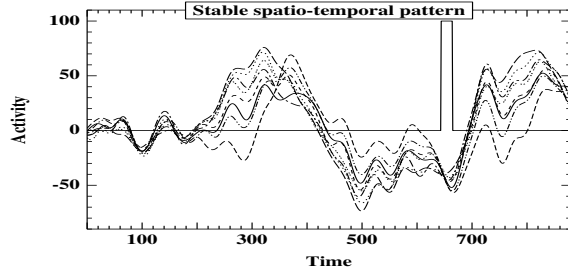


Figure 4: Another stable spatio-temporal pattern involving 9 measure points within interval [644,664], alignment .3963

5.2 Multi-objective multi-modal optimization

The good scalability of *4D-Miner* is illustrated in Fig. 5. The empirical complexity of the approach is insensitive to the number of time steps T and linear in the number N of measure points. This computational robustness confirms the analysis (section 3.1), the evaluation of each pattern has linear complexity in N . On-going work is concerned with exploiting additional data structures inspired from [Wu *et al.*, 2004] in order to decrease the complexity in N . Table 1 reports the

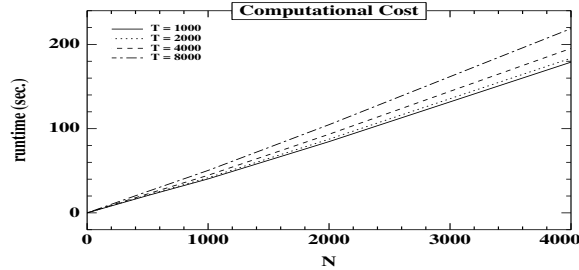


Figure 5: Computational cost vs N , for $T = 1000, 2000, 4000, 8000$. (in seconds, on PC 2.4 GHz)

recall achieved by *4D-Miner* over the range of experiments, averaged over 21 independent runs, together with the standard deviation. On-line performances are illustrated on Fig

N	T			
	1,000	2,000	4,000	8,000
500	98 ± 5	93 ± 9	92 ± 7	79 ± 16
1000	96 ± 6	96 ± 6	82 ± 14	67 ± 12
2000	96 ± 5	87 ± 12	72 ± 14	49 ± 15
4000	89 ± 10	81 ± 13	56 ± 14	32 ± 16

Table 1: Recall achieved by *4D-Miner* vs N and T (average percentage and standard deviation over 21 runs)

6. These results confirm the robustness of the proposed approach: a graceful degradation of the recall is observed as T and N increase. It must be noted that STPs occupy a negligible fraction of the spatio-temporal domain (circa 10^{-4} for $T = 8000$, $N = 4000$). The average precision is low, ranging from 12 to 20% (results omitted due to space limitations). However, post-pruning can be used to sort the wheat from the chaff in the final archive, and increase the precision up to

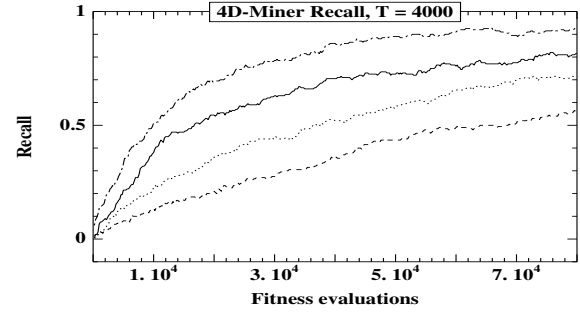


Figure 6: Recall achieved by *4D-Miner* vs the number of fitness evaluations. ($T = 4000$, $N = 500, 1000, 2000, 4000$, average over 21 runs).

50% without decreasing the recall; the post-pruning straightforwardly removes the STPs with small spatial or temporal amplitudes.

Quite different effects are obtained when the archive is pruned along the search (e.g. by increasing the thresholds on the minimal spatial and temporal amplitudes), which decreases the overall performances by an order of magnitude; interestingly, similar effects are observed in constrained evolutionary optimization when the fraction of admissible solutions is very low.

A final remark is that the performances heavily depend upon the user-supplied thresholds (section 3.1), controlling the diversity and effective size of the population. Indeed, a parametric model of the dataset would enable automatic adjustment of these parameters. It might also be viewed as an advantage of *4D-Miner* that it does not require any prior model of the data, and that inexpensive preliminary runs can be used to adjust the needed parameters.

6 Relevant work

This brief review does not claim exhaustiveness; the interested reader is referred to [Shekhar *et al.*, 2003; Roddick and Spiliopoulou, 2002] for comprehensive surveys. Spatio-temporal data mining applications (e.g., remote sensing, environmental studies, medical imaging) are known to be computationally heavy; as most standard statistical methods do not scale up properly, new techniques have been developed, including randomized variants of statistical algorithms. Many developments are targeted at efficient access primitives and/or complex data structures (see, e.g., [Wu *et al.*, 2004]); another line of research is based on visual and interactive data mining (see, e.g., [Keim *et al.*, 2004]), exploiting the unrivaled capacities of human eyes for spotting regularities in 2D-data. Spatio-temporal data-mining mostly focuses on clustering, outlier detection, denoising, and trend analysis. For instance, [Chudova *et al.*, 2003] used EM algorithms for non-parametric characterization of functional data (e.g. cyclone trajectories), with special care regarding the invariance of the models with respect to temporal translations. The main limitation of such non-parametric models, including Markov Random Fields, is their computational complexity, sidestepped by using randomized search for model estimates.

7 Discussion and Perspectives

This paper has proposed a stochastic approach for mining stable spatio-temporal patterns. Indeed, a very simple alternative would be to discretize the spatio-temporal domain and compute the correlation of the signals in each cell of the discretization grid. However, it is believed that the proposed approach presents several advantages compared to the brute force, discretization-based, alternative. First of all, *4D-Miner* is a fast and frugal algorithm; its good performances and scalability have been successfully demonstrated on medium sized artificial datasets. Second, data mining applications specifically involve two key steps, exemplified in this paper: i) understanding the expert's goals and requirements; ii) tuning the parameters involved in the specifications. With regard to both steps, the ability of working under bounded resources is a very significant advantage; any-time algorithms allow the user to check whether the process can deliver useful results, at a low cost. Further research is concerned with extending *4D-Miner* in a supervised learning perspective (finding the STPs that are complete – active in several persons undergoing the same experiment – and correct – not active in the control experiment). The challenge is to directly handle the additional constraints of completeness and correction in the multi-objective multi-modal optimization framework presented here.

Acknowledgments

The authors are partially supported by the PASCAL Network of Excellence (IST-2002-506778); this study was supported by a grant from the French Ministry of Research & Technology "ACI, New Applications of Mathematics".

References

- [Bäck, 1995] T. Bäck. *Evolutionary Algorithms in theory and practice*. New-York:Oxford University Press, 1995.
- [Bhattacharyya, 2000] S. Bhattacharyya, Evolutionary algorithms in data mining: multi-objective performance modeling for direct marketing. In *KDD'00*, pp 465-473. ACM Press, 2000.
- [Chudova et al., 2003] D. Chudova, S. Gaffney, E. Mjølness, and P. Smyth. Translation-invariant mixture models for curve clustering. In *KDD'03*, pages 79–88. ACM Press, 2003.
- [Corne et al., 2000] D. Corne, J. D. Knowles, and M. J. Oates. The Pareto envelope-based selection algorithm for multi-objective optimisation. In *PPSN VI*, pages 839–848. Springer Verlag, 2000.
- [Daida, 1999] J.M. Daida. Challenges with verification, repeatability, and meaningful comparison in Genetic Programming. In *GECCO'99*, pages 1069–1076. Morgan Kaufmann, 1999.
- [Deb, 2001] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley, 2001.
- [Francisci et al., 2003] D. Francisci, M. Collard, Multi-Criteria Evaluation of Interesting Dependencies according to a Data Mining Approach, In *CEC'03*, Vol. 3, pp. 1568-1574, IEEE Press, 2003.
- [Ghosh and Nath, 2004] A. Ghosh and B. Nath. Multi-objective rule mining using genetic algorithms. *Inf. Sci.*, 163(1-3):123–133, 2004.
- [Goldberg, 2002] D. Goldberg. *The Design of Innovation: Lessons from and for Genetic and Evolutionary Algorithms*. MIT Press, 2002.
- [Hyvarinen et al., 2001] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley New York, 2001.
- [Hämäläinen et al., 1993] M. Hämäläinen, R. Hari, R. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalography: theory, instrumentation, and applications to noninvasive studies of the working human brain. *Rev. Mod. Phys.*, 65:413–497, 1993.
- [de la Iglesia et al., 2003] B. de la Iglesia, M.S. Philpott, A.J. Bagnall, V.J. Rayward-Smith, Data Mining Using Multi-Objective Evolutionary Algorithms, In *CEC'03*, Vol. 3, pp. 1552-1559, IEEE Press, 2003.
- [Keim et al., 2004] D. A. Keim, J. Schneidewind, and M. Sips. Circleview: a new approach for visualizing time-related multidimensional data sets. In *Proc. of Advanced Visual Interfaces*, pages 179–182. ACM Press, 2004.
- [Laumanns et al., 2002] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.
- [Li et al., 2002] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation*, 10(3):207–234, 2002.
- [Pantazis et al., 2005] D. Pantazis, T. E. Nichols, S. Baillet, and R.M. Leahy. A comparison of random field theory and permutation methods for the statistical analysis of MEG data. *Neuroimage*, to appear in 2005.
- [Roddick and Spiliopoulou, 2002] J.F. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Trans. on Knowledge and Data Engineering*, 14(4):750–767, 2002.
- [Saunders et al., 2004] C. Saunders, D. R. Hardoon, J. Shawe-Taylor, and G. Widmer. Using string kernels to identify famous performers from their playing style. In *ECML04*, pages 384–395. Springer Verlag, 2004.
- [Shekhar et al., 2003] S. Shekhar, P. Zhang, Y. Huang, and R. R. Vatsavai. Spatial data mining. In H. Kargupta and A. Joshi, eds, *Data Mining: Next Generation Challenges and Future Directions*. AAAI/MIT Press, 2003.
- [Wu et al., 2004] K.L. Wu, S.K. Chen, and P.S. Yu. Interval query indexing for efficient stream processing. In *13th ACM Conf. on Information and Knowledge Management*, pages 88–97, 2004.
- [Zilberstein, 1996] S. Zilberstein. Resource-bounded reasoning in intelligent systems. *Computing Surveys*, 28(4), 1996.

Temporal-Difference Networks with History

Brian Tanner and Richard S. Sutton

University of Alberta

Reinforcement Learning and Artificial Intelligence Laboratory

Edmonton, Alberta, Canada T6G 2E8

{btanner,sutton}@cs.ualberta.ca

Abstract

Temporal-difference (TD) networks are a formalism for expressing and learning grounded world knowledge in a predictive form [Sutton and Tanner, 2005]. However, not all partially observable Markov decision processes can be efficiently learned with TD networks. In this paper, we extend TD networks by allowing the network-update process (answer network) to depend on the recent history of previous actions and observations rather than only on the most recent action and observation. We show that this extension enables the solution of a larger class of problems than can be solved by the original TD networks or by history-based methods alone. In addition, we apply TD networks to a problem that, while still simple, is significantly larger than has previously been considered. We show that history-extended TD networks can learn much of the common-sense knowledge of an egocentric gridworld domain with a single bit of perception.

Temporal-difference (TD) networks are a formalism for expressing and learning grounded knowledge about dynamical systems [Sutton and Tanner, 2005]. TD networks represent the state of the dynamical system as a vector of predictions about future action-observation sequences. Each prediction is an estimate of the probability or expected value of some future event. For example, a prediction might estimate the probability of seeing a particular observation at the next time step. The predictions generated at each time step are thought of as “answers” to a set of “questions” asked by the TD network.

Representations that encode the state of a dynamical system as a vector of predictions are known as predictive representations [Littman *et al.*, 2002; Jaeger, 1998; Rosencrantz *et al.*, 2004]. Predictive representations are a relatively new area of research; as a community we are answering fundamental questions about their possibilities and limitations. So far, the results have been encouraging. Singh *et al.* have shown that one particular representation known as linear predictive state representations (or linear PSRs) can

represent any n^{th} -order Markov model or partially observable Markov decision process (POMDP) [Singh *et al.*, 2004; Littman *et al.*, 2002]. Further, they showed that the size of the PSR scales at least as well as the existing approaches; a linear PSR model is at least as compact as the equivalent POMDP or n^{th} -order Markov model. TD networks are a generalization of linear PSRs and therefore inherit their representational power [Singh, 2004].

TD networks have been applied successfully to simple environments, both fully and partially observable. Although TD networks have the expressive power to accurately model complex partially-observable environments, we show that the existing learning algorithm is insufficient for learning some of these models.

In this paper, we explore the classes of environment whose model could not previously be learned and we improve TD networks so that they can learn models of these environments.

In Section 1 we review the TD networks specification. In Sections 2 and 3 we examine the class of problems that we have been unable to learn with the existing specification of TD networks, augment the specification, and present new results. We present the results of applying these augmented TD networks to a more complex grid-world domain in Section 5. Finally we conclude and discuss the direction of our future research in Section 6.

1 TD Networks without History

In the following text we describe a specific instantiation of the original TD networks specification that is instructive for understanding the details of our work. For information on the general specification of TD networks we direct the reader to the original work [Sutton and Tanner, 2005].

The problem addressed by TD networks is a general one of learning to predict aspects of the interaction between a decision making agent and its environment (a dynamical system). At each of a series of discrete time steps t , and agent takes an action $a_t \in \mathcal{A}$ and the environment responds by generating an observation $o_t \in \mathcal{O}$. In this work, we will consider TD networks with two observations, 0 and 1. The action and observation events occur in sequence, $a_{t-1}, o_t, a_t, o_{t+1}, a_{t+1}, o_{t+2}$. This sequence will be called *experience*. We are interested in predicting not just each next observation, but more general, action-conditional functions of future experience.

The focus of this work current is on partially observable environments—environments where the observation o_t is not a sufficient statistic to make optimal predictions about future experience (o_t does not uniquely identify the *state* of the environment). We will be using TD networks to learn a model of the environment that is accurate and can be maintained over time.

A TD network is a network of nodes, each representing a single scalar prediction. The nodes are interconnected by links representing target relationships between predictions, observations, and actions. These nodes and links determine a set of questions being asked about the data and predictions, and accordingly are called the *question network*.

Each node on the TD network is a function approximator that outputs a prediction using inputs such as the current observation, the previous action, and the predictions made at the previous time step. This computation part of the TD network is thought of as providing the answers to the questions, and accordingly is called the *answer network*.

Figure 1 shows a typical question network. The question of node y^0 at time t is ‘If the next action is a_1 , what is the probability that the next observation o_{t+1} will be 1?’. Similarly, node y^2 asks ‘If the next action is a_1 , what will node y^0 predict at time $t + 1$?’. This is a desired relationship between predictions, but also a question about the data. We can unroll this interpredictive (or TD) relationship to look at the extensive relationship between y^2 and the data, which yields the question ‘If the next two actions are a_1 , what is the probability that o_{t+2} will be 1?’.

In a fully observable (Markov) environment, it is natural for the question network to be a set of questions that are in some way interesting to the experimenter. In partially-observable environments the structure of the question network has additional constraints; the answers should be a sufficient statistic that accurately represents the state and can be updated as new data becomes available. Although the question of how to discover a question network that expresses a minimal sufficient statistic is important, it is tangential to the focus of this work. The question networks we use are not minimal, and a network like that in Figure 1 likely contains

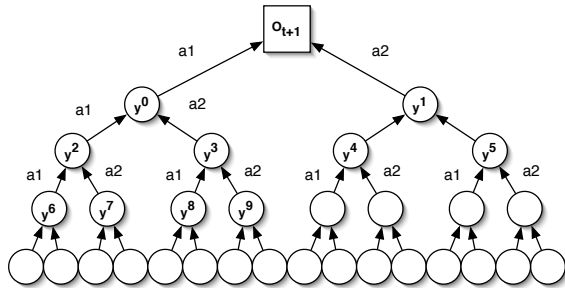


Figure 1: Symmetric action-conditional question network. The network forms a symmetric tree, with a branching factor of $|A|$. This example has depth $d = 4$. Some of the labels have been left out of this diagram for clarity, each of these nodes should have a label y^i and each is conditioned on some action.

extraneous questions.

Formally, $y_t^i \in [0, 1]$, $i = 0, \dots, n - 1$ denotes the prediction for node i at time step t . The column vector of predictions $\mathbf{y}_t = (y_t^0, \dots, y_t^{n-1})^T$ is updated according to a vector-valued prediction function with modifiable parameter $\mathbf{W}$:

$$\mathbf{y}_t = \sigma(\mathbf{W}_t \mathbf{x}_t) \quad (1)$$

This prediction function corresponds to the answer network, where $\mathbf{x}_t \in \mathbb{R}^m$ is a feature vector, $\mathbf{W}_t$ is a $n \times m$ matrix of weights, and σ is the S-shaped logistic function $\sigma(s) = \frac{1}{1+e^{-s}}$. The feature vector is a function of the preceding action, observation, and node values.

In previous work on TD networks, $\mathbf{x}_t$ had a binary component for each unique combination of the current observation and previous action (one of which is 1, the rest 0), and n more for the previous node values $\mathbf{y}_{t-1}$. Additionally, $\mathbf{x}_t$ has a bias term (constant value of 1). In our experiments we also included real valued features with the complement of the predictions from the last time step ($1 - \mathbf{y}_{t-1}$), although we later found this was not necessary. The learning algorithm for each component w_t^{ij} of $\mathbf{W}_t$ is of the form

$$\Delta w_t^{ij} = \alpha(z_t^i - y_t^i)(y_t^i)(1 - y_t^i)x_t^j c_t^i, \quad (2)$$

where α is a step-size parameter, z_t^i is a target for y_t^i defined by a question network, and $c_t^i \in \{0, 1\}$ corresponds to whether the action condition for y_t^i was met at time t .

2 Cycle-World Counterexamples

In our experimentation we found that TD networks are able to solve certain, but not all of our small testing problems. The success of learning does not seem to be directly correlated with the *complexity* of the environment. In previous work, TD networks were able to learn an accurate model of a partially-observable seven-state deterministic random walk as well as the probabilities in a fully observable stochastic random walk [Sutton and Tanner, 2005]. Since that time, we have discovered that TD networks are unable to learn an accurate model of certain small partially-observable deterministic environments. Careful analysis has determined that the question networks that were used were sufficient to represent the appropriate model, so the issue must lie somewhere in the TD network specification.

Figure 2 presents a simple example of a task and question network for which the solution is representable but not learnable by TD networks without history. The cycle world consists of the four states shown on the left. The current state of the system cycles clockwise through the states. There is a single observation bit that is 1 at the top of the cycle, and 0 at all other times. On the right is a question network which asks what the observation bit will be one, two, and three time steps in the future. Recall that at time t , $\mathbf{y}_t$ is calculated as a function of $(\mathbf{y}_{t-1}, a_{t-1}, o_t, \mathbf{W}_t)$. If we assume that the $\mathbf{y}_{t-1}$ is correct, there is a solution for the weights that will keep $\mathbf{y}_t$ correct at each successive time step. Unfortunately, $\mathbf{y}_{t-1}$ will never be correct; the solution exists but will not be found.

To illustrate this, we must look at the question network, and remember that it specifies target values for the answer

network. At the start of training, y_{t-1} will likely be incorrect. There are no actions in this environment, so the current observation o_t is the only useful input feature in $\mathbf{x}_t$. For the network to become correct, it is necessary that some sequence of questions can eventually be answered, starting or *anchored* only with knowledge of o_t . Also note that when training begins, the only node with a valid target is y^0 , because its target is not a prediction, but rather the grounded observable value of o_{t+1} . As training progresses, the agent interacts with the environment and some answers will be learned, anchored only on the grounded observations. Eventually, the environment will reach a point where $o_t = 0$ and y_t^0 should be 1. The information that distinguishes this case from the case where $o_t = 0$ and y_t^0 should be 0 lies in a *correct* answer for y_{t-1}^1 . Unfortunately, the target for y_{t-1}^1 is y_t^0 . In this case, the cyclic dependency between the question network and the temporal flow of information eliminates the possibility of the TD network learning a correct solution.

After considering this problem, it became apparent that it is necessary to find an augmentation to the TD network specification to eliminate cyclic information dependencies. Such a dependency occurs when the target z^i for node i relies on y^i making accurate predictions. This dependency can be eliminated by providing additional input features to the TD network. This additional information acts as an anchor for the TD network predictions.

Incidentally, good *approximate* solutions to the cycle world can be learned by TD networks consisting of a single node. Clearly, there is no way that a single predictive node can solve this problem perfectly, but it can achieve very low error in an unusual way. When $o_t = 1$, the network predicts approximately .001. On the next time step, the network will multiply the previous prediction so as to predict approximately .01. On the next step it will predict .1, and then finally .99 for the step where o_t will again be 1. Unfortunately this solution is not stable, and continued training leads to oscillation between this sort of approximate solution and strictly predicting 0 at each step.

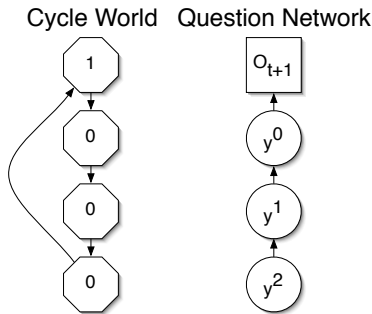


Figure 2: A counterexample for TD-network learning without history. On the left is a representation of the cycle world. This environment has four states that are cycled through deterministically. On the right is the associated question network. There are no actions in this world.

Features	Initial	Final
1	1	1
history = 000	0	0
history = 001	0	0
history = 010	0	0
history = 011	0	0
history = 100	1	1
history = 101	0	0
history = 110	0	0
history = 111	0	0
y_{t-1}^0	.5	0
y_{t-1}^1	.5	0
y_{t-1}^2	.5	1

Figure 3: Input vector for cycle world with 3-step history and 3 levels of predictive nodes. On the left is the definition of each feature. The first feature is the bias term. The next 8 features correspond to the 8 distinct 3-step histories $\{o_{t-2}o_{t-1}o_t\}$ (not all are possible in this world). The final 3 features are the predictions from the previous time step. The middle vector is a sample input vector for the third state from the top of the cycle world at the start of learning. At this point, all of the predictions are at their initial value, .5. Finally, the rightmost vector is the input vector for the third state when learning is complete, all of the predictions are accurate.

3 TD Networks with History

The cycle world is a problem in which there is a simple relationship between the observations and recent experience. Methods that try to directly learn such relationships are called history-based methods. We will consider history-based methods which predict o_{t+1} using a different variable for each unique k-length window of history where a k-length window of history is defined as $a_{t-k}o_{t-k+1} \dots a_{t-1}o_t$. In this case, a window of length 3 would be sufficient to uniquely identify each state of the system and thus would be able to make accurate predictions. Incorporating short history into the feature vector $\mathbf{x}_t$ of a TD network should allow the TD network to learn a correct solution to this problem. Figure 3 shows an example of a hybrid input vector that uses 3 time steps of history and 3 predictive nodes.

In order to test the hypothesis that incorporating history into the input vector $\mathbf{x}_t$ of a TD network would allow it to solve a greater class of problem, we used a cycle world like that in Figure 2, except with six states instead of four. This size was chosen to clearly illustrate the effectiveness of different configurations of history and predictive nodes. On this problem, we tested three different methods: (1) TD networks as previously specified without history, (2) a simple history-based approach, (3) a combination of TD networks and history together. For each method we used several values for the step size parameter; the best of these was used as the performance measure for that method. For each method and step size, the network was trained for at least one million time steps. The 1-step prediction errors were averaged over the final 20000 steps to produce an overall performance measure for each method.

The results, shown in Figure 4, show that the simple

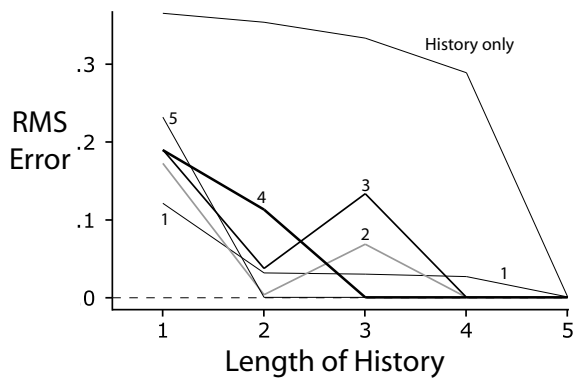


Figure 4: Performance on the 6-state cycle world of TD networks extend to incorporate various lengths of history. The different lines correspond to different depths of the question network, as indicated by the numeric label.

history-based method only performed well when it had enough history to solve the problem exactly. TD networks without history correspond to the data points with history length one. These performances are better than history of length one, but not as good as the TD networks with history. It is also interesting to notice that the TD network is able to solve the problem with a much shorter window than the history-based method alone. This illustrates that our combined algorithm is not simply using history instead of the predictive representation, but rather is leveraging the history to learn a predictive representation. It is interesting that the performance of the various combinations of history and predictive nodes do not follow a clear pattern. For example, when there are 2 predictive nodes, it appears that 2 or 4 steps of history is better than having 3 steps. We have verified that the minimum length of history required to solve the 6-state cycle world exactly with 2 predictive nodes is a 4-step history. This means that the low error seen with 2 steps of history is a case of the TD network stumbling on a good approximate solution when it could not represent an exact solution (as discussed at the end of Section 2).

4 Indefinite-memory Problems

In the previous section, we introduced history to the TD network specification in order to eliminate cyclic dependencies and increase the class of problems where solutions can be learned with TD networks. There appeared to be a tradeoff between predictive levels in the question network and lengths of history. From this example alone, it may not be clear that the combined approach is superior to a history-only approach. There is a potentially large class of problems that cannot be represented with a history-only approach, but can be represented and solved by TD networks. Environments in this class are such that there is no finite length of history that

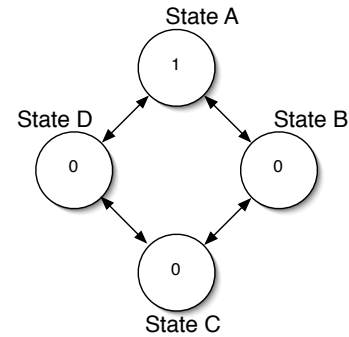


Figure 5: An indefinite-memory problem, the four-state deterministic ring world. There are two actions in this world, *next* and *previous*. *Next* advances in clockwise rotation while *previous* advances in counter-clockwise rotation. Prediction methods using a finite length history will lose localization after some number of transitions back and forth between states B and C or C and D.

can uniquely identify the current state of the environment. We will refer to problems in this class as indefinite-memory problems.

One simple example of an indefinite-memory problem is the *ring world* shown in Figure 5. Because states B, C, and D are indistinguishable, there are sequences of actions that keep the environment in that subset of states and will eventually fill a fixed-length memory with useless information. In contrast, a TD network can easily represent this environment, and can never be made to forget its location in the environment.

We applied TD networks with various depths of question network and lengths of history to the 5-state ring world problem. The performance measure used was the same as in the previous experiments, except in this case averaged over 25 independent runs of 10 million time steps. The results are shown in Figure 6. As the history window increases, the history-only method more closely approximates the correct solution. This improvement seems to diminish as the history window gets larger, and is further hampered by the fact that the number of unique histories grows exponentially with the length of the window. With the predictive approach, the problem is solved correctly with only 1 level of history and a predictive question network of depth 3.

It is interesting that, provided enough time, the TD network can learn a correct model of this environment without history, something which it could not do for the cycle world. Especially puzzling was that these two problems seemed highly related, the cycle world seemingly even less complex than the ring world. The fact that actions have inverses in the ring world eliminates the information flow dependencies that existed in the cycle world. In the ring world, the agent can incrementally learn more and more about the environment. In early training, the agent can anchor itself when $o_t = 1$ because this observation uniquely identifies this state. As time passes, the agent can learn accurate 1-step predictions from that anchored location. It can also learn 2-step predictions that involve leaving this position and then returning imme-

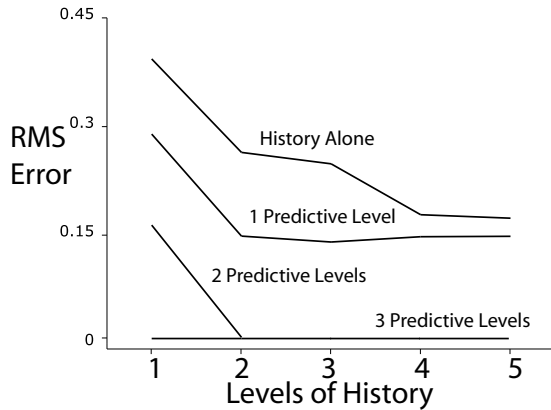


Figure 6: Performance on the 5-state ring world as a function of length of history and depth of question network. The history method suffers from diminishing returns as size of the history window increases. Learning also slows considerably because the number of unique histories that can be observed grows exponentially.

diately. This process can continue until this chaining effect has allowed the agent to make accurate predictions from all positions in the network.

5 Gridworld Experiments

In previous work, and so far in this paper, TD networks have been applied only to abstract problems with fewer than 10 states. In this final section of the paper we present suggestive results for a gridworld environment that is an order of magnitude more complex than those previously considered. The gridworld in Figure 7 shows the environment that we chose. In this environment, the agent has a single perceptual input, a single bit indicating whether there is a wall directly in front of it. The agent also has a very limited action space, it can either attempt to move forward or it can rotate 90 degrees clock-wise. We encourage the reader to consider two analogous tasks suitable for a human. First, consider sitting at a table with two buttons and a light bulb. You are told that the light bulb will turn on and off based on the buttons pressed according to some unobservable process. If that unobservable process were the map in Figure 7, could a human learn this problem?

A second interpretation of the problem which should be more familiar is navigating around a room with the lights off. At all times you can feel if you are touching a wall, otherwise nothing. What sorts of common-sense knowledge would you apply to this type of scenario? The simplest knowledge is that if you are touching a wall and you attempt to walk forward, you will still be touching a wall. You would also know that 4 consecutive 90 degree turns ends with the same observation that it started with. These are the types of knowledge that can be learned easily with a history-based approach.

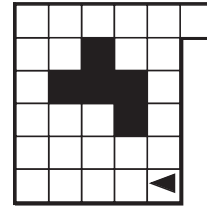


Figure 7: The gridworld used in the final experiment. The agent's location is represented by the triangle. The agent has a very limited perceptual space, it observes 1 if the arrow is pointing at a block, and 0 if the arrow is pointing at an open space. This agent has only two actions, forward and turn. Forward will move the agent 1 space in the direction that the arrow points (if it is not blocked), while turn will rotate the arrow clock-wise 90 degrees. There are 104 unique environmental states in this world.

There are other types of common-sense knowledge that are harder to learn with history. If you are facing wall, then you turn around (180 degrees), observe no wall and walk forward; what common-sense predictions could we make? First, we would know that turning 180 degrees again will be clear, and walking forward from there will take us back to the wall. We also know that if we rotate 360 degrees any number of times at any step of this process, the entire process remains intact, nothing changes. This concept is impossible to learn with a fixed length history. This scenario exemplifies the conceptual and practical difference between predictive representations and history-based representations. After facing a wall, turning 180 degrees and going forward, a 3-level history-based approach knows 'I am in the state described as {wall, turn, clear, turn, clear, forward, clear}'. The predictive agent has a different representation, more like 'I am in the state where $\{Pr(clear|turn, turn) = 1, Pr(wall|turn, turn, forward) = 1, etc.\}$ '.

We ran a TD network of depth 5 with a history of length 6 in this environment for more than 50 million time steps to give it the opportunity to learn about the causal structure of its environment. At the end of this time we took control of its decisions to explore particular places in the world as shown in Figure 8. On the left, this figure shows the state of the environment at various points in time as the agent is controlled from state to state. On the right a representation of the agents predictions at each step is shown. These predictions show that the agent has learned a great deal about its environment.

At $t = 0$ the agent is in a corner, and we can see that it knows there is a wall on its left side and a wall behind it. Notice just to the right of the 'A' is a dark bar. This bar represents the prediction for turning right, going forward twice, and then turning right three more times. The fact that this bar is dark indicates that the agent correctly knows something about the nearest interior blocked cell. Above the 'A', the agent also predicts a wall and below the 'A' it predicts gray. This shows that the agent does not completely understand that

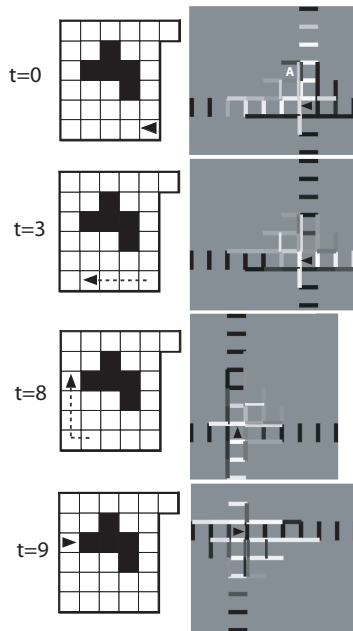


Figure 8: A sample trajectory in the gridworld and a representation of the associated predictions at several time steps. The representation of the agent's predictions is laid out as a subjective action-conditional map of the area near the agent. All of the colored bars correspond to some prediction that the agent is making. For example, if the agent believes that it will observe a wall by going forward, the bar in front of the triangle of it is black. If the agent believes it will be clear, the bar is white, and if the agent is uncertain, the bar is gray.

it will be blocked if it chooses forward, turn forward, but it surely knows it will be blocked if it went forward again at the end of that sequence.

The white bars in front of the agent indicate that it believes it will see clear if it goes forward for the next four steps, and then it will see a wall.

At $t = 3$, the agent has moved along the wall and we can see that its expectation of seeing a wall ahead of it has moved closer, while to its left are all the predictions appropriate to their being an extended wall.

At $t = 8$, the agent has moved along another wall. Note that it knows how many steps it is from the wall ahead.

At $t = 9$ the agent has turned to the right and observed that its way is blocked. This agent does not know all of the details of its world, at $t = 9$ we can see that the agents predictions do not reflect many of the details of the environment.

By inspection, we can determine that much knowledge of this gridworld environment has been obtained by the agent. For example, the agent knows that if it is facing a wall and goes forward, it will always see a wall. Further, when facing a wall, the agent correctly does not change its other predictions if told to go forward, it seems to know that the state is not changing. The agent appears to know that four consecutive turns should leave the set of predictions unchanged. The agent also seems to have a sense of rotational persistence, if

it sees a wall, and then walks away and does a few rotations, it remembers where the wall is if navigated back to it.

TD networks can learn what we would consider to be much of the common-sense knowledge in a complex, perceptually deprived grid-world. In this case the agent learned concepts related to rotation and persistence that were grounded in primitive actions and observations.

6 Conclusions and Future Work

We have presented a straightforward extension of TD networks to incorporate the strengths of history-based methods. The combination of history-based learning and TD network learning is more than putting two algorithms into one box and using the appropriate approach for a particular problem; the combined algorithm is stronger than either of its parts on their own.

There are still many questions about TD networks that deserve further attention. One question is whether there are other classes of problems that cannot be solved with the augmented TD network specification. Are there some types of common-sense knowledge that cannot be learned with a TD network? Can we learn optimal structure for the question network instead of specifying it manually? We expect to explore these and other questions questions in future work.

Acknowledgments

The authors gratefully acknowledge the ideas and encouragement they have received in this work from Satinder Singh, Doina Precup, Michael Littman, Mark Ring, Eddie Rafols, Vadim Bulitko, Anna Koop, and all the members of the rlai.net group. This work was supported in part by NSERC and iCORE.

References

- [Jaeger, 1998] Herbert Jaeger. Discrete-time, discrete-valued observable operator models: a tutorial. Technical report, German National Research Center for Information Technology, 1998.
- [Littman *et al.*, 2002] M.L. Littman, R. S. Sutton, and S. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.
- [Rosencrantz *et al.*, 2004] Matthew Rosencrantz, Geoff Gordon, and Sebastian Thrun. Learning low dimensional predictive representations. In *ICML '04: Twenty-first international conference on Machine learning*. ACM Press, 2004.
- [Singh *et al.*, 2004] Satinder Singh, Michael R. James, and Matthew R. Rudary. Predictive state representations: A new theory for modeling dynamical systems. In *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference*, pages 512–519, 2004.
- [Singh, 2004] Satinder Singh. Private communication, 2004.
- [Sutton and Tanner, 2005] Richard S. Sutton and Brian Tanner. Temporal-difference networks. In *Advances in Neural Information Processing Systems 17*, pages 1377–1384, Cambridge, MA, 2005. MIT Press.

Learning to Play Like the Great Pianists

Asmir Tobudic

Austrian Research Institute for
Artificial Intelligence, Vienna
asmir.tobudic@ofai.at

Gerhard Widmer

Department of Computational Perception,
Johannes Kepler University, Linz
Austrian Research Institute for
Artificial Intelligence, Vienna
gerhard.widmer@jku.at

Abstract

An application of relational instance-based learning to the complex task of expressive music performance is presented. We investigate to what extent a machine can automatically build ‘expressive profiles’ of famous pianists using only minimal performance information extracted from audio CD recordings by pianists and the printed score of the played music. It turns out that the machine-generated expressive performances on unseen pieces are substantially closer to the real performances of the ‘trainer’ pianist than those of all others. Two other interesting applications of the work are discussed: recognizing pianists from their style of playing, and automatic style replication.

1 Introduction

Relational instance-based learning is a machine learning paradigm that tries to transfer the successful nearest-neighbor or instance-based learning (IBL) framework to richer first-order logic (FOL) representations [Emde and Wettschereck, 1996; Tobudic and Widmer, 2005]. As such it is a part of inductive logic programming (ILP), the field of research which – after the euphoria in the nineties – suffered a certain flattening of interest in recent years, the main reason being the difficulties of effectively constraining the extremely large hypothesis spaces. Nevertheless, some ILP systems have recently been shown to achieve performance levels competitive to those of human experts in very complex domains (e.g. [King *et al.*, 2004]).

A successful application of relational IBL to a real-world task from classical music has been presented in previous work ([Tobudic and Widmer, 2005]). A system that predicts expressive timing and dynamics patterns for new pieces by analogy to similar musical passages in a training set has been shown to learn to play piano music ‘expressively’ with substantial musical quality.

Here we investigate an even more interesting question: can a relational learner learn models that are ‘personal’ enough to capture some aspects of the playing styles of truly great pianists? A system is presented that builds performance models of six famous pianists using only crude information related to expressive timing and dynamics obtained from the pianist’s

CD recordings and the printed score of the music. Experiments show that the system indeed captures some aspect of the pianists’ playing style: the machine’s performances of unseen pieces are substantially closer to the real performances of the ‘training’ pianist than those of all other pianists in our data set. An interesting by-product of the pianists’ ‘expressive models’ is demonstrated: the automatic identification of pianists based on their style of playing. And finally, the question of automatic style replication is briefly discussed.

The rest of the paper is laid out as follows. After a short introduction to the notion of expressive music performance (Section 2), Section 3 describes the data and its representation in FOL. We also discuss how the complex task of learning expressive curves can be decomposed into a well-defined instance-based learning task and shortly recapitulate the details of the relational instance-based learner DISTALL. Experimental results are presented in Section 4.

2 Expressive Music Performance

Expressive music performance is the art of shaping a musical piece by continuously varying important parameters like tempo, loudness, etc. while playing [Widmer *et al.*, 2003]. Human musicians do not play a piece of music mechanically, with constant tempo or loudness, exactly as written in the printed music score. Rather, skilled performers speed up at some places, slow down at others, stress certain notes or passages etc. The most important parameter dimensions available to a performer are *timing* (tempo variations) and *dynamics* (loudness variations). The way these parameters ‘should be’ varied is not specified precisely in the printed score; at the same time it is what makes a piece of music come alive, and what distinguishes great artists from each other.

Tempo and loudness variations can be represented as curves that quantify these parameters throughout a piece relative to some reference value (e.g. the average loudness or tempo of the same piece). Figure 1 shows a *dynamics curve* of a small part of Mozart’s Piano Sonata K.280, 1st movement, as played by five famous pianists. Each point gives the relative loudness (relative to the average loudness of the piece by the same performer) at a given point in the piece. A purely mechanical (unexpressive) rendition of the piece would correspond to a flat horizontal line at $y = 1.0$. Tempo variations can be represented in an analogous way. In the next section we discuss how predictive models of such curves can be auto-

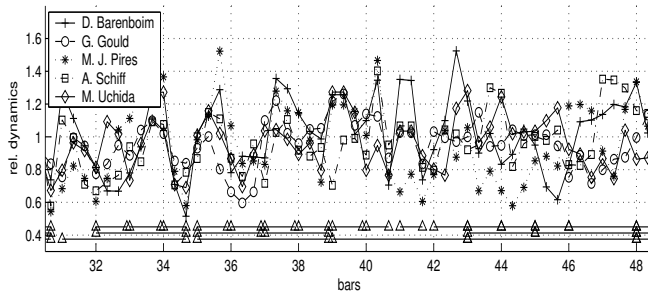


Figure 1: Dynamics curves of performances of five famous pianists for Mozart Sonata K.280, 1st mvt., mm. 31–48. Each point represents the relative loudness at the beat level (relative to the average loudness of the piece by the same performer).

matically learned from information extracted from audio CD recordings and the printed score of the music.

3 Data and Methodology

The data used in this work was obtained from commercial recordings of famous concert pianists. We analyzed the performances of 6 pianists across 15 different sections of piano sonatas by W.A.Mozart. The pieces selected for analysis are complex, different in character, and represent different tempi and time signatures. Tables 1 and 2 summarize the pieces, pianists and recordings selected for analysis.

For learning tempo and dynamics ‘profiles’ of the pianists in our data set we extract time points from the audio recordings that correspond to *beat*¹ locations. From the (varying) time intervals between these points, the beat-level tempo and its changes can be computed. Beat-level dynamics is computed from the audio signal as the overall loudness of the signal at the beat times as a very crude representation of the dynamics applied by the pianists. Extracting such information from the CD recordings was an extremely laborious task, even with the help of an intelligent interactive beat tracking system [Dixon, 2001]. From these measurements, computing pianists’ dynamics and tempo performance curves as shown in Figure 1 – which are the raw material for our experiments – is rather straightforward.

An examination of the dynamics curves in Figure 1 reveals certain trends common for all pianists (e.g. up-down, *crescendo-decrescendo* tendencies). These trends reflect certain aspects of the underlying *structure* of the piece: A piece of music commonly consists of *phrases* – segments that are heard and interpreted as coherent units. Phrases are organized hierarchically: smaller phrases are grouped into higher-level phrases, which are in turn grouped together, constituting a musical context at a higher level of abstraction etc. In Figure 1, the hierarchical, three-level phrase structure of this passage is indicated by three levels of brackets at the bottom. In this work we aim at learning expressive patterns at different levels of such a phrase structure, which roughly corresponds to various levels of musical abstraction.

¹The beat is the time points where listeners would tap their foot along with the music.

Table 1: Mozart sonata sections selected for analysis. Section ID should be read as <sonataName> : <movement> : <section>. The total numbers of phrases are also shown.

Section ID	Tempo descr.	#phrases
kv279:1:1	fast 4/4	460
kv279:1:2	fast 4/4	753
kv280:1:1	fast 3/4	467
kv280:1:2	fast 3/4	689
kv280:2:1	slow 6/8	129
kv280:2:2	slow 6/8	209
kv280:3:1	fast 3/8	324
kv280:3:2	fast 3/8	448
kv282:1:1	slow 4/4	199
kv282:1:2	slow 4/4	254
kv283:1:1	fast 3/4	455
kv283:1:2	fast 3/4	519
kv283:3:1	fast 3/8	408
kv283:3:2	fast 3/8	683
kv332:2	slow 4/4	549

Table 2: Pianists and recordings.

ID	Pianist name	Recording
DB	Daniel Barenboim	EMI Classics CDZ 7 67295 2, 1984
RB	Roland Batik	Gramola 98701-705, 1990
GG	Glenn Gould	Sony Classical SM4K 52627, 1967
MP	Maria João Pires	DGG 431 761-2, 1991
AS	András Schiff	ADD (Decca) 443 720-2, 1980
MU	Mitsuko Uchida	Philips Classics 464 856-2, 1987

3.1 Phrase Representation in FOL

Phrases and relations between them can be naturally represented in first-order logic. In our collection of pieces, phrases are organized at three hierarchical levels, based on a manual phrase structure analysis. The musical content of each phrase is encoded in the predicate *phrCont/18*. It has the form *phrCont(Id,A1,A2,...)*, where *Id* is the phrase identifier and *A1,A2,...* are attributes that describe very basic phrase properties. The first seven of these are numeric: the length of a phrase, the relative position of the highest melodic point (the ‘apex’), the melodic intervals between starting note and apex, and between apex and ending note, metrical strengths of starting note, apex, and ending note. The next three attributes are discrete: the harmonic progression between start, apex, and end, and two boolean attributes that state whether the phrase ends with a ‘cumulative rhythm’, and whether it ends with a cadential chord sequence. The remaining attributes describe – in addition to some simple information about global tempo and the presence of trills – global characteristics of the phrases in statistical terms: mean and variance of the durations of the melody notes within the phrase (as rough indicators of the general ‘speed’ of events and of durational variability), and mean and variance of the sizes of the melodic intervals between the melody notes (as measures of the ‘jumpiness’ of the melodic line).

This propositional representation ignores an essential aspect of the music: its temporal nature. The temporal re-

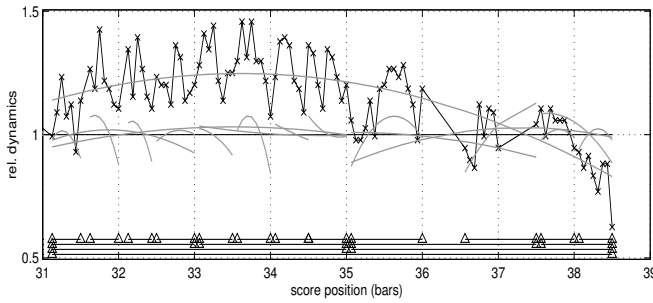


Figure 2: Multilevel decomposition of dynamics curve of performance of Mozart Sonata K.279:1:1, mm.31-38: original dynamics curve plus the second-order polynomial shapes giving the best fit at four levels of phrase structure.

relationships between successive phrases can be naturally expressed in FOL, as a relational predicate *succeeds*(*Id2*,*Id1*), which simply states that the phrase *Id2* succeeds the same-level phrase *Id1*. Supplying the same information in a propositional representation would be very difficult.

What is still needed in order to learn are the training examples, i.e. for each phrase in the training set, we need to know how it was played by a musician. This information is given in the predicate *phrShape*(*Id*,*Coeffs*), where *Coeffs* encode information about the way the phrase was played by a pianist. This is computed from the tempo and dynamics curves, as described in the following section.

3.2 Deriving the Training Instances: Multilevel Decomposition of Performance Curves

Given a complex tempo or dynamics curve and the underlying phrase structure (see Figure 1), we need to calculate the most likely contribution of each phrase to the overall observed expression curve, i.e., we need to decompose the complex curve into basic expressive phrase ‘shapes’. As approximation functions to represent these shapes we decided to use the class of second-degree polynomials (i.e., functions of the form $y = ax^2 + bx + c$), because there is ample evidence from research in musicology that high-level tempo and dynamics are well characterized by quadratic or parabolic functions [Todd, 1992]. Decomposing a given expression curve is an iterative process, where each step deals with a specific level of the phrase structure: for each phrase at a given level, we compute the polynomial that best fits the part of the curve that corresponds to this phrase, and ‘subtract’ the tempo or dynamics deviations ‘explained’ by the approximation. The curve that remains after this subtraction is then used in the next level of the process. We start with the highest given level of phrasing and move to the lowest. As tempo and dynamics curves are lists of multiplicative factors (relative to a default tempo), ‘subtracting’ the effects predicted by a fitted curve from an existing curve simply means dividing the *y* values on the curve by the respective values of the approximation curve.

Figure 2 illustrates the result of the decomposition process on the last part (mm.31–38) of the Mozart Sonata K.279, 1st movement, 1st section. The four-level phrase structure our music analyst assigned to the piece is indicated by the four

levels of brackets at the bottom of the plot. The elementary phrase shapes (at four levels of hierarchy) obtained after decomposition are plotted in gray. We end up with a training example for each phrase in the training set — a predicate *phrShape*(*Id*,*Coeff*), where *Coeff* = {*a*,*b*,*c*} are the coefficients of the polynomial fitted to the part of the performance curve associated with the phrase.

Input to the learning algorithm are the (relational) representation of the musical scores plus the training examples (i.e. timing and dynamics polynomials), for each phrase in the training set. Given a test piece the learner assigns the shape of the most similar phrase from the training set to each phrase in the test piece. In order to produce final tempo and dynamics curves, the shapes predicted for phrases at different levels must be combined. This is simply the inverse of the curve decomposition problem: Given a new piece, the system starts with an initial ‘flat’ expression curve (i.e., a list of 1.0 values) and then successively multiplies the current values by the multi-level phrase predictions.

3.3 DISTALL, a Relational Instance-based Learner

We approach phrase-shape prediction with a straightforward nearest-neighbour (NN, IBL) method. Standard propositional *k*-NN is not applicable to our data representation discussed in section 3.1. Instead, we use DISTALL, an algorithm that generalizes propositional *k*-NN to examples described in first-order logic [Tobudic and Widmer, 2005].

DISTALL is a representative of the line of research first initiated in [Bisson, 1992], where a clustering algorithm together with its similarity measure was presented. This work was later improved in [Emde and Wettschereck, 1996], in the context of the relational instance-based learning algorithm RIBL, which in turn can be regarded as DISTALL’s predecessor. We skip technical details here, but the main idea behind DISTALL’s similarity measure is that the similarity between two objects depends not only on the similarities of their attributes, but also on the similarities of the objects related to them. The similarities of the related objects depend in turn on their attributes and related objects. For our music learning task it means that the ‘shaping’ of the current (test) phrase depends not only on its attributes, but also on the preceding and succeeding music (through the relation *succeeds*(*Id1*,*Id2*), see section 3.1), which is – from a musical point of view – a rather intuitive idea. For a more detailed description of DISTALL see [Tobudic and Widmer, 2005].

Experimental results with DISTALL on MIDI-like (i.e., very detailed) performance data produced by a local pianist are reported in [Tobudic and Widmer, 2005]. The new contribution of the current paper is that we have laboriously measured audio recordings by truly famous artists and can show — for the first time — that DISTALL actually succeeds in capturing something of personal artistic performance style.

4 Experiments

4.1 Learning Predictive Performance Models

For each pianist, we conducted a systematic *leave-one-piece-out* cross-validation experiment: each of 15 pieces was once

Table 3: Results of piecewise cross-validation experiment. The table cells list correlations between learned and real curves, where rows indicate the ‘training pianist’, and columns the pianist whose real performance curves are used for comparison. The correlations are averaged over all pieces, weighted by the relative length of the piece. Each cell is further divided into two rows corresponding to *dynamics* and *tempo* correlations, respectively. The highest correlations in each row are printed in bold.

learned from	compared with					
	DB	RB	GG	MP	AS	MU
DB	.44	.21	.26	.34	.38	.28
	.44	.27	.26	.32	.31	.31
RB	.21	.32	.09	.19	.19	.17
	.28	.42	.20	.22	.30	.27
GG	.25	.09	.36	.19	.21	.22
	.25	.18	.32	.23	.29	.28
MP	.33	.19	.19	.39	.33	.28
	.31	.23	.27	.38	.28	.34
AS	.36	.17	.20	.31	.40	.26
	.32	.29	.28	.25	.41	.32
MU	.27	.18	.21	.28	.26	.38
	.34	.30	.32	.36	.37	.50

left aside as a test piece while the remaining 14 performances (by the same pianist) were used for learning. DISTALL’s parameter for the number of nearest neighbors was set to 1, and the parameter for the depth of starting clauses (see [Tobudic and Widmer, 2005]) to 4 (meaning that the distance between two phrases can be influenced by at most 4 preceding and 4 succeeding phrases).

The expressive shapes for each phrase in a test piece are predicted by DISTALL and then combined into a final tempo and dynamics curve, as described in section 3.2. The resulting curves are then compared to the real performance curves of all pianists (for the same test piece). If the curve learned from the performances of one pianist is more similar to the real performance curve of the ‘teacher’ pianist than to those of all other pianists, we could conclude that the learner succeeded in capturing something of the pianist’s specific playing style. The described procedure is repeated for all pieces and all pianists in our data set.

Correlation is chosen as a measure of how well the predicted curve ‘follows’ the real one. The curves are first normalized so that their autocorrelations are identically 1, giving a correlation estimate between curves as a number in the range [-1,1]. The results of the cross-validation experiment averaged over all pieces (weighted by the relative length of the pieces) are given in Table 3.

Interestingly, the system succeeded in learning curves that are substantially closer to the ‘trainer’ than all others, for all pianists. Some of the pianists are better ‘predictable’ than others, e.g. Daniel Barenboim and Mitsuko Uchida, which might indicate that they play Mozart in a more ‘consistent’ way. While at first sight the correlations may not seem impressive, one should keep in mind that artistic performance is far from predictable, and the numbers in Table 3 are averages over all pieces (about half an hour of concert-level piano

Table 4: Detailed results (tempo dimension) of the cross-validation experiment with Mitsuko Uchida was the ‘training’ pianist. For each piece, the correlations between predicted curve and actual tempo curves from all pianists are given. The average over all pieces is given in the last row (reproduced from the last row of Table 3)

Piece	DB	RB	GG	MP	AS	MU
kv279:1:1	.36	.31	.22	.36	.29	.50
kv279:1:2	.31	.35	.35	.28	.34	.47
kv280:1:1	.40	.25	.26	.55	.42	.78
kv280:1:2	.42	.37	.42	.37	.47	.54
kv280:2:1	.53	.34	.52	.58	.59	.69
kv280:2:2	.44	.13	.39	.42	.50	.58
kv280:3:1	.72	.66	.62	.82	.68	.77
kv280:3:2	.51	.52	.44	.49	.43	.51
kv282:1:1	.38	.45	.36	.44	.44	.72
kv282:1:2	.23	.36	.33	.42	.46	.52
kv283:1:1	.21	.10	.12	.16	.22	.31
kv283:1:2	.17	.18	.21	.23	.22	.32
kv283:3:1	.30	.28	.42	.42	.42	.52
kv283:3:2	.19	.15	.29	.21	.32	.36
kv332:2	.32	.17	.14	.22	.16	.35
Total	.34	.30	.32	.36	.37	.50

music per artist). Moreover, the correlation estimates in Table 3 are somewhat unfair, since we compare the performance curve produced by composing the polynomials predicted by the learner, with the curve corresponding to the pianists’ *actual* performances. However, what DISTALL learned from was not the actual performance curves, but an *approximation* curve which is implied by the three levels of quadratic functions that were used as training examples. Correctly predicting these is the best the learner could hope to achieve.

Table 4 shows a more detailed picture of the cross-validation experiment, where the training pianist was Mitsuko Uchida and the numbers refer to correlations in the tempo domain. In 13 out of 15 cases the learner produces tempo curves which are closer to Uchida’s playing than to any other pianist, with correlations of .7 and better (e.g., for kv280:3:1 and kv280:1:1). The results are even more interesting if we recall that the learner is given a very crude, beat-level representation of the tempo and dynamics applied by the pianist, without any details about e.g. individual voices or timing details below the level of beat. On the other hand, the piecewise results revealed that some of the pianists (e.g. Gould or Pires) seem to be less ‘predictable’ with our approach than Uchida (not reported here for lack of space).

Figure 3 shows an example of successful performance style learning. We see a passage from a Mozart piano sonata as ‘played’ by the computer after learning from recordings of other pieces by Daniel Barenboim (top) and Mitsuko Uchida (bottom), respectively. Also shown are the performance curves corresponding to these two pianists’ actual performances of the test piece. In this case it is quite clearly visible that the curves predicted by the computer on the test piece are much more similar to the curves by the respective ‘teacher’ than to those by the other pianist.

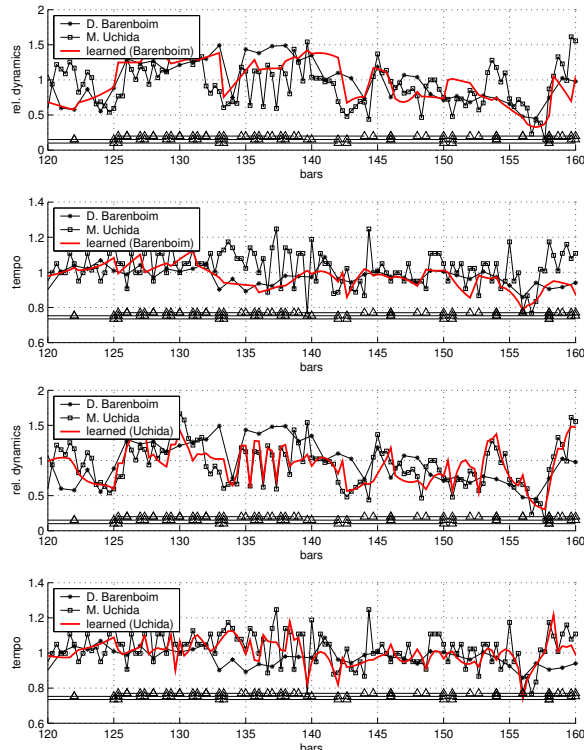


Figure 3: Dynamics and tempo curves produced by DISTALL on test piece (Sonata K.283, 3rd mvt., 2nd section, mm.120–160) after learning from Daniel Barenboim (top) and Mitsuko Uchida (bottom), compared to the artists’ real curves as measured from the recordings.

Admittedly, this is a carefully selected example, one of the clearest cases of style replication we could find in our data. The purpose of this example is more to give an indication of the complexity of the curve prediction task and the difference between different artists’ interpretations than to suggest that a machine will always be able to achieve this level of prediction performance.

4.2 Identification of Great Pianists

The primary goal of our work is learning predictive models of pianists’ expressive performances.² But the models can also be used in a straightforward way for recognizing pianists. The problem of identifying famous pianists from information obtained from audio recordings of their playing has been addressed in the recent literature [Saunders *et al.*, 2004; Stamatatos and Widmer, 2002; Widmer and Zanon, 2004]. In [Widmer and Zanon, 2004], a number of low-level scalar features related to expressive timing and dynamics are extracted from the audio CD recordings, and various machine learning

²Note that learned tempo and dynamics curves as produced by our system can be used to build truly machine generated expressive performances: using the predicted tempo and dynamics curves (i.e. relative tempo and dynamics for each beat in the piece), it is straightforward to calculate tempo and dynamics for each note in the piece (e.g. by interpolation).

Table 5: Confusion matrix of the pianist classification experiment. Rows correspond to the test performances of each pianist (15 per row), columns to the classifications made by the system. The rightmost column gives the accuracy achieved for all performances of the respective. The *baseline* accuracy in this 6-class problem is 16.67%.

pianist	prediction						Acc.[%]
	DB	RB	GG	MP	AS	MU	
DB	11	0	0	2	2	0	73.3
RB	1	12	1	0	0	1	80.0
GG	1	1	10	0	0	3	66.7
MP	0	0	1	12	0	2	80.0
AS	1	0	2	0	10	2	66.7
MU	0	0	1	0	0	14	93.3
Total	-	-	-	-	-	-	76.7

algorithms are applied to these. In [Saunders *et al.*, 2004], the sequential nature of music is addressed by representing performances as strings and using string kernels in conjunction with kernel partial least squares and support vector machines. The string kernel approach is shown to achieve better performance than the best results obtained in [Widmer and Zanon, 2004]. A clear result from both works is that identification of pianists from their recordings is an extremely difficult task.

The pianists studied in the present paper are identical to those in [Widmer and Zanon, 2004] and [Saunders *et al.*, 2004]; unfortunately, the sets of recordings differ considerably (because manual phrase structure analyses, which are needed in our approach, were available only for certain pieces), so a direct comparison of the results is impossible. Still, to illustrate what can be achieved with a relational representation and learning algorithm, we briefly describe a classification experiment with DISTALL.

Each of the 15 pieces is set aside once. The 6 performances of that piece (one by each pianist) are used as test instances. A model of each pianist is built from his/her performances of the remaining 14 pieces. The result is two predicted curves per pianist for the test piece (for tempo and dynamics), which we call model curves. The final classification of a pianist, represented by his/her tempo and dynamics curves t_t and t_d on the test piece, is then determined as

$$c(t_t, t_d) = \underset{p \in P}{\operatorname{argmax}} \left(\frac{\operatorname{corr}(t_t, m_{pt}) + \operatorname{corr}(t_d, m_{pd})}{2} \right) \quad (1)$$

where P is set of all pianists and m_{pt} and m_{pd} are the pianists’ model tempo and dynamics curves. In other words, the performance is classified as belonging to the pianist whose model curves exhibit the highest correlation (averaged over tempo and dynamics) with the test curves. For each pianist, DISTALL is tested on the 15 test pieces, which gives a total number of 90 test performances. The *baseline* accuracy – the success rate of pure guessing – is 15, or 16.67%. The confusion matrix of the experiment is given in Table 5.

Again, it turns out that the artists are identifiable to varying degrees, but the recognition accuracies are all clearly above the baseline. In particular, note that the system correctly iden-

tifies performances by Uchida in all but one case. Obviously, the learner succeeds in reproducing something of the artists' styles in its model curves. While these figures seem to compare very favourably to the accuracies reported in [Widmer and Zanon, 2004] and [Saunders *et al.*, 2004], they cannot be compared directly, because different recordings were used and, more importantly, the level of granularity of the training and test examples are different (movements in [Widmer and Zanon, 2004; Saunders *et al.*, 2004] vs. sections in this paper), which probably makes our learning task easier.

4.3 Replicating Great Pianists?

Looking at Figure 3, one might be tempted to consider the possibility of automatic style replication: wouldn't it be interesting to supply the computer with the score of a new piece and have it perform it 'in the style of', say, Vladimir Horowitz or Arthur Rubinstein? This question is invariably asked when we present this kind of research to the public. Unfortunately (?), the answer is: while it might be interesting, it is not currently feasible.

For one thing, despite the huge effort we invested in measuring expressive timing and dynamics in recordings, the amount of available training data is still vastly insufficient vis-a-vis the enormous complexity of the behaviour to be learned. And secondly, the sort of crude beat-level variations in tempo and general loudness capture only a very small part of what makes an expressive interpretation; essential details like articulation (e.g., staccato vs. legato), pedalling, the shaping of individual voices, etc. are missing (and will be very hard to measure from audio recordings at all). A computer performance based only on applying these beat-level tempo and loudness changes will not sound anything like a human performance, as can be readily verified experimentally. Thus we have to admit that the title we chose for this paper is a bit pretentious: the computer cannot be expected to play like the great pianists – at least not given the current methods and available training data. It can, however, extract aspects of personal style from recordings by great pianists, as has been shown in the our experiments.

5 Conclusion

An application of relational instance-based learning to a difficult task from the domain of classical music was presented: we showed how the problem of learning models of expressive piano performance can be reduced to applying simple expressive phrase-patterns by analogy to the most similar phrases in the training set. In particular, it was shown that a relational learner like DISTALL succeeds in learning performance strategies that obviously capture aspects individual artistic style, which was demonstrated in learning and artist classification experiments.

The ultimate goal of this kind of research is not automatic style replication or the creation of artificial performers, but to use computers to teach us more about the elusive artistic activity of expressive music performance. While it is satisfying to see that the computer is indeed capable of extracting information from performance measurements that seems to capture aspects of individual style, this can only be a first step.

In order to get real insight, we will need learning algorithms that, unlike nearest-neighbor methods, produce interpretable models. That is a natural next step in our future research.

Acknowledgments

This research is supported by the Austrian *Fonds zur Förderung der Wissenschaftlichen Forschung (FWF)* (project no. Y99-INF). The Austrian Research Institute for Artificial Intelligence acknowledges basic financial support by the Austrian Federal Ministry for Education, Science, and Culture, and the Federal Ministry of Transport, Innovation, and Technology. Thanks to Werner Goebel for performing the harmonic and phrase structure analysis of the Mozart sonatas.

References

- [Bisson, 1992] Gilles Bisson. Learning in FOL with a similarity measure. In *Proceedings of the 10th AAAI*, 1992.
- [Dixon, 2001] Simon Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 2001.
- [Emde and Wettschereck, 1996] Werner Emde and Dietrich Wettschereck. Relational instance-based learning. In *Proceedings of the 13th International Conference on Machine Learning*, 1996.
- [King *et al.*, 2004] R.D. King, K.E. Whelan, F.M. Jones, P.G.K. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, and S.G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 15 January 2004.
- [Saunders *et al.*, 2004] Craig Saunders, David R. Hardoon, John Shawe-Taylor, and Gerhard Widmer. Using string kernels to identify famous performers from their playing style. In *Proceedings of the 15th European Conference on Machine Learning*, 2004.
- [Stamatatos and Widmer, 2002] Efstathios Stamatatos and Gerhard Widmer. Music performer recognition using an ensemble of simple classifiers. In *Proceedings of the 15th European Conference on Artificial Intelligence*, 2002.
- [Tobudic and Widmer, 2005] Asmir Tobudic and Gerhard Widmer. Relational IBL in classical music. *Machine Learning*, to appear, 2005.
- [Todd, 1992] McAngus N. Todd. The dynamics of dynamics: A model of musical expression. *Journal of the Acoustical Society of America*, 91(6):3540–50, 1992.
- [Widmer and Zanon, 2004] Gerhard Widmer and Patrick Zanon. Automatic recognition of famous artists by machine. In *Proceedings of the 17th European Conference on Artificial Intelligence*, 2004.
- [Widmer *et al.*, 2003] Gerhard Widmer, Simon Dixon, Werner Goebel, Elias Pampalk, and Asmir Tobudic. In search of the Horowitz factor. *AI Magazine*, 24(3):111–130, 2003.

Sequential Genetic Search for Ensemble Feature Selection

Alexey Tsymbal
Dept. of Computer Science
Trinity College Dublin
Dublin 2, Ireland
tsymbalo@cs.tcd.ie

Mykola Pechenizkiy
Dept. of CS & ISs
University of Jyväskylä
P.O. Box 35, Finland-40351
mpechen@cs.jyu.fi

Pádraig Cunningham
Dept. of Computer Science
Trinity College Dublin
Dublin 2, Ireland
Padraig.Cunningham@cs.tcd.ie

Abstract

Ensemble learning constitutes one of the main directions in machine learning and data mining. Ensembles allow us to achieve higher accuracy, which is often not achievable with single models. One technique, which proved to be effective for constructing an ensemble of diverse classifiers, is the use of feature subsets. Among different approaches to ensemble feature selection, genetic search was shown to perform best in many domains. In this paper, a new strategy GAS-SEFS, Genetic Algorithm-based Sequential Search for Ensemble Feature Selection, is introduced. Instead of one genetic process, it employs a series of processes, the goal of each of which is to build one base classifier. Experiments on 21 data sets are conducted, comparing the new strategy with a previously considered genetic strategy for different ensemble sizes and for five different ensemble integration methods. The experiments show that GAS-SEFS, although being more time-consuming, often builds better ensembles, especially on data sets with larger numbers of features.

1 Introduction

A popular method for creating an accurate model from a set of training data is to construct a set (ensemble) of classifiers. It was shown that an ensemble is often more accurate than any of the single classifiers in it. The integration of classifiers is currently an active research area in the machine learning and neural networks communities [Dietterich, 1997].

Both theoretical and empirical research have demonstrated that a good ensemble should include diverse base classifiers. Another important issue in creating an effective ensemble is the choice of the function for combining the predictions of the base classifiers. It was shown that increasing the ensemble diversity is not enough to ensure increased accuracy – if the integration method does not properly utilize the ensemble diversity, then no benefit arises from the integration [Brodley and Lane, 1996].

One effective approach for generating an ensemble of diverse classifiers is the use of feature subsets, or *ensemble feature selection* [Opitz, 1999]. By varying the feature sub-

sets used to generate the base classifiers, it is possible to promote diversity and produce base classifiers that tend to err in different sub-areas of the instance space.

Feature selection algorithms, including ensemble feature selection, are typically composed of the following components [Aha and Bankert, 1995, Opitz, 1999]: (1) *search strategy*, that searches through the space of feature subsets; and (2) *fitness function*, that inputs a feature subset and outputs a numeric evaluation. The search strategy's goal is to find a feature subset maximizing this function.

It is reasonable to include in the fitness function, explicitly or implicitly, both accuracy and diversity. One measure of fitness, which was proposed in [Opitz, 1999], defines fitness $Fitness_i$ of classifier i corresponding to feature subset i to be proportional to classification accuracy acc_i and diversity div_i of the classifier:

$$Fitness_i = acc_i + \alpha \cdot div_i, \quad (1)$$

where α reflects the influence of diversity. Diversity div_i is the contribution of classifier i to the total ensemble diversity, which can be measured as the average pairwise diversity for all the pairs of classifiers including i . This fitness function was also used in experiments in [Tsymbal *et al.*, 2003; 2005], and it is used in the experiments in this paper.

In [Tsymbal *et al.*, 2005] a genetic search-based strategy GA has been introduced. It uses genetic search for evolving the initial population built with random subsampling. GA was shown to perform best on average with respect to the other three strategies, and two diversity measures were best for GA of the five considered measures: the kappa statistic, and the fail/non-fail disagreement.

In this paper, we introduce a new genetic search-based strategy for ensemble feature selection, GAS-SEFS, which, instead of maintaining a set of feature subsets in each generation like in GA, consists in applying a series of genetic processes, one for each base classifier, sequentially.

The paper is organized as follows. In Section 2 the task of ensemble feature selection is considered. In Section 3 we present two strategies for genetic ensemble feature selection, GA and GAS-SEFS, and two diversity measures. In Section 4 different methods for ensemble integration are reviewed. In Section 5 we present our experiments with the two genetic strategies and conclude in the next section with a summary and assessment of further research topics.

2 Ensemble Feature Selection and Random Subspacing

The task of using an ensemble of models can be broken down into two basic questions: (1) what set of models should be generated?; and (2) how should the predictions of the models be integrated? [Dietterich, 1997].

One effective approach to ensemble generation is the use of different subsets of features for each model. Finding a set of feature subsets for constructing an ensemble is also known as *ensemble feature selection* [Opitz, 1999]. While traditional feature selection algorithms have the goal of finding the best feature subset that is suitable to both the learning problem and the learning algorithm, the task of ensemble feature selection has the additional goal of finding a set of feature subsets that will promote diversity among the base classifiers [Opitz, 1999].

Ho [1998] has shown that simple random selection of features may be an effective technique for ensemble feature selection because the lack of accuracy in the ensemble members is compensated for by their diversity. This technique is called the random subspace method or simply Random Subspacing (RS).

With RS one may solve the small sample size problem, because the training sample size relatively increases in random subspaces. Ho [1998] shows that while most other classification methods suffer from the curse of dimensionality, this method does not. RS has much in common with bagging [Skurichina and Duin, 2001], but instead of sampling instances, one samples features. Like bagging, RS is a parallel learning algorithm, that is, generation of each base classifier is independent. This makes it suitable for parallel implementation that is desirable in some practical applications. It was shown that, like in bagging, accuracy could be only increased with the addition of new members, even when the ensemble complexity grew [Ho, 1998].

RS is used as a base in a number of ensemble feature selection strategies, e.g. GEFS (Genetic Ensemble Feature Selection) [Opitz, 1999] and HC (Hill Climbing) [Cunningham and Carney, 2000].

3 Two Strategies for Genetic Ensemble Feature Selection

3.1 GA and GAS-SEFS

The use of genetic search has been an important direction in the feature selection research. Genetic algorithms have been shown to be effective global optimization techniques. The use of genetic algorithms for ensemble feature selection was first proposed in [Kuncheva, 1993] and further elaborated in [Kuncheva and Jain, 2000]. As the fitness function in [Kuncheva, 1993; Kuncheva and Jain, 2000] the ensemble accuracy was used instead of the accuracy of the base classifiers. However, such a fitness function is biased towards some particular integration method (often simple voting). Besides, as it was shown e.g. in [Kuncheva, 1993], such a design is prone to overfitting, and some additional preventive measures are needed to be taken to avoid this (as in-

cluding in the fitness function penalty terms accounting for the number of features). The use of individual accuracy and diversity as in (1) is an alternative solution to this problem. Another motivation for this alternative is the fact that overfitting at the level of the base classifiers is more desirable than overfitting of the ensemble itself. It was shown recently in several studies that an ensemble of overfitted members might often be better than an ensemble of non-overfitted members. For example, in [Street and Kim, 2001] pruning trees resulted in decreased ensemble accuracy, even though the accuracy of the trees themselves increased.

The Genetic Algorithm for ensemble feature selection (GA) [Tsybmal *et al.*, 2005] is based on the GEFS strategy of Opitz [1999]. GEFS was the first genetic algorithm for ensemble feature selection that explicitly used diversity in its fitness function. GA begins with creating an initial population with RS. Then, new candidate classifiers are produced by crossover and mutation. After producing a certain number of individuals the process continues with selecting a new subset of candidates randomly with a probability proportional to fitness (so called roulette-wheel selection). The process of creating new classifiers and selecting a subset of them (a generation) continues a certain number of times. After a predefined number of generations, the fittest individuals make up the population, which comprises the ensemble. The representation of each individual is a constant-length string of bits, where each bit corresponds to a particular feature. The crossover operator uses uniform crossover, in which each feature of the two children takes randomly a value from one of the parents. The mutation operator randomly toggles a number of bits in an individual.

Instead of maintaining a set of feature subsets in each generation of one genetic process, GAS-SEFS (Genetic Algorithm-based Sequential Search for Ensemble Feature Selection) uses a series of genetic processes, one for each base classifier, sequentially. Pseudo-code for GAS-SEFS is given in Figure 1. After each genetic process one base classifier is selected into the ensemble. GAS-SEFS uses the same fitness function (1), but diversity is calculated with the base classifiers already formed by previous genetic processes instead of the members of current population. In the first GA process, the fitness function has to use accuracy only. GAS-SEFS uses the same genetic operators as GA.

```
For I=1 to EnsembleSize
  For J=1 to 10 Population(J)=RSM(#Features);
  For J=1 to #Generations
    For K=1 to 10 CalculateFitness(Population(K));
    For K=1 to 40
      //randomly proportional to log(1+fitness)
      (L,M)=Select2(Population);
      Offsprings(K)=Crossover(L,M);
    EndForK
    For K=1 to 10 Mutate1_0(Offsprings(20+K));
      Mutate0_1(Offsprings(30+K));
    For K=1 to 40 CalculateFitness(Offsprings(K));
      //randomly proportional to fitness
      Population=Select10(Population+Offsprings);
    EndForJ
    //according to fitness
    BaseClassifier(I)=Select1(Population);
  EndForJ
EndForI
```

Figure 1 Pseudo-code for the GAS-SEFS algorithm

GA has a number of peculiarities, which we use also in GAS-SEFS. Full feature sets are not allowed in RS nor may the crossover operator produce a full feature subset. Individuals for crossover are selected randomly proportional to $\log(1+fitness)$ instead of just fitness, which adds more diversity into the new population. The generation of children identical to their parents is prohibited. To provide a better diversity in the length of feature subsets, two different mutation operators are used (*Mutate1_0* and *Mutate0_1*), one of which always deletes features randomly with a given probability, and the other – adds features.

Parameter settings for our implementation of GA and GAS-SEFS include a mutation rate of 50%, a population size of 10, a search length of 40 feature subsets (the number of new individuals produced by crossover and mutation), of which 20 are offsprings of the current population of 10 classifiers generated by crossover, and 20 are mutated offsprings (10 with each mutation operator). 10 generations of individuals were produced, as our pilot studies have shown that in most cases, with this configuration, the ensemble accuracy does not improve after 10 generations, due to overfitting the training data.

The complexity of GA does not depend on the number of features, and is $O(S' \cdot N_{gen})$, where S' is the number of individuals in one generation, and N_{gen} is the number of generations [Tsymbal *et al.*, 2005]. The complexity of GAS-SEFS is $O(S \cdot S' \cdot N_{gen})$, where S is the number of base classifiers. In our experiments, on average, GA and GAS-SEFS look through about 400 and 4000 feature subsets correspondingly (given that the number of base classifiers is 10, the number of individuals in a generation is 40, and the number of generations is 10).

3.2 Diversity measures used in the fitness function

The fail/non-fail disagreement measure and the kappa statistic were shown to provide the best performance for GA in [Tsymbal *et al.*, 2005].

The *fail/non-fail disagreement* measure was defined in [Skalak, 1996] as the percentage of test instances for which the classifiers make different predictions but for which one of them is correct:

$$div_dis_{ij} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}, \quad (2)$$

where N^{ab} is the number of instances, classified correctly ($a=1$) or incorrectly ($a=0$) by classifier i , and correctly ($b=1$) or incorrectly ($b=0$) by classifier j . The denominator in (2) is equal to the total number of instances. The fail/non-fail disagreement varies from 0 to 1.

The *kappa* statistic was first introduced in [Cohen, 1960]. Let N_{ij} be the number of instances, recognized as class i by the first classifier and as class j by the second one, N_{i*} is the number of instances recognized as i by the first classifier, and N_{*j} is the number of instances recognized as j by the second classifier. Define then Θ_1 and Θ_2 as:

$$\Theta_1 = \frac{\sum_{i=1}^l N_{ii}}{N}, \text{ and } \Theta_2 = \sum_{i=1}^l \left(\frac{N_{i*}}{N} \cdot \frac{N_{*i}}{N} \right), \quad (3)$$

where l is the number of classes and N is the total number of instances. Θ_1 estimates the probability that the two classifiers agree, and Θ_2 is a correction term, which estimates the probability that the two classifiers agree simply by chance (in the case where each classifier chooses to assign a class label randomly). The pairwise diversity div_kappa_{ij} is defined as follows:

$$div_kappa_{ij} = \frac{\Theta_1 - \Theta_2}{1 - \Theta_2}. \quad (4)$$

We normalize this measure to vary from 0 to 1.

4 Integration of an Ensemble of Models

The challenging problem of integration is to decide which of the classifiers to select or how to combine the results produced by the base classifiers. A number of *selection* and *combination* approaches have been proposed.

One of the most popular and simplest techniques used to combine the results of the base classifiers, is simple voting (also called majority voting) [Bauer and Kohavi, 1999]. In the voting, the output of each base classifier is considered as a vote for that particular class value. The class value that receives the biggest number of votes is selected as the final classification. Weighted Voting (WV), where each vote has a weight proportional to the estimated generalization performance of the corresponding classifier, works usually better than simple voting [Bauer and Kohavi, 1999].

A number of selection techniques have also been proposed to solve the integration problem. One of the most popular and simplest selection techniques is Cross-Validation Majority (CVM, also called Single Best; we call it simply Static Selection, SS, in our experiments) [Schaffer, 1993]. In CVM, the cross-validation accuracy for each base classifier is estimated, and then the classifier with the highest accuracy is selected.

The described above approaches are *static*. They select one model for the whole data space or combine the models uniformly. In *dynamic* integration each new instance to be classified is taken into account. Usually, better results can be achieved if integration is dynamic.

We consider in our experiments three dynamic techniques based on the same local error estimates: Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS) [Tsymbal and Puuronen, 2000]. They contain two main phases. First, at the learning phase, the local classification errors of each base classifier for each instance of the training set are estimated according to the 1/0 loss function using cross validation. The learning phase finishes with training the base classifiers on the whole training set. The application phase begins with determining k -nearest neighbours for a new instance using a given distance metric. Then, weighted nearest neighbour regression is used to predict the local classification errors of each base classifier for the new instance.

After, DS simply selects a classifier with the least predicted local classification error. In DV, each base classifier receives a weight that is proportional to its estimated local accuracy, and the final classification is produced as in WV. In DVS, the base classifiers with the highest local classification errors are discarded (the classifiers with errors that fall into the upper half of the error interval) and locally weighted voting (DV) is applied to the remaining classifiers.

5 Experimental Investigations

5.1 Experimental setup

The experiments are conducted on 21 data sets taken from the UCI machine learning repository [Blake *et al.*, 1999]. These data sets include real-world and synthetic problems, vary in characteristics, and were previously investigated by other researchers. The main characteristics of the data sets are presented in Table 1.

Table 1 Data sets and their characteristics

Data set	Instances	Classes	Features	
			Categ.	Num.
Balance	625	3	0	4
Breast Cancer	286	2	9	0
Car	1728	4	6	0
Diabetes	768	2	0	8
Glass Recognition	214	6	0	9
Heart Disease	270	2	0	13
Ionosphere	351	2	0	34
Iris Plants	150	3	0	4
LED	300	10	7	0
LED17	300	10	24	0
Liver Disorders	345	2	0	6
Lymphography	148	4	15	3
MONK-1	432	2	6	0
MONK-2	432	2	6	0
MONK-3	432	2	6	0
Soybean	47	4	0	35
Thyroid	215	3	0	5
Tic-Tac-Toe	958	2	9	0
Vehicle	846	4	0	18
Voting	435	2	16	0
Zoo	101	7	16	0

As in [Tsybal *et al.*, 2003; 2005], we use Simple Bayes (SB) as the base classifier in the ensembles. It has been recently shown experimentally and theoretically that SB can be optimal even when the “naïve” feature-independence assumption is violated by a wide margin [Domingos and Pazzani, 1997]. Second, when SB is applied to the subproblems of lower dimensionalities, the error bias of the Bayesian probability estimates caused by the feature-independence assumption becomes smaller. It also can easily handle missing feature values. Besides, it has advantages in terms of simplicity, learning speed, classification speed, and storage space. We believe that dependencies and conclusions presented in this paper do not depend on the learning algorithm used and would be similar for most known learning algorithms.

To evaluate GA and GAS-SEFS, we have used stratified random-sampling cross validation with 60 percent of instances in the training set. The remaining 40 percent of in-

stances were divided into two sets of approximately equal size (a validation set and a test set). 70 test runs of were made on each data set for each search strategy and diversity.

Four different ensemble sizes have been tested: 3, 5, 7, and 10. The ensemble size did not exceed 10 due to two main reasons: (1) limitation in computational resources, and (2) it was shown in experiments that for guided ensemble construction such as genetic search the biggest gain is achieved already with 10 base classifiers, and much less classifiers are needed than with unguided ensemble construction such as RS and bagging.

At each run of the algorithm, accuracies for the five types of ensemble integration are collected: Static Selection (SS), Weighted Voting (WV), Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS). We have collected ensemble characteristics for four numbers of generations: 1, 3, 5, and 10.

To reduce the number of possible combinations of parameters, we conducted a separate series of preliminary experiments using the wrapper approach based on cross validation to select the best diversity coefficient α and the number of nearest neighbors in dynamic integration k as in [Tsybal *et al.*, 2005]. We have experimented with seven values of α : 0, 0.25, 0.5, 1, 2, 4, and 8. Seven values: 1, 3, 7, 15, 31, 63, 127 ($2^n - 1$, $n = 1, \dots, 7$) were used for k . From the experimental results we could see that the best value of k depended mostly only on the integration method used and on the data set. The best α 's varied with the search strategy, integration method, and data set used.

After, the experiments were repeated with the selected values of α and k . Although the same data were used for the selection and for the later experiments, we believe that this did not lead to overfitting due to the small number of possible values for α and k .

The test environment was implemented within the MLC++ framework (the machine learning library in C++) [Kohavi *et al.*, 1996]. A multiplicative factor of 1 was used for the Laplace correction in SB as in [Domingos and Pazzani, 1997]. Numeric features were discretized into ten equal-length intervals (or one per observed value, whichever was less), as it was done in [Domingos and Pazzani, 1997]. Although this approach was found to be slightly less accurate than more sophisticated ones, it has the advantage of simplicity, and is sufficient for comparing different ensembles of SB classifiers with each other.

5.2 Experimental results

To validate our findings, we divided all data sets into two groups: with less than 9 features (10 data sets, group 1), and with greater than or equal to 9 features (11 data sets, group 2); and checked all the characteristics for these groups. The ensemble accuracies were nearly the same for the two diversity measures, and the fail/non-fail disagreement was slightly better on average, so we present results for this diversity measure only here. In Figure 2 the ensemble accuracies for strategies GA and GAS-SEFS, over the two groups of data sets and four ensemble sizes are shown averaged over the data sets for the best integration method (DVS). It

can be seen from the figure that GAS-SEFS builds even more accurate ensembles than GA; especially for group 2 including data sets with larger numbers of features. Accuracy grows with the ensemble size, but the growth flattens as the number of base classifiers increases.

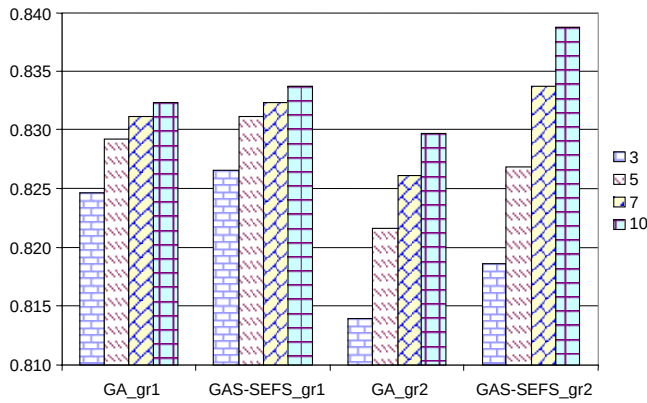


Figure 2 Ensemble accuracies for two strategies, two groups of data sets, and four ensemble sizes

In Figure 3 ensemble accuracies are shown for the two strategies, five integration methods, and four ensemble sizes on the Tic-Tac-Toe data set, as a representative of group 2 including 958 instances and 9 features. This figure supports our previous findings. Besides, it could be seen that dynamic integration, expectedly, outperforms static integration both for GA and for GAS-SEFS. Accuracy grows with the ensemble size and this growth is greater for the best integration methods (DS and DVS in this case).

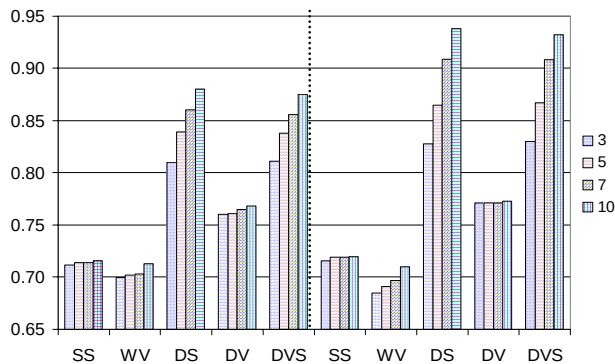


Figure 3 Ensemble accuracies for GA (left) and GAS-SEFS (right) for five integration methods and four ensemble sizes on the Tic-Tac-Toe data set

The difference between the two strategies is clearer for the best integration methods. The dependencies were the same for all the data sets, with sometimes lesser difference between the integration methods. For some data sets DV outperforms DS, which supports the previous findings about behaviour of the integration methods [Tsybmal and Puuronen, 2000; Tsybmal *et al.*, 2005].

5.3 Other interesting findings

Selected values of α were different for different data sets, supporting findings in [Tsybmal *et al.*, 2003; 2005]. In general, for both strategies, α for the dynamic integration methods is bigger than for the static ones (2.2 vs 0.8 on average). GAS-SEFS needs slightly higher values of α than GA (1.8 vs 1.5 on average). This can be explained by the fact that GAS-SEFS always starts with a classifier, which is based on accuracy only, and the subsequent classifiers need more diversity than accuracy.

The number of selected features falls as the ensemble size grows, and this is especially clear for GAS-SEFS, as the base classifiers need more diversity. As a rule, more features are needed in the static integration methods than in the dynamic ones to achieve better accuracy. GAS-SEFS results in slightly smaller feature subsets on average (48% vs 50% of features for dynamic integration strategies).

As it was also reported in [Tsybmal *et al.*, 2005], the selected k -neighbourhood values for dynamic integration change with the integration method. DS needs higher values of k . This can be explained by the fact that its prediction is based on only one classifier being selected, and thus, it is very unstable. Higher values of k provide more stability to DS. The average selected k is equal to 33 for DS, and it is only 14 for DV. For DVS, as a hybrid strategy, it is in between at 24. The selected values of k do not change significantly with the change of the search strategy and the ensemble size.

Experimental results for both GA and GAS-SEFS show that the static integration methods, SS and WV, and the dynamic DS start to overfit the validation set already after 5 generations and show lower accuracies, whereas the accuracies of DV and DVS continue to grow up to 10 generations. This shows the importance of selection of the appropriate integration method for the genetic strategies.

6 Conclusions

In our paper, we have considered two genetic search strategies for ensemble feature selection. The new strategy, GAS-SEFS, consists in employing a series of genetic search processes, one for each base classifier. It was shown in experiments that GAS-SEFS results in better ensembles having greater accuracy in many domains, especially for data sets with relatively larger numbers of features. GAS-SEFS is significantly more time-consuming than GA. However, it can be easily parallelized in a multiprocessor setting, and one processor could be used for each offspring in the current generation.

One of the reasons for the success of GAS-SEFS is the fact that each of the core GA processes leads to significant overfitting of a corresponding ensemble member, and, as it was shown before, an ensemble of overfitted members is often better than an ensemble of non-overfitted members.

In [Oliveira *et al.*, 2003] it was shown that besides the use of weights to combine a number of objectives in the fitness function in genetic algorithms (as the use of α in our case), another common approach that often gives better results for

single feature subset selection is based on Pareto-front dominating solutions. Adaptation of this technique to ensemble feature selection is an interesting topic for further research.

Acknowledgments

This study is supported by Science Foundation Ireland and COMAS Graduate School of the University of Jyväskylä, Finland. We would like to thank the UCI machine learning repository for the data sets, and the MLC++ library for the source code used in this study.

References

- [Aha and Bankert, 1995] David W. Aha and Richard L. Bankert. A comparative evaluation of sequential feature selection algorithms. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, pages 1-7, 1995.
- [Bauer and Kohavi, 1999] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: bagging, boosting, and variants. *Machine Learning* 36 (1,2): 105-139, 1999.
- [Blake et al., 1999] Catherine L. Blake, Eamonn Keogh, and Chris J. Merz. *UCI repository of machine learning databases*. Dept. of Information and Computer Science, University of California, Irvine, CA, 1999.
- [Brodley and Lane, 1996] Carla E. Brodley and Terran Lane. Creating and exploiting coverage and diversity. In *Proceedings of the Workshop on Integrating Multiple Learned Models*, pages 8-14, 1996.
- [Cohen, 1960] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, pages 37-46, 1960.
- [Cunningham and Carney, 2000] Pádraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In *Proceedings of the 11th European Conference On Machine Learning*, pages 109-116, Barcelona, Spain, 2000, Springer.
- [Dietterich, 1997] Tom G. Dietterich. Machine learning research: four current directions. *AI Magazine* 18(4): 97-136, 1997.
- [Domingos and Pazzani, 1997] Pedro Domingos and Michael Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning* 29(2,3): 103-130, 1997.
- [Ho, 1998] Tin K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(8): 832-844, 1998.
- [Kohavi et al., 1996] Ron Kohavi, Dan Sommerfield, and James Dougherty. Data mining using MLC++: a machine learning library in C++. In *Proceedings of the 18th International Conference on Tools with Artificial Intelligence*, pages 234-245, 1996, IEEE CS Press.
- [Kuncheva, 1993] Ludmila I. Kuncheva. Genetic algorithm for feature selection for parallel classifiers, *Information Processing Letters* 46: 163-168, 1993.
- [Kuncheva and Jain, 2000] Ludmila I. Kuncheva and Lakshmi C. Jain. Designing classifier fusion systems by genetic algorithms, *IEEE Transactions on Evolutionary Computation* 4(4): 327-336, 2000.
- [Oliveira et al., 2003] Luiz S. Oliveira, Robert Sabourin, Flavio Bortolozzi, and Ching Y. Suen. A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition, *Pattern Recognition and Artificial Intelligence* 17(6): 903-930, 2003.
- [Opitz, 1999] David Opitz. Feature selection for ensembles. In *Proceedings of the 16th National Conference on Artificial Intelligence*, pages 379-384, 1999, AAAI Press.
- [Schaffer, 1993] Cullen Schaffer. Selecting a classification method by cross-validation. *Machine Learning* 13: 135-143, 1993.
- [Skalak, 1996] David B. Skalak. The sources of increased accuracy for two proposed boosting algorithms. In *Proceedings of the Workshop on Integrating Multiple Learned Models*, pages 120-125, 1996, AAAI.
- [Skurichina and Duin, 2001] Marina Skurichina and Robert P.W. Duin, Bagging and the random subspace method for redundant feature spaces. In *Proceedings of the 2nd International Workshop on Multiple Classifier Systems*, pages 1-10, Cambridge, UK, 2001.
- [Street and Kim, 2001] William N. Street and Yong Kim, A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pages 377-382, 2001, ACM Press.
- [Tsymbal and Puuronen, 2000] Alexey Tsymbal, Seppo Puuronen, Bagging and boosting with dynamic integration of classifiers. In *Proceedings of the International Conference on Principles of Data Mining and Knowledge Discovery*, pages 116-125, 2000, Springer-Verlag.
- [Tsymbal et al., 2003] Tsymbal Alexey, Seppo Puuronen, and David Patterson, Ensemble feature selection with the simple Bayesian classification, *Information Fusion* 4(2): 87-100, 2003.
- [Tsymbal et al., 2005] Alexey Tsymbal, Mykola Pechenizkiy, and Pádraig Cunningham, Diversity in search strategies for ensemble feature selection, *Information Fusion* 6(1): 83-98, Special Issue "Diversity in Multiple Classifier Systems", 2005.

Stereotype Extraction with Default Clustering

Julien Velcin and Jean-Gabriel Ganascia

LIP6, Université Paris VI
8 rue du Capitaine Scott
75015 PARIS, FRANCE

Abstract

The concept of stereotype seems to be really adapted when wishing to extract meaningful descriptions from data, especially when there is a high rate of missing values. This paper proposes a logical framework called *default clustering* based on default reasoning and local search techniques. The first experiment deals with the rediscovering of initial descriptions from artificial data sets, the second one extracts stereotypes of politicians in a real case generated from newspaper articles. It is shown that default clustering is more adapted in this context than the three classical clusterers considered.

Introduction

Conceptual clustering [Michalski, 1980] is a fundamental machine learning task that is applied in various areas such as image analysis, analytical chemistry, biology, sociology. It takes a set of object descriptions as input and creates a classification scheme. The conceptual descriptions of clusters are of particular interest to reason about the categories themselves, to compare different data sets and to predict new observations. This work focuses on the extraction of such conceptual descriptions, especially in the specific context of missing data.

Automatic inductive techniques have to deal with missing information, due to voluntary omissions, human error, broken equipment [Newgard and Lewis, 2002]. In the context of sparse data, i.e. with a huge amount of missing values, the concept of stereotype seems more appropriate than the usual one of prototype to describe data clusters. Therefore, our goal is to extract stereotype sets that represent the data sets as well as possible. By analogy to default logic [Reiter, 1980], which is a specific logic for default deduction, we make use of *default subsumption*, which is a specific logic for default induction, to build such stereotypes.

Section 1 presents a new approach to conceptual clustering when missing information exists. Section 2 proposes a general framework in the attribute-value formalism. The new notion of default subsumption is introduced, before seeing how the concept of stereotype makes it possible to name clusters. A stereotype set extraction algorithm based on local search techniques is then presented. Section 3 concerns experiments,

first on artificial data sets and secondly with a real data case generated from newspaper articles.

1 Conceptual clustering with sparse data

1.1 Dealing with missing values

This paper proposes a clustering method that deals with high rates of missing values. But contrary to algorithms such as k-modes (categorical version of k-means) or EM, that can easily lead to local optima, we have chosen to achieve the clustering using a combinatorial optimization approach, like in [Figuerola *et al.*, 2003] or [Sarkar and Leong, 2001]. Note that our goal is not only to cluster examples but also and mainly to describe the cluster in a way that is simple and easy to understand. The problem can thus be stated as finding readable, understandable, consistent and rich descriptions from the data.

1.2 Overview of default logic

During the eighties, there were many attempts to model deductive reasoning when missing information exists. A lot of formalisms were developed to encompass the inherent difficulties of such models, especially their non-monotony: close-world assumption, circumscription, default logic, etc. Since our goal is to deal with missing values, it seems natural to take advantage of this work. The default logic formalism, introduced by R. Reiter in 1980, was chosen because it seemed to correspond well to our problem.

This logic for default reasoning is based on the notion of default rule, through which it is possible to infer new formulas when the hypotheses are not inconsistent with the current context. More generally, a default rule always has the following form: $A : B_1, B_2 \dots B_n / C$ where A is called the prerequisite, B_i the justifications and C the conclusion. This default rule can be interpreted as follows: if A is known to be true and if it is consistent to assume $B_1, B_2 \dots B_n$ then conclude C . For instance, let us consider the default rule below related to the experiments of the last section:

$$\frac{\text{politician}(X) \wedge \text{introducedAbroad}(X) : \neg \text{diplomat}(X)}{\text{traitor}(X)}$$

This rule translates an usual way of reasoning for many people living in France at the end of the 19th century. It states that the conclusion $\text{traitor}(X)$ can be derived if X is a politician who is known to be introduced abroad and that we cannot prove that he is a diplomat.

The key idea here is to use similar observations and their descriptions to infer new information instead of default rules, but the underlying mechanism is the same. The following subsection explains the transition from default logic to default induction.

1.3 Default clustering

E. Rosch saw the categorization itself as one of the most important issues in cognitive science [Rosch, 1975]. She introduced the concept of prototype as the ideal member of a category. Whereas categorization makes similar observations fit together and dissimilar observations be well separated, clustering is the induction process in data mining that actually build such categories. More specifically, conceptual clustering is a machine learning task defined by R. Michalski [Michalski, 1980] which does not require a teacher and uses an evaluation function to discover classes that have appropriate conceptual descriptions. Conceptual clustering was principally studied in a probabilistic context (see, for instance, D. Fisher's Cobweb algorithm [Fisher, 1987]) and rarely on really sparse data sets. For instance, the experiments done by P.H. Gennari do not exceed 30% of missing values [Gennari, 1990].

As seen above, default logic is a logic for deduction depending on background knowledge. This paper proposes a new technique called *default clustering* which uses a similar principle but for induction when missing information exists. The main assumption is the following: if an observation is grouped with other similar observations, you can use these observations to complete unknown information in the original fact if it remains consistent with the current context. Whereas default logic needs implicit knowledge expressed by default rules, default clustering only uses information available in the data set. The next section presents this new framework. It shows how to extract stereotype sets from very sparse data: first it extends classical subsumption (see section 2.1), next it discusses stereotype choice (see section 2.2), and finally it proposes a local search strategy to find the best solution (see section 2.4).

2 Logical framework

This section presents the logical framework of default clustering in the attribute-value formalism (an adaptation to conceptual graphs can be found in [Ganascia and Velcin, 2004]). The description space is noted $\mathcal{D}$, the descriptor space (i.e. the values the attributes can take) $\mathcal{V}$ and the example set E . The function δ maps each example $e \in E$ to its description $\delta(e) \in \mathcal{D}$.

2.1 Default subsumption

Contrary to default logic, the problem here is not to deduce, but to induce knowledge from data sets in which most of the information is unknown. Therefore, we put forward the notion of *default subsumption*, which is the equivalent for subsumption of the default rule for deduction. Saying that a description $d \in \mathcal{D}$ subsumes $d' \in \mathcal{D}$ by default means that there exists an implicit description d'' such that d' completed with d'' , i.e. $d' \wedge d''$, is more specific than d in the classical sense,

which signifies that $d' \wedge d''$ entails d . The exact definition follows:

Definition 1 d subsumes d' by default (noted $d \leq_D d'$) iff $\exists d_c$ such that $d_c \neq \perp$ and $d \leq d_c$ and $d' \leq d_c$ where $t \leq t'$ stands for t subsumes t' in the classical sense. d_c is a minorant of d and d' in the subsumption lattice.

To illustrate our definition, here are some descriptions based on binary attributes that can be compared with respect to the default subsumption:

$d_1 = \{(Traitor = yes), (Internationalist = yes)\}$

$d_2 = \{(Traitor = yes), (Connection_with_jews = yes)\}$

$d_3 = \{(Patriot = yes)\}$

$d_1 \leq_D d_2$ and $d_2 \leq_D d_1$ because $\exists d_c$ such that $d_1 \leq d_c$ and $d_2 \leq d_c$: $d_c = \{(Traitor = yes), (Internationalist = yes), (Connection_with_jews = yes)\}$.

However, considering that a patriot cannot be an internationalist and vice-versa, i.e. $\neg((Patriot=yes) \wedge (Internationalist=yes))$, which was an implicit statement for many people living in France at the end of the 19th century, d_1 does not subsume d_3 by default, i.e. $\neg(d_1 \leq_D d_3)$.

Property 1 The notion of default subsumption is more general than classical subsumption since, if d subsumes d' , i.e. $d \leq d'$, then d subsumes d' by default, i.e. $d \leq_D d'$. The converse is not true.

Property 2 The default subsumption relationship is symmetrical, i.e. $\forall d \forall d'$ if $d \leq_D d'$ then $d' \leq_D d$.

Note that the notion of default subsumption may appear strange for people accustomed to classical subsumption because of the symmetrical relationship. As a consequence, it does not define an ordering relationship on the description space $\mathcal{D}$. The notation $\leq_D$ may be confusing with respect to this symmetry, but it is relative to the underlying idea of generality.

2.2 Concept of stereotype

In the literature of categorization, Rosch introduced the concept of prototype [Rosch, 1975; 1978] inspired by the family resemblance notion of Wittgenstein [Wittgenstein, 1953] (see [Shawver, 2004] for an electronic version and [Narboux, 2001] for an analysis focused on family resemblance). Even if our approach and the original idea behind the concept of prototype have several features in common, we prefer to refer to the older concept of stereotype that was introduced by the publicist W. Lippman [Lippman, 1922]. For him, stereotypes are perceptive schemas (a structured association of characteristic features) shared by a group about other person or object categories. These simplifying and generalizing images about reality affect human behavior and are very subjective. Below are three main reasons to make such a choice.

First of all, the concept of prototype is often misused in data mining techniques. It is reduced to either an average observation of the examples or an artificial description built on the most frequent shared features. Nevertheless, both of them are far from the underlying idea in family resemblance. Especially in the context of sparse data, it seems more correct to speak about a combination of features found in different example descriptions than about average or mode selection. The second argument is that the notion of stereotype is often defined as an imaginary picture that distorts the reality. Our goal

is precisely to generate such pictures even if they are caricatural of the observations. Finally, these specific descriptions are better adapted for fast classification (we can even say discrimination) and prediction than prototypes. This last feature is closely linked to Lippman's definition.

In order to avoid ambiguities, we restrict the notion of stereotype to a specific description $d \in \mathcal{D}$ associated to (we can say "covering") a set of descriptions $D \subset \mathcal{D}$. However, the following subsection does not deal just with stereotypes but with stereotype sets to cover a whole description set. The objective is therefore to automatically construct stereotype sets, whereas most of the studies focus on already fixed stereotype usage [Rich, 1979; Amossy and Herschberg Pierrot, 1997]. Keeping this in mind, the space of all the possible stereotype sets is browsed in order to discover the best one, i.e. the set that best covers the examples of E with respect to some similarity measure. But just before addressing the search itself, we should consider both the relation of relative cover and the similarity measure used to build the categorization from stereotype sets.

2.3 Stereotype sets and relative cover

Given an example e characterized by its description $d = \delta(e) \in \mathcal{D}$, consider the following statement: the stereotype $s \in \mathcal{D}$ can cover e if and only if s subsumes d by default. It means that in the context of missing data each piece of information is so crucial that even a single contradiction prevents the stereotype from being a correct generalization. Furthermore, since there is no contradiction between this example and its related stereotype, the stereotype may be used to complete the example description.

In order to perform the clustering, a very general similarity measure M_{sim} has been defined, which counts the number of common descriptors of $\mathcal{V}$ belonging to two descriptions, ignores the unknown values and takes into account the default subsumption relationship:

$$\begin{aligned} M_{sim}: \mathcal{D} \times \mathcal{D} &\rightarrow \mathbb{N}^+ \\ (d_i, d_j) &\mapsto M_{sim}(d_i, d_j) = |\{v \in \mathcal{V} / d = d_i \wedge d_j\}| \\ &\quad \text{if } d_i \leq_D d_j \text{ and} \\ &\quad M_{sim}(d_i, d_j) = 0 \text{ if } \neg(d_i \leq_D d_j), \end{aligned}$$

where $d_i \wedge d_j$ is the least minorant of d_i and d_j in the subsumption lattice.

Let us now consider a set $S = \{s_0, s_1, s_2 \dots s_n\} \subset \mathcal{D}$ of stereotypes. s_0 is the absurd-stereotype linked to the set E_0 . Then, a categorization of E can be calculated using S with an affectation function that we called *relative cover*:

Definition 2 *The relative cover of an example $e \in E$, with respect to a set of stereotypes $S = \{s_0, s_1, s_2 \dots s_n\}$, noted $C_S(e)$, is the stereotype s_i if and only if:*

- $s_i \in S$,
- $M_S(\delta(e), s_i) > 0$,
- $\forall k \in [1, n], k \neq i, M_{sim}(\delta(e), s_i) > M_{sim}(\delta(e), s_k)$.

It means that an example $e \in E$ is associated to the most similar and "covering-able" stereotype *relative* to the set S . If there are two competitive stereotypes with an equal higher score or if there is no covering stereotype, then the example is associated to the absurd-stereotype s_0 . In this case, no completion can be calculated for e .

2.4 Stereotype extraction

In this paper, default reasoning is formalized using the notions of both default subsumption and stereotype set. Up to now, these stereotype sets were supposed to be given. This section shows how the classification can be organized into such sets in a non-supervised learning task. It can be summarized as follows. Given:

1. An example set E .
2. A description space $\mathcal{D}$.
3. A description function $\delta: E \rightarrow \mathcal{D}$ which associates a description $\delta(e) \in \mathcal{D}$ to each example belonging to the training set E .

The function of a non-supervised learning algorithm is to organize the initial set of individuals E into a structure (for instance a hierarchy, a lattice or a pyramid). In the present case, the structure is limited to partitions of the training set, which corresponds to searching for stereotype sets as discussed above. These partitions may be generated by $(n + 1)$ stereotypes $S = \{s_0, s_1, s_2 \dots s_n\}$: it is sufficient to associate to each s_i the set E_i of examples e belonging to E and covered by s_i relative to S . The examples that cannot be covered by any stereotype are put into the E_0 cluster and associated to s_0 .

To choose from among the numerous possible partitions, which is a combinatorial problem, a non-supervised algorithm requires a function for evaluating stereotype set relevance. Because of the categorical nature of data and the previous definition of relative cover, it appears natural to make use of the similarity measure M_{sim} . This is exactly what we do by introducing the following evaluation function h_E :

Definition 3 *E being an example set, $S = \{s_0, s_1, s_2 \dots s_n\}$ a stereotype set and C_S the function that associates to each example e its relative cover, i.e. its closest stereotype with respect to M_{sim} and S , the evaluation function h_E is defined as follows:*

$$h_E(S) = \sum_{e \in E} M_{sim}(\delta(e), C_S(e))$$

While k-modes and EM algorithms are straightforward, i.e. each step leads to the next one until convergence, we reduce here the non-supervised learning task to an optimization problem. This approach offers several interesting features: avoiding local optima (especially with categorical and sparse data), providing "good" solutions even if not the best ones, better control of the search. In addition, it is not necessary to specify the number of expected stereotypes that is also discovered during the search process.

There are several methods for exploring such a search space (hill-climbing, simulated annealing, etc.). We have chosen the meta-heuristic called *tabu search* which improves the local search algorithm. Remember that the local search process can be schematized as follows: 1. An initial solution S_{ini} is given (for instance at random). 2. A neighborhood $\mathcal{P}$ is calculated from the current solution S_i with the assistance of permitted movements. These movements can be of low influence (enrich one stereotype with a descriptor, remove a descriptor from another) or of high influence (add or retract one stereotype to or from the current stereotype set). 3. The

```

 $S_c \leftarrow \{s_0\}$  ; the current solution
 $S_b \leftarrow S_c$  ; the best up-to-now solution
 $T \leftarrow \emptyset$  ; the list that contains tabu-attributes
for  $i \leftarrow 1$  to  $NStep$  do {
  let  $S_c$  be  $\{s_0, s_1, s_2 \dots s_k\}$ 
   $\mathcal{P} \leftarrow \emptyset$  ; initialize the neighborhood
  for all  $A_i \notin T$  do {
    for  $m \leftarrow 1$  to  $k$  do {
      for all  $v \leftarrow (A_i = V_{ij})$  do
        if  $v$  does not belong to a stereotype of  $S_c$  {
           $S \leftarrow \{s_1 \dots s_{m-1}, s_m \cup v, s_{m+1} \dots s_k\}$ 
           $\mathcal{P} \leftarrow \mathcal{P} \cup S$ 
        }
        if  $v$  belongs to a stereotype of  $S_c$  {
           $s' \leftarrow s_m \setminus v$ 
          if  $s' \neq \emptyset$  then  $S \leftarrow \{s_1 \dots s_{m-1}, s', s_{m+1} \dots s_k\}$ 
          else  $S \leftarrow \{s_1 \dots s_{m-1}, s_{m+1} \dots s_k\}$ 
          if  $S \neq \emptyset$  then  $\mathcal{P} \leftarrow \mathcal{P} \cup S$ 
        }
      }
    }
    for all  $v \leftarrow (A_i = V_{ij})$ 
    if  $v$  does not belong to a stereotype of  $S_c$  do {
       $s_n \leftarrow \{v\}$ 
       $S \leftarrow \{s_1, s_2 \dots s_m, s_n\}$ 
       $\mathcal{P} \leftarrow \mathcal{P} \cup S$ 
    }
  }
   $S_N \leftarrow \operatorname{argmax}_{S \in \mathcal{P}} (h_E(S))$ 
   $T$  is updated, depending on the chosen attribute  $A_i$ 
  that permits to pass from  $S_c$  to  $S_N$ .
   $S_c \leftarrow S_N$ 
  if  $h_E(S_c) > h_E(S_b)$  then  $S_b \leftarrow S_c$ 
}
return  $S_b$ 

```

Figure 1: The default clustering algorithm

best movement, relative to the evaluation function h_E , is chosen and the new current solution S_{i+1} is computed. 4. The process is iterated a specific number of times $NStep$ and the best up-to-now discovered solution is recorded. Then, the solution is the stereotype set S_b that best maximizes h_E in comparison to all the crossed sets.

As in almost all local search techniques, there is a trade-off between exploitation, i.e. choosing the best movement, and exploration, i.e. choosing a non optimal state to reach completely different areas. The tabu search extends the basic local search by manipulating short and long-term memories which are used to avoid loops and to intelligently explore the search space. This meta-heuristic is detailed in [Glover and Laguna, 1997] and its application to clustering can be found in [Al-Sultan, 1995]. Note that only the short-term memory was used at this stage of our work.

2.5 Default clustering algorithm

Fig. 1 is the main frame of the default clustering algorithm. It is based on a very basic version of tabu search that tries to maximize our function h_E . A_i denotes the i th attribute and E is the example set. S_c and S_b stands respectively for the current and for the best solution. $NStep$ is the maximal number of iterations, $\mathcal{P}$ the current neighborhood and T the tabu-list that contains tabu-attributes. If an attribute A_i is in the tabu-list then no descriptor ($A_i = V_{ij}$) can be used to calculate the neighborhood of the current solution.

A “no-redundancy” constraint has been added in order to obtain a perfect separation between the stereotypes. In the context of sparseness, it seems really important to extract contrasted descriptions which are used to quickly classify the examples, as does the concept of stereotype introduced by Lippman.

A new constraint called *cognitive cohesion* is now defined. It verifies cohesion within a cluster, i.e. an example set $E_j \subset E$, relative to the corresponding stereotype $s_j \in S$. Cognitive cohesion is verified if and only if, given two descriptors v_1 and $v_2 \in \mathcal{V}$ of s_j , it is always possible to find a series of examples that makes it possible to pass by correlation from v_1 to v_2 . Below are two example sets with their covering stereotype. The example on the left verifies the constraint, the one on the right does not.

$s_1 : a_0, b_1, d_5, f_0, h_0$	$s_2 : a_0, b_1, d_5, f_0, h_0$
$e_1 : a_0, ?, ?, ?, h_0$	$e_0 : a_0, b_1, ?, ?, ?$
$e_2 : a_0, b_1, ?, ?, ?$	$e_8 : ?, ?, ?, f_0, ?$
$e_6 : ?, ?, d_5, ?, ?$	$e_9 : a_0, b_1, ?, ?, ?$
$e_8 : ?, b_1, d_5, f_0, ?$	$e_{51} : ?, ?, d_5, ?, h_0$
$e_{42} : a_0, ?, d_5, ?, ?$	$e_{98} : ?, ?, d_5, ?, h_0$

Hence, with s_2 it is never possible to pass from a_0 to d_5 , whereas it is allowed by s_1 (you begin with e_2 to go from a_0 to b_1 , and then you use e_8 to go from b_1 to d_5). It means that, in the case of s_1 , you are always able to find a “correlation path” from one descriptor of the description to another, i.e. examples explaining the relationship between the descriptors in the stereotype.

3 Experiments

This section presents experiments performed on artificial data sets. This is followed by an original comparison in a real data case using three well-known clusterers. Default clustering was implemented in a Java program called PRESS (*Programme de Reconstruction d’Ensembles de Stéréotypes Structurés*). All the experiments for k-modes, EM and Cobweb were performed using the Weka platform [Garner, 1995].

3.1 Validation on artificial data sets

These experiments use artificial data sets to validate the robustness of our algorithm. The first step is to give some contrasted descriptions of $\mathcal{D}$. Let us note n_s the number of these descriptions. Next, these initial descriptions are duplicated n_d times. Finally, missing data are artificially simulated by removing a percentage p of descriptors at random from these $n_s \times n_d$ artificial examples. The evaluation is carried out by testing different clusterers on these data and comparing the discovered cluster representatives with the initial descriptions. We verify what we call *recovered descriptors*, i.e. the proportion of initial descriptors that are found. This paper presents the results obtained with $n_s = 5$ and $n_d = 50$ over 50 runs. The number of examples is 250 and the descriptions are built using a language with 30 binary attributes. The tabu-list length is equal to 10 and $NStep$ to 100. Note that the first group of experiments are placed in the Missing Completely At Random (MCAR) framework.

Fig. 2 shows firstly that the results of PRESS are very good with a robust learning process. The stereotypes discovered correspond very well to the original descriptions up to 75% of missing descriptors. In addition, this score remains good (nearly 50%) up to 90%. Whereas Cobweb seems stable relative to the increase in the number of missing values, the results of EM rapidly get worse above 80%. Those obtained using k-modes are the worst, although the number of expected classes has to be specified.

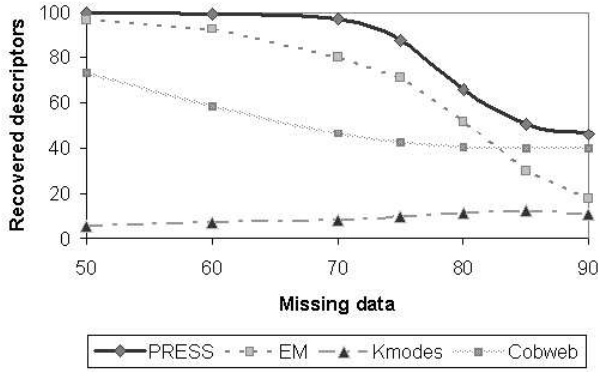


Figure 2: Proportion of recovered descriptors.

3.2 Studying social misrepresentation

The second part of the experiments deals with real data extracted from a newspaper called “Le Matin” at the end of the 19th century in France. The purpose is to automatically discover stereotype sets from events related to the political disorder in the first ten days of September 1893. The results of PRESS are compared to those of the three clusterers k-modes, EM and Cobweb. It should be pointed out that our interest focuses on the cluster descriptions, which we call *representatives* to avoid any ambiguity, rather than on the clusters themselves.

The articles linked to the chosen theme were gathered and represented using a language with 33 attributes. The terms of this language, i.e. attributes and associated values, were extracted manually. Most of the attributes are binary, 4 accept more than two values and 4 are ordinals. The number of extracted examples is 63 and the rate of missing data is nearly 87%, which is most unusual.

3.3 Evaluation of default clustering

In order to evaluate PRESS, a comparison was made with three classical clusterers: k-modes, EM and Cobweb. Hence, a non-probabilistic description of the clusters built by these algorithms was extracted using four techniques: (1) using the most frequent descriptors (mode approach); (2) the same as (1) but forbidding contradictory features between the examples and their representative; (3) dividing the descriptors between the different representatives; (4) the same as (3) but forbidding contradictory features. Two remarks need to be made. Firstly, the cluster descriptions resulting from k-modes correspond to technique (1). Nevertheless, we tried the other three techniques exhaustively. Secondly, representatives resulting from extraction techniques (3) and (4) entail *by construction* a redundancy rate of 0%. The comparison was made according to the following three points:

The first approach considers the contradictions between an example and its representative. The *example contradiction* is the percentage of examples containing at least one descriptor in contradiction with its covering representative. In addition, if you consider one of these contradictory examples, *average contradiction* is the percentage of descriptors in contradiction

	k-Modes				EM	
	(1)	(2)	(3)	(4)	(1)	(2)
n	6	6	6	6	2	2
ex. contradiction	27	0	27	0	48	0
av. contradiction	42	0	44	0	56	0
h_E	.89	.60	.74	.50	.85	.66
redundancy	70	63	0	0	17	7
cog. cohesion	×	×	×	×	×	×

	EM		Cobweb				PRESS
	(3)	(4)	(1)	(2)	(3)	(4)	
n	2	2	2	2	2	2	6
ex. cont.	48	0	56	0	57	0	0
av. cont.	56	0	52	0	51	0	0
h_E	.83	.65	.82	.56	.68	.46	.79
red.	0	0	72	55	0	0	0
cog. coh.	×	×	×	×	×	×	✓

Figure 3: Comparative results on *Le Matin*.

with its representative. This facet of conceptual clustering is very important, especially in the sparse data context.

Secondly, we check if the cognitive cohesion constraint (see 2.5) is verified. The rate of descriptor redundancy is also considered. These two notions are linked to the concept of stereotype and to the sparse data context.

Finally, we consider the degree of similarity between the examples and their covering representatives. This corresponds to the notion of compactness within clusters, but without penalizing the stereotypes with many descriptors. The function h_E seems really adapted to give an account of representative relevance. In fact, we used a version of h_E normalized between 0 and 1, by dividing by the total number of descriptors.

3.4 Results

Fig. 3 gives the results obtained from the articles published in *Le Matin*. Experiments for the k-modes algorithm were carried out with $N = 2 \dots 8$ clusters, but only $N = 6$ results are presented in this comparison. The rows of the table show the number n of extracted representatives, the two scores concerning contradiction, the result of h_E , the redundancy score and whether or not the cognitive cohesion constraint is verified. The columns represent each type of experiment (k-modes associated with techniques (1) to (4), EM and Cobweb as well, and finally our algorithm PRESS).

Let us begin by considering the contradiction scores. They highlight a principal result of default clustering: using PRESS, the percentage of examples having contradictory features with their representative is always equal to 0%. In contrast, the descriptions built using techniques (1) and (3) (whatever the clusterer used) possess at least one contradictory descriptor with 27% to 57% of the examples belonging to the cluster. Furthermore, around 50% of the descriptors of these examples are in contradiction with the covering description, and that can in no way be considered as a negligible noise. This is the reason why processes (1) and (3) must be avoided, especially in the sparse data context, when building such representatives from k-modes, EM or Cobweb clustering. Hence, we only consider techniques (2) and (4) in the following experiments.

Let us now study the results concerning clustering quality. This quality can be expressed thanks to the compactness

function h_E , the redundancy rate and cognitive cohesion.

PRESS marked the best score (0.79) for cluster compactness with six stereotypes. That means a very good homogeneity between the stereotypes and the examples covered. It is perfectly consistent since our algorithm tries to maximize this function. The redundant descriptors rate is equal to 0%, according to the no-redundancy constraint. Furthermore, PRESS is the only algorithm that is able to verify cognitive cohesion. EM obtains the second best score and redundant descriptor rate remains acceptable. However, the number of expected classes must be given or guessed using a cross-validation technique, for instance. K-modes and Cobweb come third and fourth and also have to use an external mechanism to discover the final number of clusters.

Note that the stereotypes extracted using PRESS correspond to the political leanings of the newspaper. For instance, the main stereotype produces a radical, socialist politician, corrupted by foreign money and Freemasonry, etc. It corresponds partly to the difficulty in accepting the major changes proposed by the radical party and to the fear caused in France since 1880 by the theories of Karl Marx. We cannot explain here in more detail the semantics of discovered stereotypes, but these first results are really promising.

4 Conclusion

Conceptual clustering is seldom studied with such a high number of missing values. However, it is really important to be able to extract readable, understandable descriptions from such type of data in order to complete information, to classify new observations quickly and to make predictions. In this way, default clustering presented in this paper tries to provide an alternative to the usual clusterers. Moreover, based on local optimization techniques, it proposes a very general easy-to-extend framework for stereotype set discovering: new movements, constraints added relative to the problematic chosen, adapted control indexes, etc. The results obtained, on both artificial data sets and a real case extracted from newspaper articles, are really promising and should lead to other historical studies concerning social stereotypes.

For instance, we are currently applying these techniques to the study of *social representations*, a branch of social psychology introduced by S. Moscovici [Moscovici, 1961]. More precisely, this approach is really useful for *press content study* which up to now is done manually by experts. Hence, future work could be done on choosing key dates of the Dreyfus affair and automatically extracting stereotypical characters from different newspapers. These results will then be compared and contrasted with the work of sociologists and historians of this period.

References

- [Al-Sultan, 1995] K. Al-Sultan. A tabu search approach to the clustering problem. *Pattern Recognition*, 28(9):pp.1443–1451, 1995.
- [Amossy and Herschberg Pierrot, 1997] R. Amossy and A. Herschberg Pierrot. *Stéréotypes et clichés: langues, discours, société*. Nathan Université, 1997.
- [Figuerola et al., 2003] A. Figuerola, J. Borneman, and T. Jiang. Clustering binary fingerprint vectors with missing values for DNA array data analysis. *IEEE Computer Society Bioinformatics Conference*, 2003.
- [Fisher, 1987] D.H. Fisher. Knowledge Acquisition Via Incremental Conceptual Clustering. *Machine Learning*, (2):pp.139–172, 1987.
- [Ganascia and Velcin, 2004] J.-G. Ganascia and J. Velcin. Clustering of conceptual graphs with sparse data. *Proceedings of the 12th International Conference on Conceptual Structures*, 2004.
- [Garner, 1995] S.R. Garner. Weka: The waikato environment for knowledge analysis. *Proc. of the New Zealand Computer Science Research Students Conference*, pages pp.57–64, 1995.
- [Gennari, 1990] J.H. Gennari. An experimental study of concept formation. 1990. Doctoral dissertation.
- [Glover and Laguna, 1997] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
- [Lippman, 1922] W. Lippman. *Public Opinion*. Ed. MacMillan, NYC, 1922.
- [Michalski, 1980] R.S. Michalski. Knowledge acquisition through conceptual clustering : A theoretical framework and algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems*, (4):pp.219–243, 1980.
- [Moscovici, 1961] S. Moscovici. *La psychanalyse : son image et son public*. PUF, Paris, 1961.
- [Narboux, 2001] J.-P. Narboux. Ressemblances de famille, caractères, critères. pages pp.69–95. PUF, 2001.
- [Newgard and Lewis, 2002] C.D. Newgard and R.J. Lewis. The Imputation of Missing Values in Complex Sampling Databases: An Innovative Approach. *Academic Emergency Medicine*, 9(5484), 2002.
- [Reiter, 1980] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, (13):pp.81–132, 1980.
- [Rich, 1979] E. Rich. User Modeling via Stereotypes. *International Journal of Cognitive Science*, 3:pp.329–354, 1979.
- [Rosch, 1975] E. Rosch. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, (104):pp.192–232, 1975.
- [Rosch, 1978] E. Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. NJ: Lawrence Erlbaum, Hillsdale, 1978.
- [Sarkar and Leong, 2001] M. Sarkar and T.Y. Leong. Fuzzy k-means clustering with missing values. *Proc AMIA Symp.*, pages pp.588–92, 2001.
- [Shawver, 2004] L. Shawver. Commentary on Wittgenstein's Philosophical Investigations, 2004. <http://users.rcn.com/rathbone/lw65-69c.htm>.
- [Wittgenstein, 1953] L. Wittgenstein. *Philosophical Investigations*. Blackwell, Oxford, UK, 1953.

Learning Subjective Representations for Planning

Dana Wilkinson

School of Computer Science
University of Waterloo
Waterloo, ON, Canada
d3wilkinson@uwaterloo.ca

Michael Bowling

Department of Computing Science
University of Alberta
Edmonton, AB, Canada
bowling@cs.ualberta.ca

Ali Ghodsi

School of Computer Science
University of Waterloo
Waterloo, ON, Canada
aghodsib@uwaterloo.ca

Abstract

Planning involves using a model of an agent's actions to find a sequence of decisions which achieve a desired goal. It is usually assumed that the models are given, and such models often require expert knowledge of the domain. This paper explores subjective representations for planning that are learned directly from agent observations and actions (requiring no initial domain knowledge). A non-linear embedding technique called Action Respecting Embedding is used to construct such a representation. It is then shown how to extract the effects of the agent's actions as operators in this learned representation. Finally, the learned representation and operators are combined with search to find sequences of actions that achieve given goals. The efficacy of this technique is demonstrated in a challenging robot-vision-inspired image domain.

1 Introduction

Planning, at its essence, involves searching an appropriately defined state space. This state space is a consequence of the agent's model of the effects of its actions (*e.g.*, STRIPS [Fikes and Nilsson, 1971], Markov Decision Processes [Puterman, 1994]). It is assumed that these models can be defined from domain experts and planning uses these models to find sequences of actions to achieve given goals. The problem is that models are not always known or easily built for a domain of interest. Others have studied methods for learning or augmenting models, such as STRIPS operators [Wang, 1995] or MDP transitions [Peng and Williams, 1993]. All of these techniques still require expert intuition about the domain to provide, at the least, an appropriate state representation.

This paper focuses on learning an appropriate representation for planning, using only an agent's observations and actions. We call this a *subjective representation* as the learned representation is extracted based only on the agent's experience, and requires no expert knowledge of the domain. The approach solves two important problems: (i) learning an appropriate state-space representation, and (ii) learning the effects of the agent's actions in this representation.¹ The re-

¹The approach of learning a subjective representation more

quired input is a sequence of actions and observations from the agent's subjective experience, but no semantic meaning (domain-specific or otherwise) is required. This input is used in a three step process.

First, Section 2 reviews Action Respecting Embedding (ARE). ARE is a technique for dimensionality reduction that specifically makes use of a temporal sequence of observations and actions. ARE learns manifolds that capture the important underlying dynamics of the high-dimensional data in much fewer dimensions (addressing the first problem). Next, Section 3 describes a method for learning operators for each action that can be applied to any point in the learned representation (addressing the second problem). Examples of learned operators are provided along with a discussion of how the semantic meaning of the associated actions can sometimes be extracted. Finally, Section 4 shows the results of planning in the resulting representation using the learned operators. Both the resulting plan in the learned representation, and the result of applying the plans in the original high-dimensional domain are compared.

2 Action Respecting Embedding

High-dimensional data sets, such as sequences of images, can often be characterized by a low-dimensional representation that is related to the process generating the data. For example, a low-dimensional representation for image data may correspond to the degrees of freedom of a platform moving a camera. Such a representation is ideal for planning as it directly captures the actions' effects on the world. The goal here is to take a temporal sequence of data points, $z_1, \dots, z_n$, and associated actions, $a_1, \dots, a_{n-1}$, and find a low-dimensional representation for z_i that is appropriate for planning.

Recently, *nonlinear manifold learning* techniques have been used to map a high-dimensional dataset into a smaller dimensional space. Semidefinite Embedding (SDE) [Weinberger and Saul, 2004] is one such technique. SDE learns a kernel matrix, which represents a non-linear projection of

closely resembles recent work on learning predictive representations [James and Singh, 2004; Rosencrantz *et al.*, 2004; Jaeger, 2000] than previous work on augmenting operators or transition probabilities. This approach, though, is specifically designed for very high-dimensional observation spaces as it implicitly involves a dimensionality reduction component.

Algorithm: SDE($\|\cdot\|, (z_1, \dots, z_n)$)

Construct neighbors, N , using k -NN with $\|\cdot\|$.

Maximize $\text{Tr}(K)$ subject to $K \succeq 0$, $\sum_{ij} K_{ij} = 0$, and
 $\forall ij \quad N_{ij} > 0 \vee [N^T N]_{ij} > 0 \Rightarrow$
 $K_{ii} - 2K_{ij} + K_{jj} = \|z_i - z_j\|^2$

Run Kernel PCA with learned kernel, K .

Table 1: Algorithm: Semidefinite Embedding (SDE).

the input data into a more linear representation. It then uses Kernel PCA [Scholkopf and Smola, 2002], a generalization of principle components analysis using feature spaces represented by kernels, to extract out a low-dimensional representation of the data. The kernel matrix K is learned in SDE by solving a semidefinite program with a simple set of constraints. The most important constraints encode the common requirement in dimensionality reduction that the non-linear embedding should preserve local distances. In other words, nearby points in the original input space should remain nearby in the resulting feature representation. Therefore SDE requires a distance metric $\|\cdot\|$ on the original input space, and uses this metric to construct a k -nearest neighbors graph. It then adds constraints into the semidefinite program to ensure that the distance between neighbors is preserved. The optimization maximizes $\text{Tr}(K)$, i.e., the variance of the learned feature representation, which should minimize its dimensionality. The SDE Algorithm is shown in Table 1.

SDE does not take into account two important pieces of knowledge about the data: the temporal ordering of the input vectors, z_i , and the action labels, a_i . Therefore, SDE doesn’t guarantee that temporally-nearby input points will be spatially nearby in the feature representation. Also, SDE won’t necessarily result in a space where actions have a simple interpretation. The recent Action Respecting Embedding (ARE) algorithm [Bowling *et al.*, 2005] extends SDE to make use of exactly this type of knowledge about the data.

Formally, ARE takes a set of D -dimensional input vectors, $z_1, \dots, z_n$ (e.g., images), in temporal order, along with associated discrete actions, $a_1, \dots, a_{n-1}$, where action a_i was executed between input z_i and input z_{i+1} . ARE then computes a set of d -dimensional output vectors $x_1, \dots, x_n$ in one-to-one correspondence with the input vectors. This provides a meaningful embedding in $d < D$ dimensions. ARE modifies SDE in two key ways. First, it exploits the knowledge that the images are given in a temporal sequence. It uses this knowledge to build an improved neighborhood graph based on each input’s distances to its temporal neighbors using the provided local distance metric². Second, it constrains the embedding to respect the action labels that are associated with adjacent pairs of observations. This ensures that the actions have a simple interpretation in the resulting feature space.

It is this second enhancement of ARE that is the critical

²We have found that ARE is fairly robust to the choice of distance metrics, and use simple Euclidean distance for all of the experiments in this paper.

Algorithm: ARE($\|\cdot\|, (z_1, \dots, z_n), (a_1, \dots, a_{n-1})$)

Construct neighbors, N , as in [Bowling *et al.*, 2005].

Maximize $\text{Tr}(K)$ subject to $K \succeq 0$, $\sum_{ij} K_{ij} = 0$,
 $\forall ij \quad N_{ij} > 0 \vee [N^T N]_{ij} > 0 \Rightarrow$
 $K_{ii} - 2K_{ij} + K_{jj} \leq \|z_i - z_j\|^2$, **and**
 $\forall ij \quad a_i = a_j \Rightarrow$
 $K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)} =$
 $K_{ii} - 2K_{ij} + K_{jj}$

Run Kernel PCA with learned kernel, K .

Table 2: Algorithm: Action Respecting Embedding (ARE).

feature for subjective planning. ARE constrains the learned manifold to be in a space where the labeled actions correspond to distance-preserving transformations—those consisting only of rotation and translation³. Therefore, for any two inputs, z_i and z_j , the same action from these inputs must preserve their distance in the learned feature space. Letting $\Phi(z_i)$ denote input z_i ’s representation in the feature space, action a ’s transformation, f_a , must satisfy:

$$\forall i, j \quad \|f_a(\Phi(z_i)) - f_a(\Phi(z_j))\| = \|\Phi(z_i) - \Phi(z_j)\|. \quad (1)$$

Now, let $a = a_i$ and consider the case where $a_j = a_i$. Then, $f_a(\Phi(z_i)) = \Phi(z_{i+1})$ and $f_a(\Phi(z_j)) = \Phi(z_{j+1})$, and Constraint 1 becomes:

$$\|\Phi(z_{i+1}) - \Phi(z_{j+1})\| = \|\Phi(z_i) - \Phi(z_j)\|. \quad (2)$$

In terms of the kernel matrix, this can be written as:

$$\forall i, j \quad a_i = a_j \Rightarrow$$

$$K_{(i+1)(i+1)} - 2K_{(i+1)(j+1)} + K_{(j+1)(j+1)} =$$

$$K_{ii} - 2K_{ij} + K_{jj} \quad (3)$$

ARE simply adds Constraint 3 into SDE’s usual constraints to arrive at the optimization and algorithm shown in Table 2.

Experiments. Here we define IMAGEBOT, a synthetic image interaction domain used for all experiments. Given an image, imagine a virtual robot that can observe a small patch on that image and also take actions to move the patch around the larger image. This “image robot” provides an excellent domain in which subjective planning can be demonstrated.

For these experiments, IMAGEBOT will always be viewing a 200 by 200 patch of a 2048 by 1536 image displayed Figure 1. IMAGEBOT has eight distinct actions: four translations, two zoom actions, and two rotation actions. The allowed translations are forward (F), back (B), left (L) and right (R) by 25 pixels. The zoom changes the scale of the underlying image by a factor of $2^{1/8}$ (i) or $2^{-1/8}$ (o). The rotation rotates the square left (l) or right (r) by $\frac{\pi}{8}$ radians.

There are three distinct experimental data sets that are looked at in this paper.

³Notice this is not requiring the actions in the objective space to be rotations and translations, since ARE is learning a non-linear feature representation.

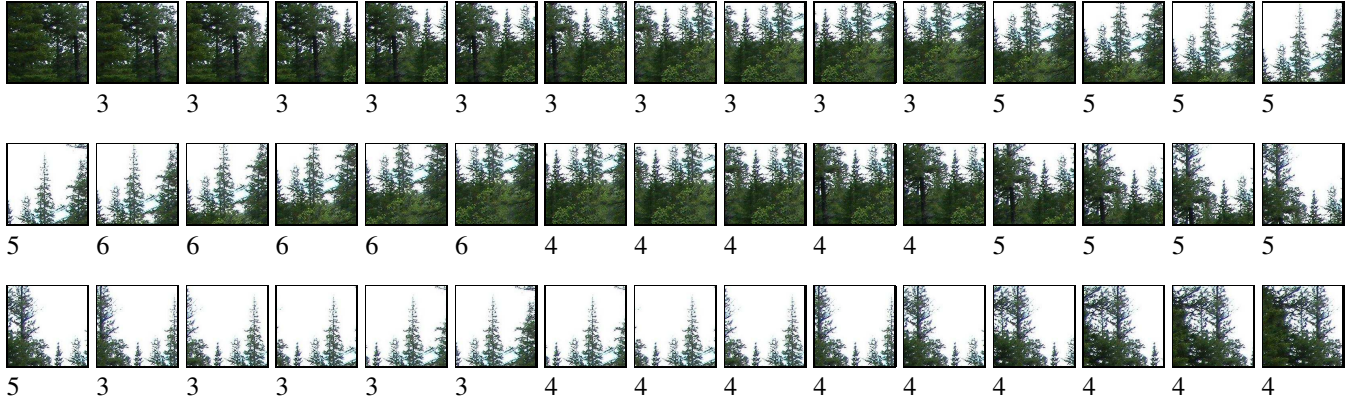


Figure 2: IMAGEBOT's output (and ARE's input) for the A_T data set.



Figure 1: IMAGEBOT's world.

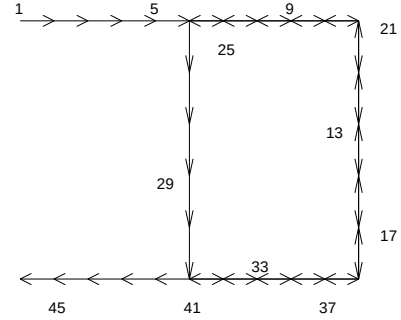


Figure 3: The path IMAGEBOT follows to generate A_T .

A_T : IMAGEBOT follows a path which looks like an “A” using only the translation actions:

$$F \times 10, L \times 5, R \times 5, B \times 5, L \times 5, F \times 5, B \times 10$$

A_Z : IMAGEBOT follows the same A pattern but substituting zoom in for left actions and zoom out for right actions:

$$F \times 10, i \times 8, o \times 8, B \times 5, i \times 8, F \times 10, B \times 20$$

Fr : IMAGEBOT moves back and forth in a line, but only using F and r actions:

$$F \times 10, r \times 8, F \times 10, r \times 8, F \times 5, r \times 16, F \times 5$$

Note that in A_Z the F and B actions only move half as much when zoomed in as when zoomed out. Note that in Fr there are no opposites for the two actions used. An example of the output of IMAGEBOT, and correspondingly an input for ARE, is shown in Figure 2. These are the images seen in the A_T data set. Note that while we know that, for example, action label 3 corresponds to action F , ARE gets no such semantic information—it gets as input only the images and the labels associated with them.

The effectiveness of ARE has been demonstrated previously [Bowling *et al.*, 2005]. Here evidence is shown of its power in capturing useful representations for planning. Figure 3 shows the actual manifold underlying the A_T test set (*i.e.*, IMAGEBOT's path). Figure 4 shows the manifold learned by ARE on that test set. Clearly the structure has been captured.

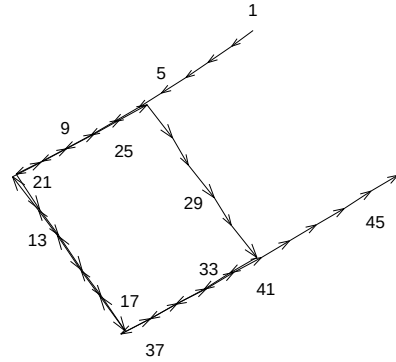


Figure 4: The representation learned by ARE for A_T .

Figure 5 shows a “top” view and a “side” view of the 3-dimensional manifold learned for the Fr data set. here the portion of the manifold corresponding to rotating to the right (r) is a black line while the portion corresponding to moving forward (F) is a light-gray line. The point corresponding to the first image is circled. Although not as clear as in the previous case, this manifold is clearly and distinctly capturing the structure of the original path, with two dimensions capturing the r action and a third capturing the F action. In Figure 3, the original domain was one in which the actions were distance-preserving and this structure was, indeed, extracted. In Figure 5 it is not immediately obvious what man-

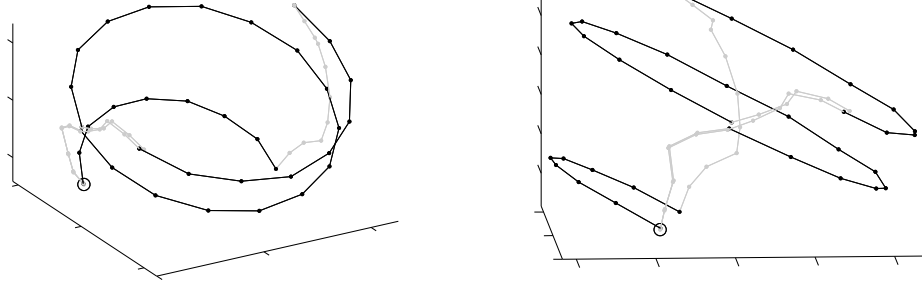


Figure 5: Two different views of the three-dimensional cylindrical manifold learned for the *Fr* data set.

ifold will be learned which makes the resulting one all the more impressive—not just as a representation appropriate for planning but as an aid to intuitive understanding of the original underlying structure.

3 Distance Preserving Operators

ARE learns a representation with explicit constraints that the actions correspond to distance-preserving transformations in that representation. Before one can plan, though, one needs to discover these transformations. For each unique action a there is a collection of data point pairs (x_t, x_{t+1}) which are connected by that action. Another way of thinking of this is that there is a function f_a where $f_a(x_t) = x_{t+1}$, and such a function needs to be learned for each action. Because of the distance-preserving constraints, f_a can be represented as:

$$f_a(x_t) = A_a x_t + b_a = x_{t+1}$$

Recall that transformations of the above form encode translation in the b_a vector, and rotation and scaling in the A_a matrix. A_a and b_a could be learned using simple linear regression but scaling is not distance preserving so there is the additional constraint that A_a does not scale, i.e., $A_a^T A_a = I$. It turns out that this is similar to the extended orthonormal Procrustes problem [Schoenemann and Carroll, 1970], but without allowing for a global scaling constant. Here the solution to the regression problem is derived.

Let X_a be the d by n matrix whose columns are x_t for all t such that $a_t = a$, and let Y_a be the d by n matrix whose columns are x_{t+1} for the same t . The goal is to learn a rotation matrix A_a and a translation b_a which maps X_a to Y_a . Formally, the following optimization problem needs to be solved.

$$\begin{aligned} \text{minimize: } & \|A_a X_a + b_a e^T - Y_a\| \\ \text{subject to: } & A_a^T A_a = I \end{aligned}$$

where e is a column vector with n ones.

In order to obtain the least squares estimation of A_a and b_a , write the Lagrangian function L :

$$\begin{aligned} L(A_a, b_a, \Lambda) = & \\ & \text{Tr}((A_a X_a + b_a e^T - Y_a)^T (A_a X_a + b_a e^T - Y_a)) + \\ & \text{Tr}(\Lambda(A_a^T A_a - I)) \end{aligned}$$

where Λ is a matrix of lagrangian multipliers, and $\text{Tr}(\cdot)$ stands for the trace of a matrix.

$$\begin{aligned} L(A_a, b_a, \Lambda) = & \\ & \text{Tr}(Y_a^T Y_a) + \text{Tr}(X_a^T A_a^T A_a X_a) + n b_a^T b_a \\ & - 2\text{Tr}(Y_a^T A_a X_a) - 2\text{Tr}(e b_a^T Y_a) \\ & + 2\text{Tr}(e b_a^T A_a X_a) + \text{Tr}(\Lambda(A_a^T A_a - I)) \end{aligned}$$

Now take the derivative of the Lagrangian function with respect to the unknowns and set to zero:

$$\begin{aligned} \frac{\partial L}{\partial A_a} = & 2A_a X_a^T X_a - 2Y_a X_a^T + \\ & + 2b_a e^T X_a^T + A_a(\Lambda + \Lambda^T) = 0 \end{aligned} \quad (4)$$

$$\frac{\partial L}{\partial b_a} = 2n b_a - 2Y_a e + 2A_a X_a e = 0 \quad (5)$$

The translation vector from Equation 5 gives:

$$b_a = \frac{(Y_a - A_a X_a)e}{n} \quad (6)$$

Multiplying Equation 4 by $A_a^T/2$ on the right:

$$\begin{aligned} A_a X_a^T X_a A_a^T + A_a(\Lambda + \Lambda^T) A_a^T/2 = \\ Y_a X_a^T A_a^T - b_a e^T X_a^T A_a^T \end{aligned}$$

Since the left hand side is symmetric, the right hand side must also be symmetric. Substituting Equation 6, the right hand side can be written as:

$$Y_a X_a^T A_a^T - Y_a \left(\frac{e e^T}{n} \right) X_a^T A_a^T + A_a X_a \left(\frac{e e^T}{n} \right) X_a^T A_a^T$$

where the last term is symmetric. Thus the rest of the expression must also be symmetric. This can be simplified as:

$$\left(Y_a \left(I - \frac{e e^T}{n} \right) X_a^T \right) A_a^T \quad (7)$$

Since 7 is symmetric, it should be equivalent to its transpose:

$$\left(Y_a \left(I - \frac{e e^T}{n} \right) X_a^T \right) A_a^T = A_a \left(Y_a \left(I - \frac{e e^T}{n} \right) X_a^T \right)^T \quad (8)$$

One can easily verify that the following satisfies Equation 8:

$$\begin{aligned} V S W^T &= \text{svd} \left(Y_a \left(I - \frac{e e^T}{n} \right) X_a^T \right) \\ A_a &= V W^T \end{aligned} \quad (9)$$

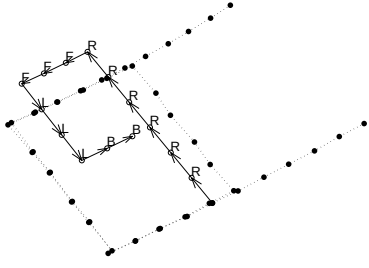


Figure 6: Demonstrating distance-preserving operators in the representation learned for A_T .

where $\text{svd}(\cdot)$ is the singular value decomposition, so,

$$VSW^T W V^T = V W^T W S^T V^T$$

because $W^T W = I$ (since W is orthonormal) and $S = S^T$ (since S is diagonal). Thus Equations 6 and 9 are a solution to the dual function, $D = \min_{\Lambda} L(A_a, b_a, \Lambda)$. Since this solution is also feasible with regards to the primal problem, strong duality holds even though the original problem is non-convex.

Results. Figure 6 shows the two-dimensional representation that ARE generated for A_T . The solid arrows show a path which consists of new points resulting from the application of operators learned for each action (F , B , L and R) as described above. Clearly, the operators are intuitively capturing the essence of the actions used to generate the data.

Note that while there is no semantic meaning given with the input actions, such meaning can now be derived. It can easily be tested whether a pair of actions are opposites, such as (F, B) or (R, L) . Also two actions can be tested for orthogonality or independence, such as F and R . If the learned representation captures some underlying structure within the data, the learned operators will maintain that structure and, from them, relationships can be successfully hypothesized.

Figure 7 is similar to Figure 6, except the underlying representation is from the A_Z data set. Here, note that while some actions are again opposite to each other, no actions are orthogonal—the F and B actions are not independent of the i and o actions. This critical facet of the original data set has been successfully captured in the representation learned by ARE and consequently in the action functions learned in that manifold. Note, in particular, that when zoomed in all the way ($i \times 8$) 10 F actions are equivalent to 5 F actions when zoomed out all the way. Although F action when zoomed in half way was never observed, the learned operators capture the fact that between 7 and 8 F actions at this scale are equivalent to 5 and 10 actions at the other scales.

4 Planning

Now that low-dimensional representations and operators in those representations can be learned, all the pieces are in place to perform planning. The points in the learned representation are states, and the operators learned in Section 3 define transitions between the states. This new domain has two advantages over the original data set. First, the dimensionality has been reduced drastically (from 40,000 to 2 or 3). Second,

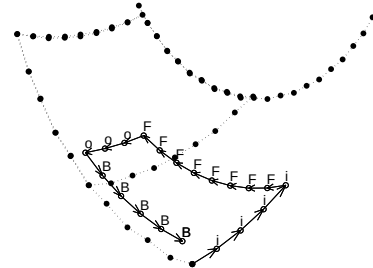


Figure 7: Demonstrating distance-preserving operators in the representation learned for A_Z .

from the action labels functions have been learned for each unique action that explicitly give the resulting state when that action is applied from any state. Learning such a function in the original space is not only intractable, but would require application of knowledge specific to IMAGEBOT (or even the underlying image) in order to attain any success.

Given any two images, one can find a shortest path between them, even if it traverses unobserved parts of the space. First, find the corresponding points in the low-dimensional representation, then find the shortest path between them using traditional search methods and the set of learned operators. Since each operator in the low-dimensional space corresponds to an action label in the original space the list of action labels that indicate the desired path can be returned. For the following results, iterative-deepening depth-first search was used and the path whose final point was closest to the desired goal was returned. The quality of a path is demonstrated by starting IMAGEBOT at the initial state, applying the sequence of actions and showing the resulting image.

Results. For each of the three test sets, two sub-figures will be shown. The first shows the representation learned for that data set and the shortest path between a chosen initial state (labelled with a triangle pointing right) and a chosen goal state (labelled with a triangle pointing left). The second figure contains two images. The left image shows the image at the initial state, the right image contains two highlighted boxes. The light-gray dotted box shows the image at the goal state, the darker-gray solid box highlights the image obtained after executing the resulting sequence of actions.

Figures 4(a) and 4(b) show the results for A_T —the goal state image and the image corresponding to the final state in our path in Figure 4(b) are the same. Note that the shortest path was successfully found, even though it involves moving through portions of the space that we have never seen.

Figures 4(c) and 4(d) show the results for A_Z —the goal state image and the image corresponding to the final state in the path in Figure 4(d) are the same. Note, the path found successfully identifies that action B must occur before action o —if taken after then the B action would jump over the desired end state to a state halfway between it and the next state.

Figures 4(e) and 4(f) show the results for Fr —the goal state image and the image corresponding to the final state in the path in Figure 4(f) are very close to each other. Recall that for this data set, unlike the others, the only actions were

Maximum A Posteriori Path Estimation with Input Trace Perturbation: Algorithms and Application to Credible Rating of Human Routines

Daniel H. Wilson
Robotics Institute
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh PA 15213
dwilson@cs.cmu.edu

Matthai Philipose
Intel Research
6th Floor
1100 NE 45th Street
Seattle WA 98115
matthai@cs.washington.edu

Abstract

Rating how well a routine activity is performed can be valuable in a variety of domains. Making the rating inexpensive and credible is a key aspect of the problem. We formalize the problem as MAP estimation in HMMs where the incoming trace needs repair. We present polynomial time algorithms for computing minimal repairs with maximal likelihood for HMMs, Hidden Semi-Markov Models (HSMMs) and a form of HMMs constrained with a fragment of the temporal logic LTL. We present some results to show the promise of our approach.

1 Introduction

Rating how well a person performs a routine activity is a broadly useful capability with many applications: professors train medical students by rating their execution of established procedures, caregivers assess the well-being of their wards by rating how well they are able to perform activities of daily living, and managers and workflow experts identify poorly performed procedures that cause bottlenecks in a system. Although rating routine activity is certainly useful, as conventionally done it is also very expensive – each activity performance requires a dedicated human observer (often an expert). Many situations where gauging the performance of routine activities could be helpful are therefore either not rated at all, or rated in a cursory manner. Clearly, an opportunity exists for automated techniques to reduce the cost of rating. In this paper, we explore methods for automatically rating performances of routine activities.

The basic classification task of rating, going from observations to scores, is amenable to a variety of standard approaches. Rating becomes challenging, however, if we wish to make it both *incrementally inexpensive* and *credible*. We define an incrementally inexpensive rater to be a rater in which the extra cost of rating a new activity is relatively low. The main determinant of cost is whether rating a new activity requires a custom classifier to be developed from scratch, or whether a generic classifier of some kind can be easily customized to the task. A credible rater is one that is both *relevant* and *transparent*. By relevant, we mean that the classification model for a particular rating task should reflect constraints on activity performance that are important to those

using the rating. For example, a professor grading anesthesia students performing an intubation may want to indicate what her notion of good performance is. By transparency, we mean that the system should be able to justify why it has assigned a particular rating. Ideally, the justification should be *constructive*, in that it should suggest how a low-rated performance may be altered to obtain a high-rated one.

Our techniques for rating activity routines are designed to satisfy the above requirements. To lower incremental cost, we choose a representation that is easily learned: all activities to be rated in our system are modeled by variants of Hidden Markov Models (HMMs). We intend that these models, especially given simple prior information, can be learned easily from training examples. More crucially, we formulate the justification for a rating relative to this model generically as the set of edits required on the trace generated by the rated activities; we therefore do not require special identification and modeling of errors and their causes. A fundamental weakness of these models is that they are first order, preventing them from capturing certain important correlations. We augment the Markov models with an intuitive constraint formalism (a small fragment of the temporal logic LTL [1]) that allows raters to explicitly state relevant constraints. Given these relevant and easy-to-construct models, we formulate rating as the likelihood of (possibly edited) observation sequences.

The core of this paper consists of efficient algorithms to compute maximum likelihood paths of minimally edited versions of incoming observations with respect to various representations for activities, including HMMs, HSMMs and temporally constrained HMMs. The algorithms use the dynamic programming technique used to great effect by the well-known Viterbi algorithm. A preliminary evaluation shows the promise of our techniques.

2 Overview

In this section, we describe how we expect our system to be used, and we sketch how our system supports this usage model. In this paper, our goal is to develop a system that rates how well an elder performs day-to-day activities. Such a system is of great interest to the eldercare industry. In theory, caregivers will assess the elders' well-being by consulting ratings summaries and credible explanations of performance deficits. For example, the system may recognize that an elder is no longer able to prepare their daily bowl of soup,

and report why (e.g., can't reach cabinet or difficulty holding spoon).

To end-users, our system represents activities as a set of steps. Each step has a duration and a set of observed actions performed, and is succeeded by other steps. For instance, the activity “making soup” for a particular elder may have the following steps: “preheat water,” “open can,” “mix and boil ingredients,” “serve,” and “clean up.” The step “open can,” may have an average duration of 45 seconds and contain the following actions: “use utensil drawer,” “use can opener,” “use can,” and “use pantry door.”

For concreteness, we will assume in what follows that we are using RFID-based [4] sensors that will directly sense the action of using particular objects. Therefore, all of our actions are of the form “use X” where X is some object. Inherently, our system requires that actions are observable by sensors. Given an activity trace (i.e., a trace of actions that constitutes a particular execution of an activity), our system provides a rating (e.g., *pass* or *fail*). If the grade is a *fail*, the system provides an alternate sequence of actions as close to the original as possible that would have elicited a *pass* grade (essentially a constructive justification of the grade). In more detail, use of the system proceeds as follows:

Learning the model A human demonstrator performs the routine in an exemplary fashion. The system collects traces $Y_1, \dots, Y_n$ of the routine. Each trace Y_i is a sequence of time-stamped observations $y_{i1}, \dots, y_{im_i}$ of the demonstrator's actions. The traces are used to learn a dynamic stochastic model (either an HMM or an HSMM) with parameters λ . The hidden states $s_1, \dots, s_N$ of the model correspond to the “activity steps” above, and are labelled $l_1, \dots, l_N$ with the names of the step.

Adding global constraints Typically, the first-order model learned in the previous step cannot capture important higher-order correlations. For instance, in a successful soup-making routine, the stove, if it is used, should eventually be turned off after it is turned on. The turning on would happen in the “preheat water” step, but the turning off may not happen until the end of the “serve” step. The human rater explicitly adds a set $\mathcal{C}$ of constraints on the sequence of hidden states or observations that specify these required higher-order correlations. In this case, a possible constraint would be of the form $\text{use}(\text{“stove control knob”}) \mathcal{E} \text{use}(\text{“stove control knob”})$, read as “a use of a stove control knob should eventually succeeded by a use of a stove control knob”.

Learning rating thresholds A human rater rates each trace Y_i with a rating $r_i \in \{\text{pass}, \text{fail}\}$. Let the *constrained MAP likelihood* of trace Y given λ and $\mathcal{C}$, $\hat{l}_Y = \text{CMAP}(M, Y, \mathcal{C})$, be the likelihood of the path $\hat{S}_Y$ with maximum a posteriori (MAP) likelihood given λ and Y that satisfies $\mathcal{C}$. We perform a simple thresholding computation to calculate the likelihood threshold L such that, given the classification function $R(l) = \text{if } l < L \text{ then fail else pass}$, $R(\hat{l}_{Y_i}) = r_i$ for as many of the Y_i as possible. Intuitively, L separates the passes from the fails.

Generating a rating and justification Given the constrained model $(\lambda, \mathcal{C})$ and threshold L , the automated rater is ready for use. The person to be rated generates a trace $Y = y_1, \dots, y_m$ to be rated automatically. The rater finds the constrained MAP likelihood $\hat{l}_Y$ and path $\hat{S}_Y = (\hat{s}_1, y_1), \dots, (\hat{s}_m, y_m)$ for Y , and assigns it the rating $r = R(\hat{l}_Y)$. If $r = \text{fail}$, the rater attempts to produce a *repaired trace* trace $Y' = y'_1, \dots, y'_m$ such that the edit distance between Y and Y' is as small as possible, and $\hat{l}_{Y'} > L$. In other words, Y' is the closest trace to T that passes. The rater offers r as the rating for the activity and, if appropriate, $\delta_{\hat{S}_Y, \hat{S}_{Y'}}$, the set of edits needed to transform $\hat{S}_Y$ into $\hat{S}_{Y'}$, as the justification for the rating.

As described above, our rating system employs two key non-standard pieces of machinery.

1. A method to compute the repaired observation trace T' , that is a minimum edit distance from a given trace T whose likelihood is above a pre-specified threshold L .
2. A method to compute the constrained MAP likelihood function $\text{CMAP}(M, T, \mathcal{C})$.

3 Trace Repair for Hidden Markov Models

A Hidden Markov Model (HMM) $\lambda = (A, B, \pi)$ is a commonly used stochastic model for dynamic systems [5]. We formally pose the trace repair problem as a variation of estimating the most likely state sequence given a sequence of observations (classically solved via the Viterbi algorithm). An HMM is defined as follows. Let $Q_A = \{q_1, \dots, q_N\}$ be the states of the process being modeled, and $O_B = \{o_1, \dots, o_M\}$ the observation signals possibly generated by the process. We use meta-variables s_t and y_t to denote the states and observations respectively at time t . A_{ij} is the probability $p(s_{t+1} = q_j | s_t = q_i)$ of transitioning from state q_i at time t to q_j at time $t + 1$ for any t ; B_{ij} is the probability $p(y_t = o_j | s_t = q_i)$ of generating observation o_j when in state q_i (we write B_{iy_t} for B_{ij} such that $y_t = o_j$). The initial state distribution $\pi_i = p(s_0 = q_i)$.

3.1 The Repaired MAP Path Estimation Problem

We now formulate the problem of MAP path estimation given an observation sequence if we are allowed to first make a limited number of edits or “repairs” to the sequence. We begin by formalizing the notion of an edit. We then state the repaired MAP path estimation problem and present a variation of the Viterbi algorithm to solve it.

Let Y^N be the set of length- N strings of observations over some finite alphabet Y . Then $e^{k,N} = ((b_1, s_1), \dots, (b_N, s_N))$ is a *length- N k -edit vector* on Y^N , with b_i boolean, s_i strings over Y , and $k = \sum_{1 \leq i \leq N} (b_i + |s_i|)$. For instance, $\hat{y}_1 = \text{“cat”}$ is a string in Y^3 ; $e_1^{4,3} = ((\text{false}, \text{“BB”}), (\text{false}, \text{“”}), (\text{true}, \text{“R”}))$ is an edit vector on Y^3 . Applying an edit vector e to string $\hat{y}_n = y_1 \dots y_n$, written $e(\hat{y})$ results in a new string $\hat{y}'$ obtained as follows. For $1 \leq i \leq n$, let if $e.b_i$ is true, then replace y_i with $\hat{y}'_i$, else replace y_i with $y_i e.s_i$ ($e.s_i$ appended to y_i). For example,

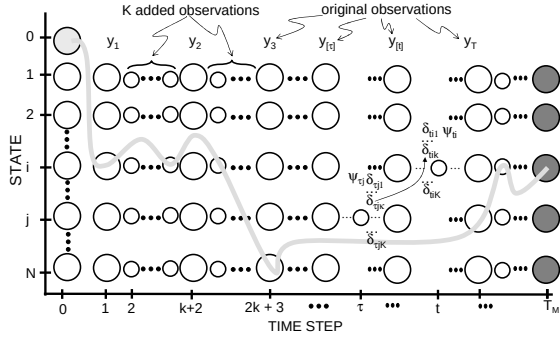


Figure 1: Trellis for k -Edit Viterbi on HMMs

$e_1^{4,3}(\hat{y}_1) = \text{"cBBaR"}$. A string $\hat{y}'$ is a k -edit of another $\hat{y}$ if there exists edit vector $e^{k,N}$ such that $\hat{y}' = e^{k,N}(\hat{y})$.

We are now ready to specify the problem of MAP estimation with repairs:

Definition 1. (Repaired MAP Path Estimation Problem (RMAP)) *Given observation sequence $\hat{y}_T$, HMM $\lambda = (A, B, \pi)$ and edit distance K find observation sequence $\hat{y}'_{T'} = \hat{y}_1 \dots \hat{y}_{T'}$ that is a K -edit of $\hat{y}_T$ and path $\hat{s}_{T'} = \hat{s}_1, \dots, \hat{s}_{T'}$ maximizing $p(\hat{s}_{T'}, \hat{y}'_{T'})$ over all T' , $\hat{s}_{T'}$ and $\hat{y}'_{T'}$.*

Before discussing our solution, we define string $\hat{y}'$ as the (k, a) -edit of string $\hat{y}_n$ if $\hat{y}' = e^{k,n}(\hat{y})$ for some $e^{k,n}$, $|e^{k,n}.s_n| \leq a$, and additionally, $e^{k,n}.b_n$ if $a = 0$ and $|e^{k,n}.s_n| > 0$ if $a > 0$. The (k, a) -edit of a string requires its last character to be either preserved or replaced by at least one character, with at most a characters added. Edits compose as follows ($d_{\nu n} = n - \nu$; $a' = 1$ if $a \neq 0$, 0 otherwise; $(\nu, \alpha) <_k(n, a)$ if $\nu < n$ and $\alpha \leq k$, or if $\nu = n$ and $\alpha < a$):

Lemma 1. (Edit Composition) *Let $Y_n^{ka\hat{y}}$ be the set of all (k, a) -edits of the n -prefix of string $\hat{y}$ over Y . Then $Y_n^{ka\hat{y}} = \{\hat{y}' | \hat{y}' \in Y, \hat{y}' \in Y_{\nu}^{\kappa\alpha\hat{y}}, (\nu, \alpha) <_k(n, a), \kappa + d_{\nu n} + a' = k\}$.*

Table 1 specifies an algorithm (the k -Edit Viterbi (KEV) algorithm) to solve RMAP. KEV iterates over the T original observations in the incoming observation string. For each original observation, it iterates over possibilities for the K added observations at that position, for a total of $TK + T$ iterations. At each iteration t corresponding to original observation $[t] = t \div (K + 1) + 1$ and added observation $\#t = t \bmod (K + 1)$, KEV computes the likelihood δ_{tik} of the most likely path ending in state i given an observation string that is a $(k, \#t)$ -edit of $y_1 \dots y_{[t]}$ over all such edit vectors; KEV also records as ψ_{tik} the penultimate state and edit in this path. Following the chain of ψ_{tik} 's back to the start state iteration gives the MAP repaired path.

The trellis of figure 1 illustrates KEV. Columns of the trellis represent edits considered for inclusion into the final string. Large circles represent original observations and small ones represent adds. For technical reasons (to allow skipping the first original observation), we add a distinguished start state q_0 with new start probabilities $\pi'_0 = 1$, $A_{0i} = \pi_i$ and $A_{i0} = 0$ and add a column ($t = 0$) processed in the initialization step. To allow skipping the last original observation with

Initialization: $t = 0, k = 1 \dots K, 1 \leq i \leq N$

$$\delta_{t0k} = 1 \quad \delta_{tik} = 0 \quad \psi_{tik} = -1$$

Iteration: $1 \leq t \leq TK + T, k = a_t, a_t + 1, \dots, K$

$$(\psi_{tik})\delta_{tik} = (\arg\max_{\tau, j, \kappa \text{ s.t. } \kappa + a_t + d_{\tau t} = k} \delta_{\tau j \kappa} B_{iy_{it}} A_{ji})$$

Termination: $t = T_M = TK + T + 1$

$$(\psi_{tik})\delta_{tik} = (\arg\max_{\tau, j, \kappa \text{ s.t. } \kappa + a_t + d_{\tau t} = k} \delta_{\tau i \kappa} ; i_M = \arg\max_{1 \leq i \leq N} \delta_{tiK})$$

Backtracking: $(t, i, k) = \psi_{T_M i_M K}$; while $t > 0$,

$$1) (\hat{s}_t, \hat{y}_t) = (q_i, y_{it}) \quad 2) (t, i, k) \leftarrow \psi_{tik}$$

Table 1: The k -Edit Viterbi Algorithm for HMMs

no adds, we add a column ($t = T_M = TK + T + 1$). Rows represent possible hidden states.

The end result of the algorithm is a forward path (shown in light grey in figure 1) through the trellis that, unlike in the conventional Viterbi algorithm, may jump between nodes in non-adjacent time slices. If the path jumps over the slice for an original observation y_i (where i is the position of the observation in the input string $\hat{y}$), we conclude that y_i was deleted from $\hat{y}$, otherwise not. Further, if the path passes through a sequence of added nodes with no intervening original node such that y_i is the first original observation to the left of the sequence, and the observations at these nodes are $y_{i_1}, \dots, y_{i_n}$, we conclude that the string $y_{i_1} \dots y_{i_n}$ was added at the i 'th spot in the incoming string. The forward path is the required solution $\hat{s}^{T'}$, and the string of observations along the path is the edited string $\hat{y}^{T'}$.

The algorithm uses three intermediate variables, a_t , $d_{\tau t}$ and y_{it} . Variable $a_t = 1$ if $\#t \neq 0$ and 0 otherwise; $d_{\tau t} = [t] - [\tau]$, represents the number of deletes skipping original observations between τ and t ; y_{it} is the observation considered when processing state i at slice t . Note that we only process original observations at time slices $1, K + 1, 2K + 1, \dots$. In all other "added" slices, we need to propose the observed value to be added. A simple but inefficient approach would be to consider for each state, k -value and iteration t , every possible observable $o \in O_B$ as a candidate. In fact, we can consider a single observation instead of all $|O_B|$. The key insight is that, when processing state i in an added slice, it is sufficient to consider adding as observable the most likely observable in that state. Let S_N and Y_N be the sets of all length- N sequences of states and observables. Let $\hat{y}_i = \arg\max_{1 \leq j \leq M} B_{ij}$. Let $\hat{s}_q$ be the result of appending state q to sequence $\hat{s}$, and similarly for $\hat{y}_y$. Then, for all $q_i \in Q_A$:

Lemma 2. $\max_{\hat{s}, \hat{y}' \in S_N, Y_{N+1}} p(\hat{s}q_i, \hat{y}') = \max_{\hat{s}, \hat{y} \in S_N, Y_N} p(\hat{s}q_i, \hat{y}\hat{y}_i)$

This follows from the fact that $\max_{y_i} p(\hat{s}q_i, \hat{y}y_i) = \max_{y_i} \pi_{s1} B_{s1y_1} (\prod_{1 \leq i \leq N, s_i q_j \in \hat{s}_N} A_{ij} B_{jy_j}) (A_{s_N i} B_{iy_i}) = \pi_{s1} \dots A_{s_N i} \max_{y_i} B_{iy_i} = \pi_{s1} \dots A_{s_N i} \hat{y}_i$. Given this identity for the optimal observable to be added in state q_i , we set y_{it}

to y_i if t is an “added” timeslice, and to $y_{[t]}$ otherwise.

We are now ready to establish the soundness of the KEV algorithm. Let S_n^i be the set of length- n sequence of states ending in state q_i . Let Y_n^{tik} be the set of length- n strings of observables that are $(k, \#t)$ -edits of $y_1 \dots y_{[t]}$.

Lemma 3. $(\psi_{tik})\delta_{tik} = (\arg)\max_{n, \hat{s} \in S_n^i, \hat{y} \in Y_n^{tik}} p(\hat{s}, \hat{y})$

Proof sketch. Proof is by induction on t . We focus on the inductive case for δ . For ψ , replace “max” with “argmax”.

$$\begin{aligned} \delta_{tik} &= \max_{\tau, j, \kappa} A_{ji} B_{iy_{it}} \delta_{\tau j \kappa} \forall_{j, \tau, \kappa} \text{ s.t. } (\kappa + a_t + d_{\tau t}) = k \\ &= (\text{by the inductive hypothesis}) \\ &\quad \max_{\tau, j, \kappa} (A_{ji} B_{iy_{it}} \max_{n, \hat{s} \in S_n^j, \hat{y} \in Y_n^{\tau j \kappa}} p(\hat{s}, \hat{y})) \end{aligned}$$

Given s_n, s_{n+1} is independent of $\hat{s}^{n-1}, \hat{y}^n$:

$$\begin{aligned} A_{ji} &= p(s_{n+1} = q_i | \hat{s}, s_n = q_j, \hat{y}) \quad \forall_{n, \hat{s} \in S_{n-1}, \hat{y} \in Y_n^{\tau j \kappa}} \\ &= p(s_{n+1} = q_i | \hat{s}, \hat{y}) \quad \forall_{n, \hat{s} \in S_n^j, \hat{y} \in Y_n^{\tau j \kappa}} \end{aligned}$$

Similarly, for y_{n+1} given s_{n+1} :

$$B_{iy_{it}} = p(y_{n+1} = y_{it} | \hat{s}, s_{n+1} = q_i, \hat{y}) \quad \forall_{n, \hat{s} \in S_n^j, \hat{y} \in Y_n^{\tau j \kappa}}$$

Substituting for A_{ji} and $B_{iy_{it}}$ above, and using $p(A, B) = p(A|B)p(B)$ twice, we have, with $(\kappa + a_t + d_{\tau t}) = k$:

$$\delta_{tik} = \max_{\tau, j, \kappa, n, \hat{s} \in S_n^j, \hat{y} \in Y_n^{\tau j \kappa}} p(s_{n+1} = q_i, \hat{s}, y_{n+1} = y_{it}, \hat{y})$$

Lemma 2 ensures that maximizing over $\hat{y} y_{it}$ maximizes over all strings $\hat{y} y$. By lemma 1 maximizing over all $\hat{y} y$ with $\hat{y} \in Y_n^{\tau j \kappa}$ maximizes over $\hat{y} \in Y_{n+1}^{tik}$. Finally, $\forall_{1 \leq j \leq N, \hat{s} \in S_n^j} \hat{s} = \forall_{1 \leq i \leq N, \hat{s} \in S_{n+1}^i} \hat{s}$. Modifying the previous equation to reflect these insights:

$$\delta_{tik} = \max_{n, \hat{s} \in S_{n+1}^i, \hat{y} \in Y_{n+1}^{tik}} p(\hat{s}, \hat{y}) = \max_{n, \hat{s} \in S_n^i, \hat{y} \in Y_n^{tik}} p(\hat{s}, \hat{y})$$

□

The soundness of KEV follows in a straightforward way from the above lemma. Further, given that the trellis has $O(TKN)$ nodes, that at each node we compute $O(K)$ δ and ψ values, and we consult $O(NK)$ preceding data values to do so, the complexity of KEV as a whole is $O(TN^2K^3)$.

4 Trace Repair for HSMMs

A Hidden Semi-Markov Model (HSMM) [3] $\lambda = (A, B, D, \pi)$ is identical to an HMM except for the *duration distribution* D . Where an HMM generates a single observation according to B on each visit to a state s , the HSMM generates l independent observations from B on each visit, where l is drawn according to $D_{sl} = p(l|s)$. The added flexibility is useful when modeling human activities, since the duration of stay in a state is restricted to be geometric (and therefore biased to small values) in HMMs. In what follows, we assume that D is over a finite set (of size $|D|$) of durations, where the longest duration is L steps.

The RMAP problem is: given HSMM (A, B, D, π) , observations $\hat{y}^T$ and limit K , find $\arg\max_{t, \hat{s} \in Q_A^t, \hat{y} \in O_B^t, \hat{y} \text{ k-edit of } \hat{y}^T} p(\hat{s}, \hat{y})$.

Init., Term., Backtracking: See table 1.

Iteration: $1 \leq t \leq TK + T$, $k = a_t, a_t + 1, \dots, t$

$$(\psi_{tik})\delta_{tik} = (\arg)\max_{1 \leq \tau \leq t, j, \kappa, y_{\kappa k \tau ti}} \delta_{\tau j \kappa} p(y_{\kappa k \tau ti}) A_{ji} D_{i|y_{\kappa k \tau ti}|}$$

Table 2: The k -Edit Viterbi Algorithm for HSMMs

Table 2 specifies a variant of KEV to solve the problem, and figure 2 shows a trellis for this algorithm. The trellis is identical to that used by KEV (we represent k added nodes with a single small circle), only its use is different. We focus on how δ_{tik} is calculated. At each timestep t , state i and edit distance k , as with KEV, we iterate over previous timesteps, states and edit distances τ, j and κ . However, this time instead of discarding the observations in the intervening timesteps, we seek their sub-sequence $y_{\kappa k \tau ti}$. We assume that step t only ends a stay in state q_i that begins immediately after the stay in q_j that ended in step τ . If $e_{\tau t}$ is the number of edits in $y_{\kappa k \tau ti}$ (added nodes included + original nodes ignored), we require $\kappa + a_t + e_{\tau t} = k$. The problem of maximizing the likelihood of the path ending at (i, t) then reduces to the problem of finding $y_{\kappa k \tau ti}$ maximizing $p(y_{\kappa k \tau ti}) D_{i|y_{\kappa k \tau ti}|}$.

We find this maximum by iterating through durations l in D_i ; for each l , we iterate through predecessors (τ, j, κ) of (t, i) , finding a sequence $y_{\kappa k \tau ti}$ of length l with the highest probability; we keep a running tally of the maximum $p(y_{\kappa k \tau ti}) D_{i|y_{\kappa k \tau ti}|}$. Finding $y_{\kappa k \tau ti}$ reduces to identifying N_A added nodes (to include in $y_{\kappa k \tau ti}$) and N_O original nodes (to ignore), such that $N_A + N_O = k - \kappa - a_t$ (to satisfy the k -edit criterion), and $N_A + (([t] - [\tau]) - N_O) = l$ (to satisfy the duration constraint). The two equations fix N_A and N_O . Since all the added nodes have the same probability $y_i = p(o_i|q_i)$, it doesn’t matter which particular N_A we pick. On the other hand, we pick the N_O original nodes with lowest probability of observation for exclusion; this can be done by sorting the original nodes in $O(T \log T)$ time *offline*, with $O(L)$ access during execution. Once the sequence of nodes is picked to get $y_{\kappa k \tau ti}$, we simply multiply their observation and transition probabilities together to get $p(y_{\kappa k \tau ti})$, a process that takes l operations, since $|y_{\kappa k \tau ti}| = l = O(L)$.

Given $O(TNK)$ trellis nodes, computing $O(K)$ δ and ψ values at each node, consulting $O(|D|NK)$ preceding values for each value, and spending $O(L)$ for each preceding value considered, the entire algorithm takes $O(TN^2|D|LK^3)$ steps. Note that in the (fairly) common case that D and L are unbounded, this running time becomes $O(T^3N^2K^3)$.

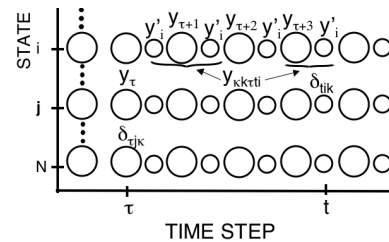


Figure 2: Trellis for KEV on HSMMs

5 Trace Repair for Constrained HMMs

We define a *temporally constrained HMM (TCHMM)* as $\lambda = (A, B, C, \pi)$, where C is a *temporal constraint* of the form $\phi_1 \mathcal{E} \phi_2 \mathcal{E} \dots \mathcal{E} \phi_{|C|}$. The ϕ_i are propositional boolean formulas over state labels l and observations y : $\phi ::= \text{state}(l) | \text{obs}(y) | \phi \wedge \phi | \neg \phi$. Path suffix $s_i \dots s_T$ and observations $y_i \dots y_T$ satisfy the constraint suffix $C_j = \phi_j \dots \phi_W$ if for any $k \geq j$, $\phi_j(s_k, y_k)$ implies that $(s_{k+1} \dots s_T, y_{k+1} \dots y_T)$ satisfy C_{j+1} , written $(s_{k+1} \dots s_T, y_{k+1} \dots y_T) \vdash C_j$. Intuitively if one formula in the constraint sequence is true w.r.t. the head of the state/observation sequences, then the formulas that follow must also eventually be true in their specified order later in the sequences. The constraint $(\text{state}(\text{COOK}) \wedge \text{obs}(\text{oil})) \mathcal{E} (\text{state}(\text{WASH}) \wedge \text{obs}(\text{soap}))$ could, for instance capture the constraint that if oil is used in the cooking step of making dinner, soap should be used in the eventual required washing step.

The RMAP problem may now be reformulated as given TCHMM with constraints C , observations $\hat{y}^T$ and limit K , find $SY = \underset{t, \hat{s} \in Q_A^t, \hat{y} \in O_B^t, \hat{y} \text{ k-edit of } \hat{y}^T}{\text{argmax}} \phi(\hat{s}, \hat{y})$ such that $SY \vdash C$.

Our solution for RMAP estimation is restricted to formulas of the form $\phi ::= \text{state}(l) | \phi \wedge \phi | \neg \phi$ (we disallow dependences on observables). A small modification to the KEV algorithm enables polynomial time solution of this problem. We use the same trellis as in KEV. For each timestep t , state i and edit distance k , we also now maintain an additional $|C|$ -vector. An element δ_{tikm} with $0 \leq m < |C|$ represents the likelihood of the MAP path ending at state i in time slice t with $(k, \#t)$ edits that still requires constraint suffix C_{m+1} to be satisfied (except δ_{tik0} , which has no outstanding constraints to be satisfied). This likelihood can be computed compositionally from $\delta_{\tau j \kappa \mu}$, with $\tau < t$ and (κ, μ) pointwise $\leq (k, m)$ in $O(TN^2 k^3 |C|^2 P)$ steps, where formulas ϕ_i can be evaluated in $O(P)$ steps (where P is the size of the formulas).

Even MAP estimation (without trace perturbation) for TCHMM's has apparently neither been formulated nor solved previously, although it is potentially quite powerful. For instance, the constrained inference work of Culotta *et al.* [2] is a special case of TCHMM k -edit MAP estimation (with $k = 0$, and $C = \text{state}(q_0) \mathcal{E} \text{state}(q_i)$). MAP estimation is a special case of RMAP estimation with $k = 0$. Our variant of KEV above therefore performs MAP estimation. Interestingly with $k = 0$, we can allow the more general version of formulas ϕ and still retain the fast running time. It is open how general C can be while remaining tractable. For instance, our constraints can be viewed as a fragment of Linear Temporal Logic (LTL) [1]. It is interesting to consider larger fragments as candidates.

6 Evaluation

How does model choice affect advice? The k -edit Viterbi algorithm dispenses advice based on the parameters of its activity models. The credibility of this advice will suffer from any differences between these models and the reality they represent. In order to illustrate this point, we conducted two experiments over three different activity models. The two experi-

ments compare a regular HMM and a time-sensitive HSMM, and a regular HSMM with an HSMM that has temporal logic constraints, respectively.

First, we compare the output of HMMs versus HSMMs on three activity traces from different activity models (see the top row of Figure 3). Each activity trace was intentionally made incorrect: for making tea, the preparation step was hurried; for making a sandwich, not enough ingredients were collected; and for grooming, brushing teeth and combing hair were performed too rapidly. In the top row of Figure 3, we plotted the maximum likelihood values at each step of the activity traces (where a “step” is considered to be a state transition). HMMs fail to detect any problem, exhibiting high likelihood. However, HSMM likelihoods plummet, due to sensitivity to the amount of time spent in each state. The KEDIT trace correctly adds the proper number of observations to each state, resulting in a high likelihood.

Second, we compare the output of HSMMs with and without temporal logic constraints (TLCs) (see the bottom row of Figure 3). Again, we intentionally chose incorrect sequences for the three activities: for making tea, the stove is turned on but never turned off; for preparing a sandwich, the refrigerator door is opened and never closed; and for grooming, the sink water is turned on and never turned back off. In the top row of Figure 3, we plotted the maximum likelihood values at each step of the activity traces. Regular HSMMs fail to detect any problem, reporting high likelihood. HSMMs with TLCs report low likelihood, because they are only allowed to consider state-transitions which satisfy all constraints. In these traces, constraints are broken and alternate, low-likelihood, paths must be considered. The KEDIT trace correctly adds the necessary steps (i.e., turn off stove, shut refrigerator, and turn off sink), resulting in high likelihood.

How does the rating change as k increases? The k -edits Viterbi algorithm provides advice for up to K edits. Ideally, we desire a trace that is above the likelihood threshold with the minimum number of edits. One method is to incrementally increase k until the threshold is exceeded. For this reason, we are interested in how the likelihood changes as k increases.

In this experiment we ran k -edits Viterbi for HSMMs on an empty trace of the “making tea” activity. In Figure 4 we plotted the overall likelihood of each trace as the number of possible edits was increased. The dashed line is a threshold showing the likelihood of an acceptable “good” trace. Obviously, the original empty sequence had low likelihood. As k was increased from one to three, the algorithm was forced to assemble partially complete activity traces which had even lower overall likelihood. When $k = 4$ the algorithm formed a complete trace and met the threshold. As k increased further, the algorithm tweaked the sequence for a slightly higher likelihood. The most likely possible path was reached at $k = 6$. Afterwards, we see an “odd-even” effect as the algorithm is forced to add new (less likely) observations, and then opportunistically delete other observations. For $k \geq 9$, the likelihood drops as the algorithm performs too many modifications to the trace and is unable to reach the optimal solution.

How intuitive is the advice? We now examine the advice dispensed by k -edits Viterbi in several scenarios. We ran the

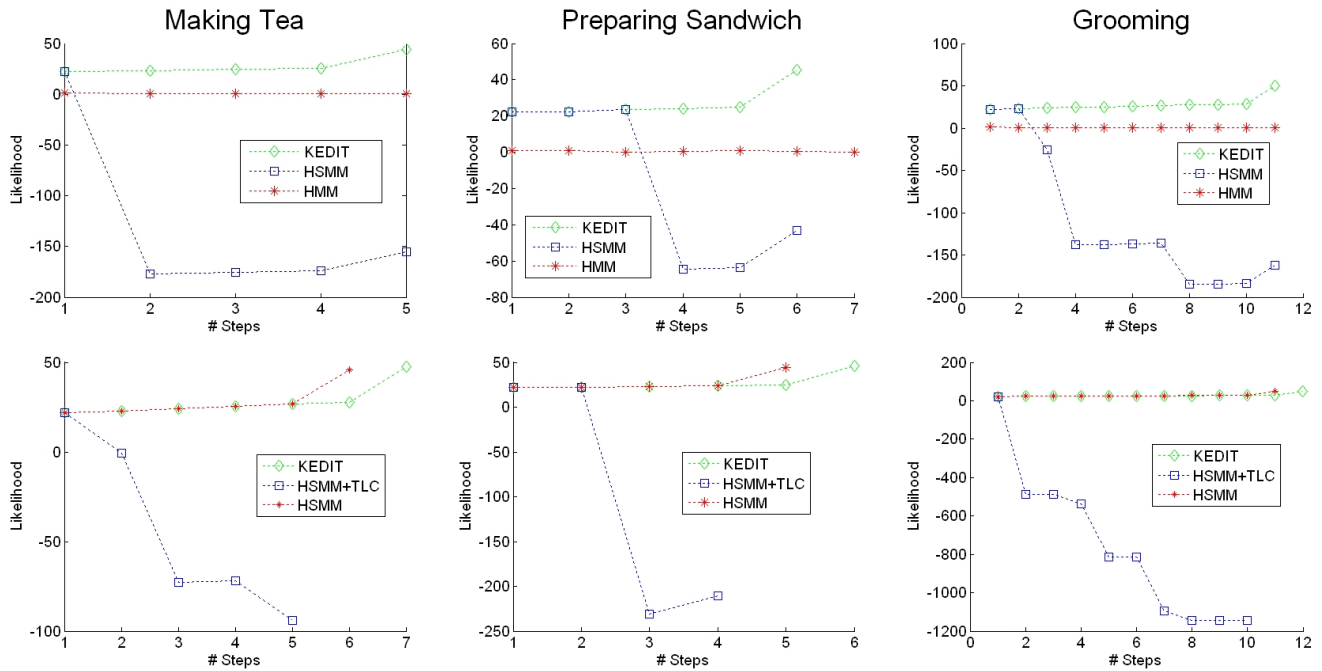


Figure 3: One activity per column; top row compares HMMs to HSMMs, bottom to TCHMMs

algorithm on activity traces that had the following problems: restarted the activity, got two steps out of order, performed a step too quickly, and missed or sped through several non-consecutive steps. All traces are from the “making tea” activity and likelihoods are reported using the optimal number of edits (i.e., k value).

The beginning steps of the next trace were performed twice (i.e., a “start” and a “restart”). The algorithm finds the maximum likelihood solution by deleting extra observations. However, the algorithm did not delete the entire start or restart, but decided to “pick and choose” among the two, keeping the best observations of both. In contrast, our intuition would be to advise the user to keep either the start or the restart. Similarly, in a trace in which two steps were performed out of order, the algorithm deletes one of the mis-ordered steps and inserts new steps in the correct position. We found this to be less intuitive than simply telling the user to switch the two steps to the correct order.

In the next trace, one step was performed too quickly: the “preparation step” only generated one observation, when it should have generated at least two. The algorithm suggested new observations that corrected the amount of time is spent in the state. However, the algorithm will always suggest the most likely observation from the state, because this maximizes the overall likelihood. At first glance, we found this suggestion strategy to be non-intuitive (although mathematically optimal), however, it became a non-issue for models in which observations were spread across multiple states.

In the last trace several non-contiguous steps were missed entirely or performed too quickly. As k increased, the algorithm first chose to insert states that had been missed entirely, and then to add more observations to states that had been vis-

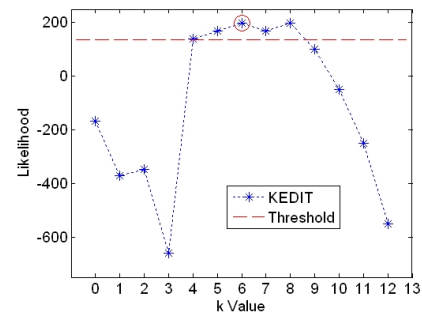


Figure 4: The likelihood of KEDIT traces as k increases.

ited too briefly. In other words, the algorithm advises the user to at least visit each step of the activity before it advises how to perfect each step. This “top-down” approach fits with our intuition of how advice should be given.

7 Conclusions

In this paper, we describe the credible activity rating problem. We introduced the k -edits Viterbi algorithm and showed that given model parameters and an activity trace it can provide optimally repaired traces with from zero to k edits. We improved the algorithm by incorporating high-level temporal logic constraints. Finally, we evaluated the strengths and limitations of the algorithm on data from three activity models.

References

- [1] E.M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, Boston, 2000.
- [2] A. Culotta and A. McCallum. Confidence estimation for information extraction. In *Proc. of Human Lang. Tech. Conf. (HLT-NAACL)*, 2004.
- [3] M. Ostendorf, V. V. Digalakis, and O. A. Kimball. From hmms to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Trans. Speech Audio Process*, 4(5):360–378, 1996.
- [4] M. Philipose, K.P. Fishkin, M. Perkowitz, D.J. Patterson, H. Kautz, and D. Hahnel. Inferring activities from interactions with objects. *IEEE Pervasive Computing Magazine*, 3(4):50–57, 2004.
- [5] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Mining Spatial Object Associations for Scientific Data*

Hui Yang, Srinivasan Parthasarathy and Sameep Mehta

Department of Computer Science and Engineering

Ohio State University

2015 Neil Avenue, Columbus, OH 43210

Abstract

In this paper, we present efficient algorithms to discover spatial associations among features extracted from scientific datasets. In contrast to previous work in this area, features are modeled as geometric objects rather than points. We define multiple distance metrics that take into account objects' extent. We have developed algorithms to discover two types of spatial association patterns in scientific data. We present experimental results to demonstrate the efficacy of our approach on real datasets drawn from the bioinformatic domain. We also highlight the importance of the discovered patterns by integrating the underlying domain knowledge.

1 Introduction

Advances in simulation and data collection techniques in different scientific domains such as bioinformatics, computational fluid dynamics, and computational molecular dynamics, have resulted in huge amounts of data. It is necessary to develop computational techniques to extract features from scientific datasets, for instance, vortices in fluid flow fields and substructures from molecular datasets. It is also very important to subsequently uncover meaningful relationships among extracted features. Such relationships can provide valuable information towards understanding or explaining the underlying scientific phenomenon. Many algorithms [Kramer et al., 2001; Jiang et al., 2003; Yang et al., 2004] have been proposed to extract features from these scientific datasets. In this paper we focus on the latter problem of finding relationships among features.

Recently, researchers have started to explore the relationships [Morimoto, 2001; Munro et al., 2003; Zhang et al., 2004] among features. Most of the previous techniques represent features by single points. However, this representation leads to loss of important information. The shape, size, and orientation of a feature in scientific datasets are very important. A simple solution to this problem is to use Minimum Bounding Boxes (MBB). However, MBBs are not well-suited for every domain. For example, ellipsoids are

more suitable for capturing the shape of vortices in fluid flow datasets [Sadarjoen et al., 1998]. Alternatively, defect structures in materials may require irregular shape descriptors [Mehta et al., 2004]. In this work, we propose to use different shape descriptors for features from different domains.

Frequently recurring relationships among features across different datasets can guide the domain expert to find useful knowledge. Such relationships are especially useful in bioinformatics. One important issue in bioinformatics is to identify structurally similar proteins. To address this issue, one can first discover non-local patterns that frequently occur in proteins known to be structurally similar [Zaki, 2002; Yang et al., 2004]. Such patterns can then be used to indicate whether a new protein is potentially similar to known proteins. We propose algorithms that discover relationships across multiple maps. This is different from previous work on spatial association mining [Koperski and Han, 1995; Shekhar and Huang, 2001; Morimoto, 2001; Zhang et al., 2004], where features are located in the same dataset and represented as points.

We define Spatial Object Association Patterns (SOAP) to characterize spatial relationships among object types. Our algorithm finds two different types of SOAPS: *Star* and *Sequence* (Figure 2). They capture different aspects of neighborhood relationships among features. For example, in protein contact maps, the formation of *Star* SOAPS among non-local patterns indicates such patterns have a compact spatial relationship. The formation of *Sequence* SOAPS on the other hand indicates an extended spatial relationship among the involved patterns. Furthermore, by extracting SOAPS from contact maps that are associated with proteins in different protein classes, we can establish the associations between different types of SOAPS and protein classes. Such associations can help us to identify structural characteristics of different protein classes.

The rest of this document is organized as follows. In Section 2, we present the key ideas underpinning our work: (i) the notion of spatial feature representation using extents and shapes, (ii) the different object-oriented distance metrics, and (iii) the different association pattern types used to characterize spatial relationships. In Section 3, we detail the algorithms and efficient realizations of the key ideas. Section 4 describes the efficacy of our approach on datasets drawn from bioinformatics. Finally, we present our conclusions and outline directions of ongoing and future research in Section 5.

*This work is funded by NSF grants ITR-NGS ACI-0326386, CAREER IIS-0347662, and SOFTWARE ACI-0234273. All correspondence should be addressed to Srinivasan Parthasarathy at srini@cse.ohio-state.edu.

2 Background and Definitions

2.1 Spatial Feature Representation

We propose two basic shape representation schemes: parallelepiped (or parallelogram in 2-D) and ellipsoid (or ellipse in 2-D). As demonstrated in Section 4, parallelograms are more appropriate to capture the shape and extent of non-local structures in protein contact maps as opposed to MBBs. Whereas ellipsoids (or ellipses) are suitable for vortices in fluid flow data as mentioned earlier. Note that the first scheme subsumes the MBB representation, and the second scheme subsumes circles. For highly irregular-shaped features such as defect structures in materials, we plan to use sampled boundary points, known as landmarks [Rao and Suryawanshi, 1996].

As shown in Figures 1(a) and 1(b), the shape descriptors of a parallelogram and an ellipse can be described as $A_{parallel} = \langle (x, y), l_1, l_2, \theta, \phi \rangle$ and $A_{ellipse} = \langle (x, y), l_1, l_2, \theta \rangle$, respectively. These descriptors can also be extended to 3D.

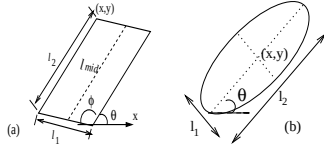


Figure 1: Object type: (a)Parallelogram (b)Ellipse

2.2 Dataset Representation

The dataset $\mathbb{D}$ consists of n features located in r maps ($r > 1$), denoted as $M = \{m_1, m_2, \dots, m_r\}$. The n features are categorized into l types, corresponding to l unique labels $\Sigma = \{c_1, c_2, \dots, c_l\}$. The categorization of features is governed by the underlying domain. A feature's geometric properties such as shape and size are captured by adopting one of the supported representation schemes. A feature f thus can be described as a vector $\langle mapID, location, A_{geo}, type \rangle$, where $type \in \Sigma$, $mapID \in M$ indicates the map where f occurs, $location$ identifies f 's position within the map, and $A_{geo} \in \{A_{parallel}, A_{ellipse}\}$ captures the shape of f .

Note that in the rest of the paper, we refer to a feature corresponding to the above vector as a *spatial object*. We assume the following order among the r maps: $m_1 < m_2 < \dots < m_r$. If the r maps correspond to r snapshots, they are ordered temporally. If they are from different datasets (e.g., different protein contact maps), the order among them is imposed by arbitrarily assigning each map a unique ID. Furthermore, the lexicographic order among the l feature types is imposed as follows: $c_1 < c_2 < \dots < c_l$.

2.3 Object-oriented Distance Metrics

The proposed algorithms use the following metrics to measure the distance between two objects o_i and o_j located in the same map.

- **Point-Point distance:** This is simply the Euclidian distance between object centroids.
- **Line-Line distance:** If o_i and o_j are parallelepipeds (or parallelograms), we first identify the line segment between the midpoints of the top and bottom surfaces

(or sides) in each object, then compute the shortest distance between these two line segments as the line-line distance between o_i and o_j . We identify the top and bottom surfaces (or sides) by selecting a reference axis in the underlying Cartesian coordinate space, specifically, the z -coordinate in 3D and y -coordinate in 2D. If o_i and o_j are ellipsoids (or ellipses), the line-line distance is between the two major axes.

- **Boundary-Boundary distance:** This is the shortest distance between the boundaries of o_i and o_j . When o_i and o_j are represented as ellipses, the boundary-boundary distance is the shortest pair-wise distance between points sampled on the boundaries (or surfaces in 3D). The number of sampled points is user-specified.

Notice that the line-line and boundary-boundary metrics are able to take objects' geometric properties into account. The algorithms also support Hausdorff distance [Atallah, 1983]. Since this distance is not applicable to the applications described in this article, we do not discuss it here.

Two objects o_i and o_j have a *closeTo* relationship if the distance between them is $\leq \epsilon$, where ϵ is a user-specified parameter. Two objects are *neighbors* if they have a *closeTo* relationship. We also define the *isAbove* relationship between o_i and o_j . In a coordinate system, o_i is said to have a *isAbove* relationship with o_j , if the upper-left corner of o_i 's Minimum Bounding Box (MBB)¹, denoted as (x_i, y_i, z_i) , and the upper left corner of o_j 's MBB, denoted as (x_j, y_j, z_j) , meets the following condition: $(z_i > z_j) \vee [(z_i = z_j) \wedge ((y_i > y_j) \vee ((y_i = y_j) \wedge (x_i < x_j)))]$ in a 3-D map, or $(y_i > y_j) \vee [(y_i = y_j) \wedge (x_i < x_j)]$ in a 2-D map.

2.4 Spatial Object Association Pattern (SOAP)

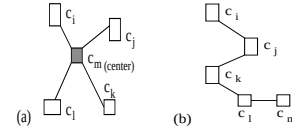


Figure 2: SOAP Types: (a)Star (b)Sequence

A *Spatial Object Association Pattern (SOAP)* of size k , denoted as k -SOAP, characterizes the *closeTo* relationships among k object types. In this article, we focus on two types of SOAPS: *Star* and *Sequence* (Figure 2). As discussed earlier, these two SOAP types can characterize different spatial relationships among objects. These two SOAP types can be abstracted as undirected graphs, where a node corresponds to an object-type $c_i \in \Sigma$, and an edge (c_i, c_j) indicates that c_i and c_j are required to have a *closeTo* relationship.

- **Star SOAPS** (Figure 2a) have a *center* object-type, which is required to have a *closeTo* relationship with all the other object-types in the same SOAP.
- **Sequence SOAPS** (Figure 2b) of size k $p = (c_{[i]} : i \in [1, k])$ satisfy two constraints, where $c_{[i]}$ is the i^{th} element in p : (1) $closeTo(c_{[i]}, c_{[i+1]}) = true$ and (2) $isAbove(c_{[i]}, c_{[i+1]}) = true$, where $1 \leq i \leq k-1$. Sequence SOAPS are mainly motivated by our observation on protein contact maps, where non-local structures, i.e. features, tend to line up in a Sequence like manner.

¹For a 3D MBB, it is the upper-left corner of its top surface.

These two SOAP types can also be represented as lists. Let c_{ctr} be the center of a star k -SOAP p , and $\{c_{[i]}: i \in [1, k-1]\}$ be the other $k-1$ object-types in p , where $c_{[1]} \leq \dots \leq c_{[k-1]}$. SOAP p can then be described by the list $p=(c_{ctr}, c_{[1]}, \dots, c_{[k-1]})$, where $closeTo(c_{ctr}, c_{[i]})=true$ ($i \in [1, k-1]$). Whereas elements in a sequence SOAP correspond to a list by definition and cannot be forced into lexicographical order. For instance, the two sequence SOAPS $(A B C)$ and $(A C B)$ are different.

A SOAP is *autocorrelated* if an object-type occurs multiple times. For example, (c_1, c_1, c_2) is an autocorrelated 3-SOAP, where c_1 occurs twice. An *instance* of a SOAP p is the set of objects that meet all the requirements specified by p , including those on object-types and *closeTo* (or *isAbove*) relationships.

We define two measures—*support* and *realization*—to characterize the importance of a SOAP. The support of a SOAP p is the number of maps in the dataset where p occurs. Assume $support(p)=s$, let n_i be the number of p 's instances in the i^{th} map where p appears, $realization(p)=\min\{n_i\}$. A pattern p is *frequent* if $support(p) \geq minSupp$, and *prevalent* if $realization(p) \geq minRealization$, where $minSupp$ and $minRealization$ are user-specified parameters.

3 Algorithms

3.1 Data Organization

We organize $\mathbb{D}$ in the following manner. The n objects are first grouped into l partitions, where each partition is composed of objects of the same type. Within each partition, objects are ordered by their map IDs and locations in a map. This data organization is analogous to the format used for association rule mining [Zaki et al., 1997]. Note that each object can be uniquely identified by combining the following information: *mapID*, *label*, and *locationID* in the map.

3.2 Equivalence Classes

Based on the list-based SOAP representation (Section 2.4), we organize SOAPS into equivalence classes. A *k-equivalence class*, denoted as *k-EquiClass*, is defined as the set of k -SOAPS that (i) are of the same SOAP type, and (ii) have the same prefix, where the *prefix* of a k -SOAP is its first $k-1$ elements. By using equivalence classes, our mining algorithms only need to compute the *closeTo* or *isAbove* relationships between objects once. The equivalence classes also help to improve the memory locality of the algorithms, which result in significant performance gains. Moreover, equivalence classes also enable the algorithms to smoothly scale to large datasets [Zaki et al., 1997].

The next two sections describe the algorithms that discover *Star* and *Sequence* SOAPS. For each frequent SOAP, the algorithms store information about all of its instances. Such information allows the algorithms to locate every object involved in an instance.

3.3 Mining Star SOAPS

Figure 3 outlines the algorithm that discovers *star* SOAPS. The first step generates frequent 1-SOAPS (line 1). For each object-type c_i , the procedure *gen1SOAP* counts the number of maps that contain at least one object of type c_i . If the count

$\geq minSupp$, then $p_1=(c_i)$ is a frequent 1-SOAP. The set of all frequent 1-SOAPS is denoted by P_1 .

The next step, *gen2EquiClass*, discovers 2-SOAPS and organizes them into 2-EquiClasses (line 2). The pseudo-code of *gen2EquiClass* is described in Figure 4. A 2-EquiClass is generated for each frequent 1-SOAP (Figure 4:line 2). The 2-EquiClass of 1-SOAP $(c_i) \in P_1$, denoted by E_{2,c_i} , contains frequent 2-SOAPS in the form of $(c_i, *)$. To generate E_{2,c_i} , the procedure considers the following 2-SOAPS as candidates: (c_i, c_j) , where $(c_j) \in P_1$ (Figure 4:line 3). For each candidate 2-SOAP $p_2=(c_i, c_j)$, the procedure identifies all the maps where p_2 occurs (Figure 4:lines 6-11). An instance of $p_2=(c_i, c_j)$ is an object pair (o_i, o_j) , where $(o_i.type=c_i) \wedge (o_j.type=c_j) \wedge (distance(o_i, o_j, distType) \leq \epsilon)$. If p_2 occurs in $\geq minSupp$ maps, then it is frequent and is added to E_{2,c_i} (Figure 4:line 12).

```

Algorithm mine_starSOAP( $\mathbb{D}, minSupp, minRealization, distType, \epsilon$ )
1.  $P_1 \leftarrow gen1SOAP()$ ; // 1-SOAPS
2.  $EC_2 \leftarrow gen2EquiClass(\mathbb{D}, P_1, star, parList)$ ; //parList: parameters
3.  $k \leftarrow 2$ ;
4. while (1){
5.    $EC_{k+1} \leftarrow \emptyset$ ; //the set of (k+1)-EquiClasses
6.   foreach  $k$ -EquiClass  $E_k \in EC_k$ 
7.     foreach  $k$ -SOAP  $p_{k,i} \in E_k$ 
8.        $E_{k+1} \leftarrow \emptyset$ ; // the prefix of  $E_{k+1}$  is  $p_{k,i}$ 
9.       foreach  $k$ -SOAP  $(p_{k,j} \in E_k) \wedge (i \leq j)$ 
10.         $p_{k+1} \leftarrow append(p_{k,i}, lastElement(p_{k,j}))$ ;
11.         $M_{p_{k+1}} \leftarrow \{m_i : m_i \text{ contains } p_{k,i} \text{ and } p_{k,j}\}$ ;
12.        foreach  $m_i \in M_{p_{k+1}}$ 
13.          if (countStarInstances( $m_i, p_{k,i}, p_{k,j}$ )  $\geq 1$ ) mapCnt++;
14.          if (mapCnt  $\geq minSupp$ )  $E_{k+1} \leftarrow E_{k+1} \cup \{p_{k+1}\}$ ;
15.        if ( $E_{k+1} \neq \emptyset$ )  $EC_{k+1} \leftarrow EC_{k+1} \cup E_{k+1}$ ;
16.        if ( $minRealization > 1$ ) markPrevFreqSOAPS( $E_{k+1}$ );
17.        if ( $EC_{k+1} = \emptyset$ ) return; //terminate
18.        k++; //increase SOAP size }while(1)

```

Figure 3: Mining Star SOAPS

The algorithm next discovers SOAPS of size > 2 (Figure 3:lines 4-17). Two k -SOAPS in the same k -EquiClass are combined to construct a candidate $(k+1)$ -SOAP. For each candidate $(k+1)$ -SOAP p_{k+1} derived by appending the last element of $p_{k,j}$ to $p_{k,i}$ (Figure 3: line 10), the algorithm identifies all the maps where p_{k+1} occurs (Figure 3:lines 11-14). (The i^{th} k -SOAP in a k -EquiClass is denoted as $p_{k,i}$.) The procedure *countStarInstances*($m_i, p_{k,i}, p_{k,j}$) (Figure 3:line 13) computes the instances of p_{k+1} in map m_i by combining instances of $p_{k,i}$ and $p_{k,j}$. Two instances from $p_{k,i}$ and $p_{k,j}$ are joined to produce a p_{k+1} instance if they have the same first $k-1$ objects and different last object. SOAPS with the same prefix are organized into one equivalence class as they are being generated (see Figure 3:lines 5, 8, 14, 15). The mining process stops when all the frequent SOAPS have been discovered (Figure 3:line 17).

To discover frequent SOAPS, the algorithm only needs to consider a SOAP's presence in a map (Figure 3:line 13). Hence, some of the discovered frequent SOAPS may not be prevalent if $minRealization > 1$. In this case, the procedure *markPrevFreqSOAPS* is called to identify the SOAPS that are both prevalent and frequent (Figure 3:line 16). It is necessary to keep SOAPS that are frequent but not prevalent. We explain this by a simple example. Let $\{a_1, b_1, b_2\}$ be the three neighboring objects in a map and $minRealization=2$. To derive the two instances of SOAP (a, b) , (a_1, b_1) and (a_1, b_2) ,

```

Algorithm gen2EquiClass ( $\mathbb{D}, P_1, soapType, parList$ ) //  $P_1$ :freq. 1-SOAPs
1.  $EC_2 \leftarrow \emptyset$ ; //the set of 2-EquiClasses
2. foreach ( $c_i$ )  $\in P_1$ 
3.    $E_{2,c_i} \leftarrow \emptyset$  //2-EquiClass with prefix of  $c_i$ ;
4.   foreach ( $c_j$ )  $\in P_1$ 
5.      $p_2 \leftarrow (c_i, c_j)$ ; //a candidate
6.      $M_{p_2} \leftarrow \{m_i : (m_i \text{ contains } c_i \text{ and } c_j)\}$ ;
7.     foreach  $m_i \in M_{p_2}$ 
8.       foreach ( $o_i, o_j$ )  $\in m_i : (o_i.label = c_i) \wedge (o_j.label = c_j)$ 
9.         if ( $(soapType=Sequence) \wedge (\neg isAbove(o_i, o_j))$ ) continue;
10.        if  $distance(o_i, o_j, distType) \leq \epsilon$  addInstance( $p_2, (o_i, o_j)$ );
11.        if ( $countInst(p_2, m_i) \geq 1$ ) mapCnt++;
12.        if ( $mapCnt \geq minSupp$ )  $E_{2,c_i} \leftarrow E_{2,c_i} \cup \{p_2\}$ ;
13. if ( $E_{2,c_i} \neq \emptyset$ )  $EC_2 \leftarrow EC_2 \cup E_{2,c_i}$ ;

```

Figure 4: 2-EquiClass Generation

the 1-SOAP (a) must be maintained even though its realization is 1 (< 2) in the map.

Correctness: It is straightforward to show that the algorithm discovers all the frequent 1-SOAPs and 2-SOAPs. Thus to prove the algorithm is correct, we only need to show that every frequent k -SOAP ($k > 2$) will be considered as a candidate. Assume $p_k = (c_{[1]}, \dots, c_{[k-2]}, c_{[k-1]}, c_{[k]})$ is frequent, then the two $(k-1)$ -SOAPs $p_{k-1,i} = (c_{[1]}, \dots, c_{[k-2]}, c_{[k-1]})$ and $p_{k-1,j} = (c_{[1]}, \dots, c_{[k-2]}, c_{[k]})$ must also be frequent and in the same equivalence class. Thus the algorithm will consider p_k as a candidate. It is trivial to show that the procedure *countStarInstances* identifies all the instances of a candidate SOAP. $\square$

3.4 Mining Sequence SOAPs

The pseudo-code is given in Figure 5. Unlike mining Star SOAPs, which uses two k -SOAPs in the same equivalence class to generate a candidate $(k+1)$ -SOAP ($k \geq 2$), the algorithm joins one k -SOAP and one 2-SOAP.

The first two steps (lines 1-2) discover all the frequent 1-SOAPs and 2-SOAPs. The *isAbove* relationship is checked by the procedure *isAbove*(o_i, o_j) (Figure 4:line 9). For each k -SOAP $p_k = (c_{[1]}, \dots, c_{[k]})$, the algorithm first locates the 2-EquiClass $E_{2,c_{[k]}}$, in which every 2-SOAP is in the form $(c_{[k]}, c_{[j]})$: *closeTo*($c_{[k]}, c_{[j]}) \wedge isAbove(c_{[k]}, c_{[j]})$ (line 7). A set of candidate $(k+1)$ -SOAPs are then generated by combining p_k with each 2-SOAP in $E_{2,c_{[k]}}$ (lines 8-9). Same as mining Star SOAPs, a candidate $(k+1)$ -SOAP p_{k+1} is frequent if it appears in $\geq minSupp$ maps (lines 10-13). The procedure *countSeqInstances*($m_i, p_{k,i}, p_{2,j}$) (line 12) identifies instances of p_{k+1} in map m_i , where p_{k+1} is a candidate SOAP based on $p_{k,i}$ and $p_{2,j}$. Two instances $I_{k,i}$ and $I_{2,j}$, from $p_{k,i}$ and $p_{2,j}$ respectively, are combined to produce an instance of p_{k+1} if the last object in $I_{k,i}$ is the same as the first object in $I_{2,j}$. For the same reason explained before, the algorithm calls the procedure *markPrevFreqSOAPs* to label SOAPs being both prevalent and frequent if *minRealization* > 1 (line 15). The algorithm stops when no more SOAPs can be discovered (line 14).

Correctness: For each frequent sequence k -SOAP $p_k = (c_{[1]}, \dots, c_{[k-1]}, c_{[k]})$, the following two SOAPs must also be frequent: $p_{k-1} = (c_{[1]}, \dots, c_{[k-1]})$ and $p_2 = (c_{[k-1]}, c_{[k]})$. Thus, p_k will be considered as a candidate and be discovered. It is trivial to show that the procedure *countSeqInstances* identifies all the instances of a candidate SOAP. $\square$

```

Algorithm mine_sequenceSOAP ( $\mathbb{D}, minSupp, minRealization, distType, \epsilon$ )
1.  $P_1 \leftarrow gen1SOAP()$ ;
2.  $EC_2$  (or  $P_2$ )  $\leftarrow gen2EquiClass(\mathbb{D}, P_1, Sequence, parList)$ ; //parList:parameters
3.  $k \leftarrow 2$ ;
4. while (1){
5.    $P_{k+1} \leftarrow \emptyset$ ; //initialize the set of  $(k+1)$ -SOAPs;
6.   foreach SOAP  $p_{k,i} = (c_{[0]}, \dots, c_{[k-1]}) \in P_k$ 
7.      $E_{2,c_{[k]}} \leftarrow E_{2,c_{[k]}} \in EC_2 \wedge (E_{2,c_{[k]}}.prefix = c_{[k]})$ ;
8.     foreach 2-SOAP  $p_{2,j} \in E_{2,c_{[k]}}$ 
9.        $p_{k+1} \leftarrow append(p_{k,i}, lastElement(p_{2,j}))$ ;
10.       $M_{p_{k+1}} \leftarrow \{m_i : m_i \text{ contains both } p_{k,i} \text{ and } p_{2,j}\}$ ;
11.      foreach  $m_i \in M_{p_{k+1}}$ 
12.        if ( $countSeqInstances(m_i, p_{k,i}, p_{2,j}) \geq 1$ ) mapCnt++;
13.        if ( $mapCnt \geq minSupp$ )  $P_{k+1} \leftarrow P_{k+1} \cup \{p_{k+1}\}$ ;
14.    if ( $P_{k+1} = \emptyset$ ) return; //terminate the process;
15.    if ( $minRealization > 1$ ) markPrevFreqSOAPs( $P_{k+1}$ );
16.     $k++$ ; //increase SOAP size}/while(1)

```

Figure 5: Mining Sequence SOAPs

Other SOAP types can also be defined. For instance, we can define *Clique* SOAPs, which require every pair of objects in the same SOAP have a *closeTo* relationship. We can also define SOAPs that involve other types of spatial relations such as topological relations.

4 Experimental Evaluation

In this section, we evaluate the algorithms on a protein contact map dataset. We start with a brief description on the dataset generation process and the domain-specific usefulness measurement adopted in this work. We then present results to analyze the impact from different distance metrics and SOAP types. Finally, we present performance characterization of the algorithms.

4.1 Data Preprocessing and Representation

We first generate contact maps for 8,732 proteins taken from the Protein Data Bank [Berman et al., 2000]. For a protein with N amino acids, its contact map C is a $N \times N$ binary matrix. The position $C(i, j)$ is set to 1 if the distance between the i^{th} and j^{th} residues is less than a threshold and 0 otherwise [Vendruscolo and Domany, 1997]. We use 6Å as the threshold as suggested in the literature [Zaki, 2002]. We then extract features in contact maps. A *contact map feature* is composed of a set of positions, where each position and at least one of its eight neighbors contain a '1' (see Figure 6 for examples). We apply a simple region growing approach to extract features in contact maps [Yang et al., 2004]. We then use an entropy-based clustering algorithm to cluster features into l groups (or classes) [Cheng et al., 1999]. Features in a class have similar geometric properties such as shape and extent.

A total of 1,009,755 features are extracted from the 8,732 contact maps. These features are clustered into 28 classes. The average number of features in each map is about 115. Many of these features correspond to well-known protein secondary structures and are also validated by domain experts. We represent each feature by its minimum bounding parallelogram and label the parallelogram by the feature's class ID.

4.2 Domain Specific Usefulness Measurement

The structure of a protein often provides information about its functionality. Thus, one can predict a protein's function based on the function of other structurally similar proteins.

Frequent SOAPs in contact maps characterize proteins' structure. They can be used to generate signatures for different proteins. For instance, the SOAP (7 7 7 22), which is automatically discovered by our algorithms as both *Star* and *Sequence*, can be used to identify the following *structurally and functionally similar* β -proteins: 1h3t, 1h3u and 1h3v (IDs from PDB). (See Figures 6(a) and 6(b) for an illustration of these SOAPs.)

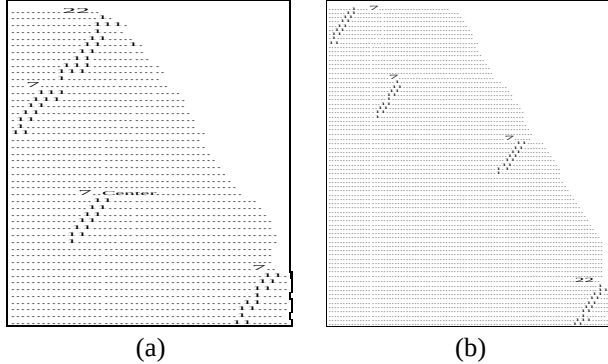


Figure 6: SOAP (7 7 7 22) in proteins 1h3u, 1h3t & 1h3v as: (a) Star and (b) Sequence

However, a SOAP is unlikely to add any value to the signature, if it is randomly associated with proteins of different classes. To prune away such SOAPS, we use an entropy-based approach. For a given SOAP, we first identify its associated proteins. We then compute the SOAP's entropy by integrating the proteins' lineage information from the database of Structural Classification of Proteins (SCOP)². A protein's SCOP lineage is organized into 6 hierarchical levels according to its structure. We look at the first two levels: L_1 :*class*, and L_2 :*fold*. L_1 consists of 11 classes such as α -protein, β -protein, and small protein. L_2 further divides proteins into sub-classes based on proteins' folding structures. Let N be the number of proteins whose contact maps contain the SOAP p , and n_i be the number of proteins among these N proteins in the i^{th} class, the entropy of p at the first SCOP level is then computed as: $H(p) = \sum_{i=1}^{11} -(n_i/N) \times \log_2(n_i/N)$ [Shannon, 2001]. $H(p)$ essentially measures how well p 's associated proteins distribute among different proteins classes. In our experiments, we observe that SOAPS with entropy ≤ 2.0 show good quality and can be used to generate signatures. This observation is also validated by domain experts.

For each SOAP with entropy ≤ 2.0 , we next identify the protein class that dominates its associated proteins. For example, the dominating protein class of the clique SOAP (5 7 7 22) is β -protein, as 191 out of its 215 associated proteins are β -proteins. As suggested by domain experts, we would like to have many SOAPS for a certain protein class. A large number of SOAPS not only cover a wide range of proteins, but can also be used to identify structurally similar proteins from different aspects. This criterion is used to evaluate the discovered SOAPS. Note that even if two SOAPS have the same dominating protein class, their associated proteins can be very different.

²<http://scop.mrc-lmb.cam.ac.uk/scop/>

4.3 Impact of Distance Metrics

Due to lack of space, we only report the results produced from the following parameter setting: $minSupp=1\%$, $minRealization=1$, and $\epsilon=55\text{\AA}$.

Figure 7 shows the impact of the three distance metrics on different SOAP types. One can observe that the Line-Line (L-L) distance generates the most number of SOAPS for both SOAP types, whereas the Point-Point (P-P) distance the least. This is expected as the P-P distance does not consider the shape and size of spatial objects, in contrast with the other two. One may argue that more SOAPS can be identified by increasing the distance threshold in the case of P-P distance. However, it is very difficult to find an appropriate threshold value, as the size of a feature, i.e., the number of bit-1 positions, varies drastically, from 2 up to several thousand.

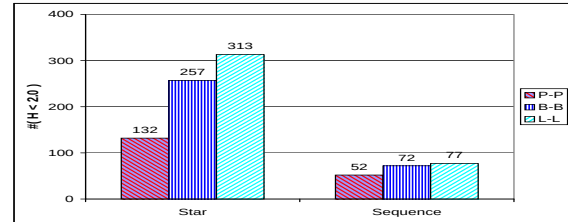


Figure 7: Distance metric vs #SOAPS discovered

The advantage of using L-L and B-B distance metrics becomes more significant when looking at the SOAPS' dominating protein classes. Tables 1-3 summarize the number of SOAPS in major protein classes. One can see that, if the P-P distance is used (Table 1), only 31 SOAPS are discovered in small proteins as against 133 SOAPS in β -proteins. On the other hand, SOAPS based on L-L or B-B distance show a relatively more balanced behavior (Tables 2-3). Compared with P-P distance, much more SOAPS in β -proteins or small proteins are discovered. For instance, in the L-L case (Table 2), there are 180 and 203 SOAPS are identified in β -proteins and small proteins respectively. Also the B-B distance produces the most number (14) of SOAPS for α -proteins (Table 3).

The differences exhibited by SOAPS from different distance metrics show that it is important for our algorithms to support multiple distance metrics, especially metrics which consider objects' extent.

Type	#(α)	#(β)	#(small)	#(peptide)
Star	2	103	19	7
Sequence	4	30	12	4
Total	6	133	31	11

Table 1: #SOAPS in major protein groups, distType=P-P

Type	#(α)	#(β)	#(small)	#(peptide)
Star	1	133	167	12
Sequence	4	37	36	6
Total	5	170	203	18

Table 2: #SOAPS in major protein groups, distType=L-L

4.4 Impact of SOAP Types

As we move down to the second SCOP lineage level, L_2 :*fold*, we discover that different types of SOAPS can actually distinguish different protein folding structures. Proteins in the same class (e.g., β -protein) are further classified

Type	#(α)	#(β)	#(small)	#(peptide)
Star	8	145	91	13
Seq.	6	40	25	6
Total	14	185	116	19

Table 3: #SOAPs in major protein groups, distType=B-B

into structurally similar sub-groups according to their folding structures. Table 4 lists the β -protein folds that are distinguished by each SOAP type, where SOAPs are generated based on the L-L distance. The folds in bold are those that are associated with only one SOAP type. Whereas other folds in the table are distinguished by two or more SOAP types. Folds in other protein classes show a similar trend. For example, the α -protein fold "Cyclin-like" is only associated with sequence SOAPs.

The above results can potentially help domain experts address some important biological issues, for instance, predicting a protein's function based on the SOAPs contained in its contact map.

Star	Immunoglobulin-like beta-sandwich Concanavalin A-like lectins/glucanases Trypsin-like serine proteases Cupredoxin-like Acid proteases Cysteine proteinases
Sequence	Immunoglobulin-like beta-sandwich Concanavalin A-like lectins/glucanases Trypsin-like serine proteases Lipocalins Nucleoplasmin-like/VP

Table 4: List of β -protein folds associated with each SOAP type, distType=L-L

Type	minSupp=2%	minSupp=1%
Star	15.54	21.53
Sequence	4.83	7.79

Table 5: Running Time (in seconds) (distType=L-L)

4.5 Running Time

Table 5 shows the time taken to discover the two types of SOAPs at two *minSupp* values. All the experiments were carried out on a Pentium 1.7GHz computer of 256MB main memory. To discover SOAPs in 8,732 protein contact maps containing 1,009,755 objects, the algorithms take about 20 seconds when *minSupp*=2% and about 30 seconds when *minSupp*=1%. Although running time increases as *minSupp* decreases, the algorithm scales very well. We attribute this good performance to the use of equivalence classes.

5 Conclusion and Ongoing Work

In this paper, we present a general framework to uncover two types of spatial association patterns among features in scientific data. The framework represents features as spatial objects instead of points. It also supports multiple distance metrics. Empirical results on protein contact maps show that the framework is both efficient and scalable. Furthermore, the discovered SOAPs are meaningful and can potentially be used to address important biological issues.

The SOAP mining problem in this article shares some similarity with frequent subgraph mining [Yan and Han, 2002]. In order to apply the conventional graph mining algorithms for

SOAP mining, the notion of nodes needs to be extended to integrate spatial properties such as location and shape. Also, the notion of edge needs to be modified to reflect different spatial relationships such as *closeTo*.

We are currently extending the framework in several directions. First, we are examining other shape representation schemes such as the landmark-based approach for highly irregularly shaped features. Second, we are evaluating other types of object association patterns, for example, clique SOAPs, where each object has a *closeTo* relationship with every other object in the same SOAP. Third, we are interested in evaluating the algorithms on datasets from other scientific domains, including molecular dynamics and fluid flow dynamics. Some of our ongoing work is reported in a technical report [Yang *et al.*, 2005]. Finally, we are investigating potential approaches towards association based spatio-temporal reasoning.

Acknowledgments: We thank D. Polshakov and K. Marsolo for helping us validate the experimental results, and thank Dr. J. Wilkins and Dr. R. Machiraju for valuable comments on the initial ideas.

References

- [Atallah, 1983] M. J. Atallah. A linear time algorithm for the hausdorff distance between convex polygons. *Information Processing Letter*, 17(207-209), 1983.
- [Berman *et al.*, 2000] H. Berman *et al.* The protein data bank. *Nucleic Acids Research*, 28(235-242), 2000.
- [Burdick *et al.*, 2001] C. Burdick *et al.* Mafia: A maximal frequent itemset algorithm for transactional databases. In *ICDM*, 01.
- [Cheng *et al.*, 1999] C. Cheng *et al.* Entropy-based subspace clustering for mining numerical data. In *SIGKDD*, 1999.
- [Jiang *et al.*, 2003] M. Jiang *et al.* Feature mining paradigms for scientific data. In *SIAM*, 2003.
- [Koperski and Han, 1995] K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. In *SSD*, 1995.
- [Kramer *et al.*, 2001] Stefan Kramer *et al.* Molecular feature mining in HIV data. In *SIGKDD*, 2001.
- [Mehta *et al.*, 2004] S. Mehta *et al.* Detection and visualization of anomalous structures in molecular dynamics simulation data. In *IEEE VIS*, 2004.
- [Morimoto, 2001] Y. Morimoto. Mining frequent neighboring class sets in spatial databases. In *SIGKDD*, 2001.
- [Munro *et al.*, 2003] R. Munro *et al.* Complex spatial relationships. In *ICDM*, 2003.
- [Rao and Suryawanshi, 1996] C. R. Rao and S. Suryawanshi. Statistical analysis of shape of objects based on landmark data. *Proc National Academy of Science, USA*, 1996.
- [Richie *et al.*, 2001] D.A. Richie *et al.* Real-time multiresolution analysis for accelerated molecular dynamics simulations. In *American Physics Society March Meeting*, 2001.
- [Sadarjoen *et al.*, 1998] A. Sadarjoen *et al.* Selective visualization of vortices in hydrodynamic flows. In *IEEE VIS*, 1998.
- [Shannon, 2001] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mobile Computing Community Review*, 5(1):3-55, 2001.
- [Shekhar and Huang, 2001] S. Shekhar and Y. Huang. Discovering spatial co-location patterns: A summary of results. *Lecture Notes in Computer Science*, 2001.
- [Vendruscolo and Domany, 1997] M. Vendruscolo and E. Domany. Recovery of protein folding from contact maps. *Folding and Design*, 2(5):295-306, 1997.
- [Yan and Han, 2002] X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In *ICDM*, 2002.
- [Yang *et al.*, 2004] H. Yang *et al.* Discovering spatial relationships between approximately equivalent patterns. In *BIODDD*, 2004.
- [Yang *et al.*, 2005] H. Yang, S. Mehta, and S. Parthasarathy. A generalized framework for mining spatio-temporal patterns in scientific data. In *Technical Report OSU-CISRC-5/05-TR14, Ohio State University*, 2005.
- [Zaki *et al.*, 1997] M.J. Zaki *et al.* New algorithms for fast discovery of association rules. In *Technical Report TR651, Rensselaer Polytechnic Institute*, 1997.
- [Zaki, 2002] M.J. Zaki. Mining protein contact maps. In *BIODDD*, 2002.
- [Zhang *et al.*, 2004] X. Zhang *et al.* Fast mining of spatial collocations. In *SIGKDD*, 2004.

Semi-Supervised Regression with Co-Training

Zhi-Hua Zhou and Ming Li

National Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{zhouzh, lim}@lamda.nju.edu.cn

Abstract

In many practical machine learning and data mining applications, unlabeled training examples are readily available but labeled ones are fairly expensive to obtain. Therefore, semi-supervised learning algorithms such as *co-training* have attracted much attention. Previous research mainly focuses on semi-supervised classification. In this paper, a co-training style semi-supervised regression algorithm, i.e. COREG, is proposed. This algorithm uses two k -nearest neighbor regressors with different distance metrics, each of which labels the unlabeled data for the other regressor where the labeling confidence is estimated through consulting the influence of the labeling of unlabeled examples on the labeled ones. Experiments show that COREG can effectively exploit unlabeled data to improve regression estimates.

1 Introduction

In many practical machine learning and data mining applications such as web user profile analysis, unlabeled training examples are readily available but labeled ones are fairly expensive to obtain because they require human effort. Therefore, semi-supervised learning methods that exploit unlabeled examples in addition to labeled ones have attracted much attention.

Many current semi-supervised learning methods use a generative model for the classifier and employ Expectation-Maximization (EM) [Dempster *et al.*, 1977] to model the label estimation or parameter estimation process. For example, mixture of Gaussians [Shahshahani and Landgrebe, 1994], mixture of experts [Miller and Uyar, 1997], and naive Bayes [Nigam *et al.*, 2000] have been respectively used as the generative model, while EM is used to combine labeled and unlabeled data for classification. There are also many other methods such as using transductive inference for support vector machines to optimize performance on a specific test set [Joachims, 1999], constructing a graph on the examples such that the minimum cut on the graph yields an optimal labeling of the unlabeled examples according to certain optimization functions [Blum and Chawla, 2001], etc.

A prominent achievement in this area is the co-training paradigm proposed by Blum and Mitchell [1998], which trains two classifiers separately on two *sufficient and redundant views*, i.e. two attribute sets each of which is sufficient for learning and conditionally independent to the other given the class label, and uses the predictions of each classifier on unlabeled examples to augment the training set of the other.

Dasgupta *et al.* [2002] have shown that when the requirement of sufficient and redundant views is met, the co-trained classifiers could make few generalization errors by maximizing their agreement over the unlabeled data. Unfortunately, such a requirement can hardly be met in most scenarios. Goldman and Zhou [2000] proposed an algorithm which does not exploit attribute partition. This algorithm requires using two different supervised learning algorithms that partition the instance space into a set of equivalence classes, and employs cross validation technique to determine how to label the unlabeled examples and how to produce the final hypothesis. Although the requirement of sufficient and redundant views is quite strict, the co-training paradigm has already been used in many domains such as statistical parsing and noun phrase identification [Hwa *et al.*, 2003][Pierce and Cardie, 2001][Sarkar, 2001][Steedman *et al.*, 2003].

It is noteworthy that previous research mainly focuses on classification while regression remains almost untouched. In this paper, a co-training style semi-supervised regression algorithm named COREG, i.e. CO-training REGressors, is proposed. This algorithm employs two k -nearest neighbor (k NN) regressors, each of which labels the unlabeled data for the other during the learning process. In order to choose appropriate unlabeled examples to label, COREG estimates the labeling confidence through consulting the influence of the labeling of unlabeled examples on the labeled examples. The final prediction is made by averaging the regression estimates generated by both regressors. Since COREG utilizes different distance metrics instead of requiring sufficient and redundant views, its applicability is broad. Moreover, experimental results show that this algorithm can effectively exploit unlabeled data to improve regression estimates.

The rest of this paper is organized as follows. Section 2 proposes the COREG algorithm. Section 3 presents an analysis on the algorithm. Section 4 reports on the experimental results. Finally, Section 5 concludes and raises several issues for future work.

2 COREG

Let $L = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_{|L|}, \mathbf{y}_{|L|})\}$ denote the labeled example set, where $\mathbf{x}_i$ is the i -th instance described by d attributes, $\mathbf{y}_i$ is its real-valued label, i.e. its expected real-valued output, and $|L|$ is the number of labeled examples; let U denote the unlabeled data set, where the instances are also described by the d attributes, whose real-valued labels are unknown, and $|U|$ is the number of unlabeled examples.

Two regressors, i.e. h_1 and h_2 , are generated from L , each of which is then refined with the help of unlabeled examples that are labeled by the latest version of the other regressor. Here the k NN regressor [Dasarathy, 1991] is used as the base learner to instantiate h_1 and h_2 , which labels a new instance through averaging the real-valued labels of its k -nearest neighboring examples.

The use of k NN regressor as the base learner is due to the following considerations. First, the regressors will be refined in each of many learning iterations. If neural networks or regression trees are used, then in each iteration the regressors have to be re-trained with the labeled examples in addition to the newly labeled ones, the computational load of which will be quite heavy. Since k NN is a lazy learning method which does not hold a separate training phase, the refinement of the k NN regressors can be efficiently realized. Second, in order to choose appropriate unlabeled examples to label, the labeling confidence should be estimated. In COREG the estimation utilizes the neighboring properties of the training examples, which can be easily coupled with the k NN regressors.

It is noteworthy that the initial regressors should be diverse because if they are identical, then for either regressor, the unlabeled examples labeled by the other regressor may be the same as these labeled by the regressor for itself. Thus, the algorithm degenerates to *self-training* [Nigam and Ghani, 2000] with a single learner. In the standard setting of co-training, the use of sufficient and redundant views enables the learners be different. Previous research has also shown that even when there is no natural attribute partitions, if there are sufficient redundancy among the attributes then a fairly reasonable attribute partition will enable co-training to exhibit advantages [Nigam and Ghani, 2000]. While in the extended co-training algorithm which does not require sufficient and redundant views, the diversity among the learners is achieved through using different learning algorithms [Goldman and Zhou, 2000]. Since COREG does not assume sufficient and redundant views and different learning algorithms, the diversity of the regressors should be sought from other channels.

Here the diversity is achieved through utilizing different distance metrics. In fact, a key of k NN learner is how to determine the distances between different instances. The Minkowsky distance shown in Eq. 1 is usually used for this purpose. Note that different concrete distance metrics can be generated through setting different values to the distance order, p . Roughly speaking, the smaller the order, the more robust the resulting distance metric to data variations; while the bigger the order, the more sensitive the resulting distance metric to data variations. Therefore, the vicinities identified for a given instance may be different using the Minkowsky distance with different orders. Thus, the k NN regressors h_1

and h_2 can be diverse through instantiating them with different p values. Such a setting can also bring another profit, that is, since it is usually difficult to decide which p value is better for the concerned task, the functions of these regressors may be somewhat complementary to be combined.

$$Minkowsky_p(\mathbf{x}_r, \mathbf{x}_s) = \left(\sum_{l=1}^d |\mathbf{x}_{r,l} - \mathbf{x}_{s,l}|^p \right)^{1/p} \quad (1)$$

In order to choose appropriate unlabeled examples to label, the labeling confidence should be estimated such that the most confidently labeled example can be identified. In classification this is relatively straightforward because when making classifications, many classifiers can also provide an estimated probability (or an approximation) for the classification, e.g. a Naive Bayes classifier returns the maximum posteriori hypothesis where the posterior probabilities can be used, a BP neural network classifier returns thresholded classification where the real-valued outputs can be used, etc. Therefore, the labeling confidence can be estimated through consulting the probabilities of the unlabeled examples being labeled to different classes. For example, suppose the probability of the instance a being classified to the classes c_1 and c_2 is 0.90 and 0.10, respectively, while that of the instance b is 0.60 and 0.40, respectively. Then the instance a is more confident to be labeled (to class c_1).

Unfortunately, in regression there is no such estimated probability that can be used directly. This is because in contrast to classification where the number of class labels to be predicted is finite, the possible predictions in regression are infinite. Therefore, a key of COREG is the mechanism for estimating the labeling confidence.

Heuristically, the most confidently labeled example of a regressor should be with such a property, i.e. the error of the regressor on the labeled example set should decrease the most if the most confidently labeled example is utilized. In other words, the most confidently labeled example should be the one which makes the regressor most *consistent* with the labeled example set. Thus, the mean squared error (MSE) of the regressor on the labeled example set can be evaluated first. Then, the MSE of the regressor utilizing the information provided by $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ can be evaluated on the labeled example set, where $\mathbf{x}_u$ is an unlabeled instance while $\hat{\mathbf{y}}_u$ is the real-valued label generated by the original regressor. Let Δ_u denote the result of subtracting the latter MSE from the former MSE. Note that the number of Δ_u to be estimated equals to the number of unlabeled examples. Finally, $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ associated with the biggest positive Δ_u can be regarded as the most confidently labeled example.

Since repeatedly measuring the MSE of the k NN regressor on the whole labeled example set in each iteration will be time-consuming, considering that k NN regressor mainly utilizes local information, COREG employs an approximation. That is, for each $\mathbf{x}_u$, COREG identifies its k -nearest neighboring labeled examples and uses them to compute the MSE. In detail, let Ω denote the set of k -nearest neighboring labeled examples of $\mathbf{x}_u$, then the most confidently labeled example is identified through maximizing the value of $\Delta_{\mathbf{x}_u}$ in Eq. 2, where h denotes the original regressor while h' denotes the

Table 1: Pseudo-code describing the COREG algorithm

ALGORITHM: COREG

INPUT: labeled example set L , unlabeled example set U ,
number of nearest neighbors k ,
maximum number of learning iterations T ,
distance orders p_1, p_2

PROCESS:

$L_1 \leftarrow L; L_2 \leftarrow L$
Create pool U' by randomly picking examples from U
 $h_1 \leftarrow kNN(L_1, k, p_1); h_2 \leftarrow kNN(L_2, k, p_2)$
Repeat for T rounds:
 for $j \in \{1, 2\}$ **do**
 for each $\mathbf{x}_u \in U'$ **do**
 $\hat{\mathbf{y}}_u \leftarrow h_j(\mathbf{x}_u)$
 $\Omega \leftarrow Neighbors(\mathbf{x}_u, k, L_j)$
 $h'_j \leftarrow kNN(L_j \cup \{(\mathbf{x}_u, \hat{\mathbf{y}}_u)\}, k, p_j)$
 $\Delta_{\mathbf{x}_u} \leftarrow \sum_{\mathbf{x}_i \in \Omega} ((\mathbf{y}_i - h_j(\mathbf{x}_i))^2 - (\mathbf{y}_i - h'_j(\mathbf{x}_i))^2)$
 end of for
 if there exists an $\Delta_{\mathbf{x}_u} > 0$
 then $\tilde{\mathbf{x}}_j \leftarrow \arg \max_{\mathbf{x}_u \in U'} \Delta_{\mathbf{x}_u}; \tilde{\mathbf{y}}_j \leftarrow h_j(\tilde{\mathbf{x}}_j)$
 $\pi_j \leftarrow \{(\tilde{\mathbf{x}}_j, \tilde{\mathbf{y}}_j)\}; U' \leftarrow U' - \pi_j$
 else $\pi_j \leftarrow \emptyset$
 end of for
 $L_1 \leftarrow L_1 \cup \pi_2; L_2 \leftarrow L_2 \cup \pi_1$
 if neither of L_1 and L_2 changes **then exit**
 else
 $h_1 \leftarrow kNN(L_1, k, p_1); h_2 \leftarrow kNN(L_2, k, p_2)$
 Replenish U' by randomly picking examples from U
 end of Repeat

OUTPUT: regressor $h^*(\mathbf{x}) \leftarrow \frac{1}{2} (h_1(\mathbf{x}) + h_2(\mathbf{x}))$

refined regressor which has utilized the information provided by $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$. Note that $\hat{\mathbf{y}}_u = h(\mathbf{x}_u)$.

$$\Delta_{\mathbf{x}_u} = \sum_{\mathbf{x}_i \in \Omega} ((\mathbf{y}_i - h(\mathbf{x}_i))^2 - (\mathbf{y}_i - h'(\mathbf{x}_i))^2) \quad (2)$$

The pseudo code of COREG is shown in Table 1, where the function $kNN(L_j, k, p_j)$ returns a kNN regressor on the labeled example set L_j , whose distance order is p_j . The learning process stops when the maximum number of learning iterations, i.e. T , is reached, or there is no unlabeled example which is capable of reducing the MSE of any of the regressors on the labeled example set. According to Blum and Mitchell [1998]’s suggestion, a pool of unlabeled examples smaller than U is used. Note that in each iteration the unlabeled example chosen by h_1 won’t be chosen by h_2 , which is an extra mechanism for encouraging the diversity of the regressors. Thus, even when h_1 and h_2 are similar, the examples they label for each other will still be different.

3 Analysis

This section attempts to analyze whether the learning process of COREG can use the unlabeled examples to improve the

regression estimates. In order to simplify the discussion, here the effect of the pool U' is not considered as in [Blum and Mitchell, 1998]. That is, the unlabeled examples are assumed as being picked from the unlabeled example set U directly.

In each learning iteration of COREG, for each unlabeled example $\mathbf{x}_u$, its k -nearest neighboring labeled examples are put into the set Ω . As mentioned before, the newly labeled example should make the regressor become more consistent with the labeled data set. Therefore, a criterion shown in Eq. 3 can be used to evaluate the goodness of $\mathbf{x}_u$, where h is the original regressor while h' is the one refined with $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$. If the value of Δ_u is positive, then utilizing $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is beneficial.

$$\Delta_u = \frac{1}{|L|} \sum_{\mathbf{x}_i \in L} (\mathbf{y}_i - h(\mathbf{x}_i))^2 - \frac{1}{|L|} \sum_{\mathbf{x}_i \in L} (\mathbf{y}_i - h'(\mathbf{x}_i))^2 \quad (3)$$

In the COREG algorithm, the unlabeled example which maximizes the value of $\Delta_{\mathbf{x}_u}$ is picked to be labeled. Therefore, the question is, whether the unlabeled example chosen according to the maximization of $\Delta_{\mathbf{x}_u}$ will result in a positive Δ_u value or not.

First, assume that $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is among the k -nearest neighbors of some examples in Ω , and is not among the k -nearest neighbors of any other examples in L . In this case, it is obvious that utilizing $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ will only change the regression estimates on the examples in Ω , therefore Eq. 3 becomes Eq. 4. Comparing Eqs. 2 with 4 it can be found that the maximization of $\Delta_{\mathbf{x}_u}$ also results in the maximization of Δ_u .

$$\Delta_u = \frac{1}{k} \sum_{\mathbf{x}_i \in \Omega} (\mathbf{y}_i - h(\mathbf{x}_i))^2 - \frac{1}{k} \sum_{\mathbf{x}_i \in \Omega} (\mathbf{y}_i - h'(\mathbf{x}_i))^2 \quad (4)$$

Second, assume that $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is not among the k -nearest neighbors of any example in Ω . In this case, the value of $\Delta_{\mathbf{x}_u}$ is zero, therefore $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ won’t be chosen in COREG.

Third, assume that $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ is among the k -nearest neighbors of some examples in Ω as well as some examples in $L - \Omega$, and assume these examples in $L - \Omega$ are $(\mathbf{x}'_1, \mathbf{y}'_1), \dots, (\mathbf{x}'_m, \mathbf{y}'_m)$. Then Eq. 3 becomes Eq. 5.

$$\Delta_u = \frac{1}{k} \sum_{\mathbf{x}_i \in \Omega} ((\mathbf{y}_i - h(\mathbf{x}_i))^2 - (\mathbf{y}_i - h'(\mathbf{x}_i))^2) + \frac{1}{m} \sum_{q \in \{1, \dots, m\}} ((\mathbf{y}'_q - h(\mathbf{x}'_q))^2 - (\mathbf{y}'_q - h'(\mathbf{x}'_q))^2) \quad (5)$$

Maximizing $\Delta_{\mathbf{x}_u}$ will maximize the first sum term of Eq. 5, but whether it can enable Δ_u be positive should also refer the second sum term. Unfortunately, the value of this sum term is difficult to be measured except that the neighboring relationships between all the labeled examples and $(\mathbf{x}_u, \hat{\mathbf{y}}_u)$ are evaluated. Therefore, there may exist cases where the unlabeled example chosen according to the maximization of $\Delta_{\mathbf{x}_u}$ may decrease Δ_u , which is the cost COREG takes for using $\Delta_{\mathbf{x}_u}$ that can be more efficiently computed to approximate Δ_u . Nevertheless, experiments show that in most cases such an approximation is effective.

It seems that using only one regressor to label the unlabeled examples for itself might be feasible, where the unlabeled examples can be chosen according to the maximization of $\Delta_{\mathbf{x}_u}$.

While considering that the labeled example set usually contains noise, the use of two regressors can be helpful to reduce overfitting.

Let Λ denote the subset of noisy examples in L . For the unlabeled instance $\mathbf{x}_u$, either of the regressors h_1 and h_2 will identify a set of k -nearest neighboring labeled examples for $\mathbf{x}_u$. Assume these sets are Ω_1 and Ω_2 , respectively. Since h_1 and h_2 use different distance orders, Ω_1 and Ω_2 are usually different, and therefore $\Omega_1 \cap \Lambda$ and $\Omega_2 \cap \Lambda$ are also usually different. Suppose $\mathbf{x}_u$ is labeled by h_1 and then $(\mathbf{x}_u, h_1(\mathbf{x}_u))$ is put into L_1 , where $h_1(\mathbf{x}_u)$ suffers from $\Omega_1 \cap \Lambda$. For another unlabeled instance $\mathbf{x}_v$ which is very close to $\mathbf{x}_u$, its k -nearest neighbors identified by h_1 will be very similar to Ω_1 except that $(\mathbf{x}_u, h_1(\mathbf{x}_u))$ has replaced a previous neighbor. Thus, $h_1(\mathbf{x}_v)$ will suffer from $\Omega_1 \cap \Lambda$ more seriously than $h_1(\mathbf{x}_u)$ does. While, if the instance $\mathbf{x}_u$ is labeled by h_2 and $(\mathbf{x}_u, h_2(\mathbf{x}_u))$ is put into L_1 , then $h_1(\mathbf{x}_v)$ will suffer from $\Omega_1 \cap \Lambda$ only once, although $\mathbf{x}_u$ is still very close to $\mathbf{x}_v$.

4 Experiments

Experiments are performed on ten data sets listed in Table 2 where “# attribute” denotes the number of input attributes. These data sets have been used in [Zhou *et al.*, 2002] where the detailed descriptions of the data sets can be found. Note that the input attributes as well as the real-valued labels have been normalized to $[0.0, 1.0]$.

Table 2: Experimental data sets

Data set	# attribute	Size
2-d Mexican Hat	1	5,000
3-d Mexican Hat	2	3,000
Friedman #1	5	5,000
Friedman #2	4	5,000
Friedman #3	4	3,000
Gabor	2	3,000
Multi	5	4,000
Plane	2	1,000
Polynomial	1	3,000
SinC	1	3,000

For each data set, 25% data are kept as the test set, while the remaining 75% data are partitioned into the labeled and unlabeled sets where 10% (of the 75%) data are used as labeled examples while the remaining 90% (of the 75%) data are used as unlabeled examples.

In the experiments, the distance orders used by the two k NN regressors in COREG are set to 2 and 5, respectively, the k value is set to 3, the maximum number of iterations T is set to 100, and the pool U' contains 100 unlabeled examples randomly picked from the unlabeled set in each iteration.

A self-training style algorithm is tested for comparison, which is denoted by SELF. This algorithm uses a k NN regressor and in each iteration, it chooses the unlabeled example which maximizes the value of $\Delta_{\mathbf{x}_u}$ in Eq. 2 to label for itself. Moreover, a co-training style algorithm, denoted by ARTRE, is also tested. Since the experimental data sets are with no sufficient and redundant views, here an artificial redundant view is developed through deriving new attributes

from the original ones. For example, on 3-d Mexican Hat two new attributes, i.e. x_3 and x_4 , are constructed from $x_1 + x_2$ and $x_1 - x_2$, and then a k NN regressor is built on x_1 and x_2 while the other is built on x_3 and x_4 . In each iteration, each k NN regressor chooses the unlabeled example which maximizes the value of $\Delta_{\mathbf{x}_u}$ in Eq. 2 to label for the other regressor. The final prediction is made by averaging the regression estimates of these two refined regressors. Besides, a k NN regressor using only the labeled data is tested as a baseline for the comparison, which is denoted by LABELED.

All the k NN regressors used in SELF, ARTRE, and LABELED employ 2nd-order Minkowski distance, and the k value is set to 3. The same pool, U' , as that used by COREG is used in each iteration of SELF and ARTRE, and the maximum number of iterations is also set to 100.

One hundred runs of experiments are carried out on each data set. In each run, the performance of all the four algorithms, i.e. COREG, SELF, ARTRE, and LABELED, are evaluated on randomly partitioned labeled/unlabeled/test splits. The average MSE at each iteration is recorded. Note that the learning processes of the algorithms may stop before the maximum number of iterations is reached, and in that case, the final MSE is used in computing the average MSE of the following iterations.

The improvement on average MSE obtained by exploiting unlabeled examples is tabulated in Table 3, which is computed by subtracting the final MSE from the initial MSE and then divided by the initial MSE.

Table 3: Improvement on average mean squared error

Data set	SELF	ARTRE	COREG
2d Mexican Hat	9.2%	12.8%	19.6%
3d Mexican Hat	3.9%	3.7%	5.7%
Friedman #1	-1.8%	-4.0%	0.5%
Friedman #2	-1.3%	-4.3%	2.1%
Friedman #3	-0.9%	-3.6%	0.0%
Gabor	4.0%	3.8%	9.0%
Multi	-1.9%	-4.4%	1.4%
Plane	-3.8%	-3.5%	-1.6%
Polynomial	15.1%	17.4%	22.0%
SinC	13.0%	16.4%	26.0%

Table 3 shows that SELF and ARTRE improve the regression estimates on only five data sets, while COREG improves on eight data sets. Moreover, Table 3 discloses that the improvement of COREG is always bigger than that of SELF and ARTRE. These observations tell that COREG can effectively exploit unlabeled examples to improve regression estimates.

For further studying the compared algorithms, the average MSE of different algorithms at different iterations are plotted in Fig 1, where the average MSE of the two k NN regressors used in COREG are also depicted. Note that in each subfigure, every curve contains 101 points corresponding to the 100 learning iterations in addition to the initial condition, where only 11 of them are explicitly depicted for better presentation.

Fig. 1 shows that COREG can exploit unlabeled examples to improve the regression estimates on most data sets, except that on Friedman #3 there is no improvement while on Plane

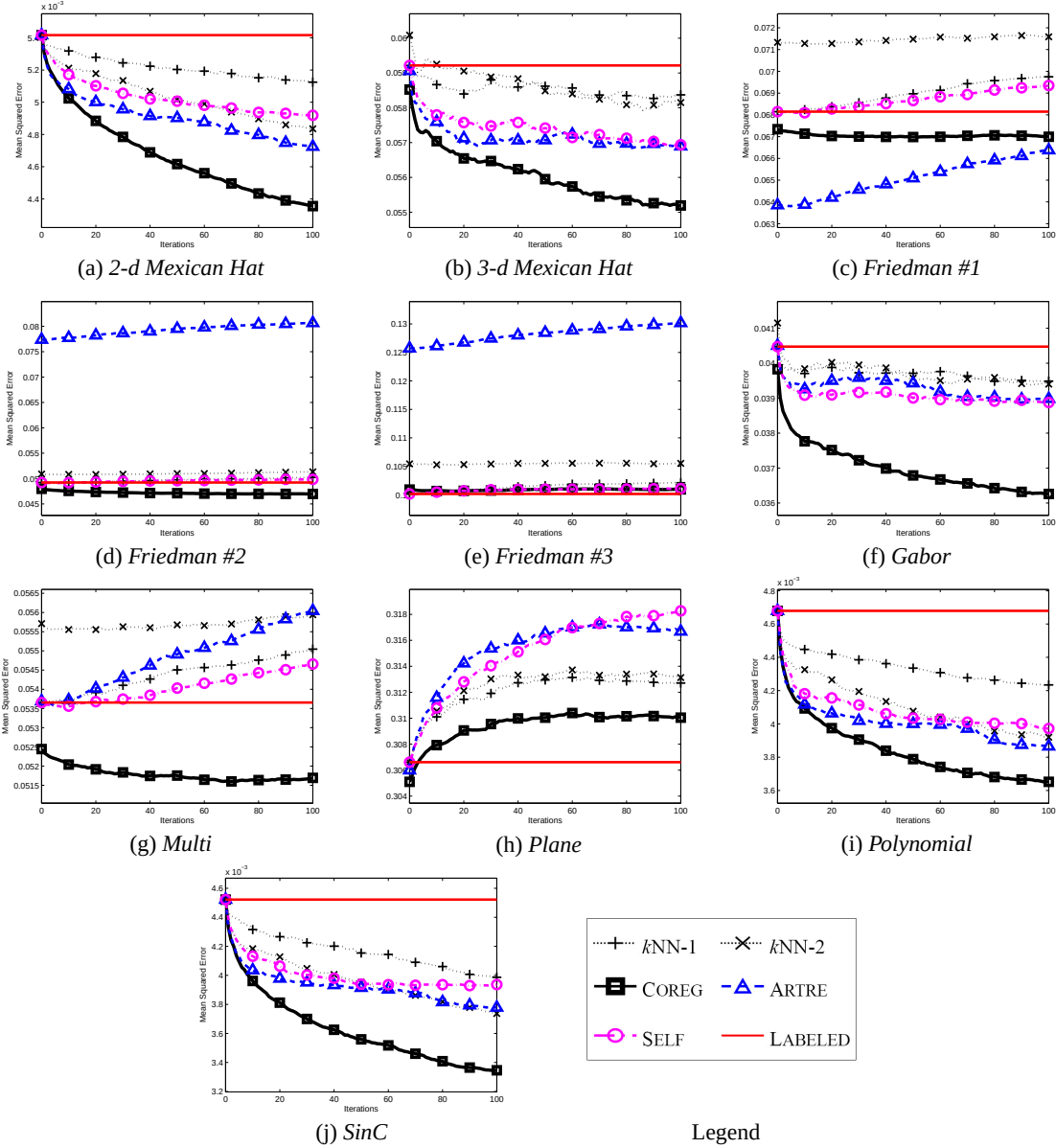


Figure 1: Comparisons on average mean squared error of different algorithms at different iterations

the performance is degenerated. While, SELF and ARTRE degenerate the regression estimates on five data sets, i.e. *Friedman #1* to #3, *Multi*, and *Plane*. Moreover, the average MSE of the final prediction made by COREG is almost always the best except on *Friedman #1* where ARTRE is slightly better and on *Plane* where LABELED is the best while all the semi-supervised learning algorithms degenerate the performance. These observations disclose that COREG is apparently the best algorithm in the comparison.

Pairwise two-tailed t -tests with 0.05 significance level reveal that the final regression estimates of COREG are significantly better than its initial regression estimates on almost all the data sets except that on *Plane* the latter is better while on *Friedman #3* there is no significant difference. Moreover,

the final regression estimates of COREG are significantly better than these of ARTRE on almost all the data sets except on *Friedman #1* where the latter is better. Furthermore, the final regression estimates of COREG are significantly better than these of SELF and LABELED on almost all the data sets except on *Plane* where LABELED is better and on *Friedman #3* where there is no significant difference. These results of t -tests confirm that COREG is the strongest among the compared algorithms, which can effectively exploit unlabeled data to improve the regression estimates.

5 Conclusion

This paper proposes a co-training style semi-supervised regression algorithm COREG. This algorithm employs two k -

nearest neighbor regressors using different distance metrics. In each learning iteration, each regressor labels the unlabeled example which can be most confidently labeled for the other learner, where the labeling confidence is estimated through considering the consistency of the regressor with the labeled example set. The final prediction is made by averaging the predictions of both the refined k NN regressors. Experiments show that COREG can effectively exploit unlabeled data to improve the regression estimates.

In contrast to standard co-training setting, COREG does not require sufficient and redundant views, which enables it have broad applicability. However, this forces COREG generate diverse initial regressors with specific mechanisms. In this paper the diversity is obtained by instantiating the Minkowski distance with different distance orders. It is obvious that using completely different distance metrics may be more helpful. Moreover, trying to obtain the diversity of the initial regressors from channels other than using different distance metrics is an issue to be investigated in future work. Note that although this paper uses k NN regressor as the base learner, an important idea of COREG, i.e. regarding the labeling of the unlabeled example which makes the regressor most consistent with the labeled example set as with the most confidence, can also be used with other base learners. Therefore, designing semi-supervised regression algorithms with other base learners along the way of COREG is another interesting issue to be explored in the future. Furthermore, designing semi-supervised regression algorithms outside the co-training framework is also well-worth studying.

Acknowledgments

Supported by NSFC (60325207), FANEDD (200343), and JIANGSUSF (BK2004001).

References

- [Blum and Chawla, 2001] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pages 19–26, Williamston, MA, 2001. Morgan Kaufmann.
- [Blum and Mitchell, 1998] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, pages 92–100, Madison, WI, 1998. ACM Press.
- [Dasarathy, 1991] B. V. Dasarathy. *Nearest Neighbor Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
- [Dasgupta et al., 2002] S. Dasgupta, M. Littman, and D. McAllester. PAC generalization bounds for co-training. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 375–382. MIT Press, Cambridge, MA, 2002.
- [Dempster et al., 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.
- [Goldman and Zhou, 2000] S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the 17th International Conference on Machine Learning*, pages 327–334, San Francisco, CA, 2000. Morgan Kaufmann.
- [Hwa et al., 2003] R. Hwa, M. Osborne, A. Sarkar, and M. Steedman. Corrected co-training for statistical parsers. In *Working Notes of the ICML '03 Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, Washington, DC, 2003.
- [Joachims, 1999] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, pages 200–209, Bled, Slovenia, 1999. Morgan Kaufmann.
- [Miller and Uyar, 1997] D. J. Miller and H. S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In M. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 571–577. MIT Press, Cambridge, MA, 1997.
- [Nigam and Ghani, 2000] K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management*, pages 86–93, Washington, DC, 2000. ACM Press.
- [Nigam et al., 2000] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2–3):103–134, 2000.
- [Pierce and Cardie, 2001] D. Pierce and C. Cardie. Limitations of co-training for natural language learning from large data sets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 1–9, Pittsburgh, PA, 2001.
- [Sarkar, 2001] A. Sarkar. Applying co-training methods to statistical parsing. In *Proceedings of the 2nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 95–102, Pittsburgh, PA, 2001.
- [Shahshahani and Landgrebe, 1994] B. Shahshahani and D. Landgrebe. The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5):1087–1095, 1994.
- [Steedman et al., 2003] M. Steedman, M. Osborne, A. Sarkar, S. Clark, R. Hwa, J. Hockenmaier, P. Ruhlén, S. Baker, and J. Crim. Bootstrapping statistical parsers from small data sets. In *Proceedings of the 11th Conference on the European Chapter of the Association for Computational Linguistics*, pages 331–338, Budapest, Hungary, 2003.
- [Zhou et al., 2002] Z.-H. Zhou, J. Wu, and W. Tang. Ensembling neural networks: many could be better than all. *Artificial Intelligence*, 137(1–2):239–263, 2002.

MULTI-AGENT SYSTEMS

On the Axiomatic Foundations of Ranking Systems

Alon Altman and Moshe Tennenholtz

Faculty of Industrial Engineering and Management

Technion – Israel Institute of Technology

Haifa 32000

Israel

Abstract

Reasoning about agent preferences on a set of alternatives, and the aggregation of such preferences into some social ranking is a fundamental issue in reasoning about multi-agent systems. When the set of agents and the set of alternatives coincide, we get the ranking systems setting. A famous type of ranking systems are page ranking systems in the context of search engines. In this paper we present an extensive axiomatic study of ranking systems. In particular, we consider two fundamental axioms: Transitivity, and Ranked Independence of Irrelevant Alternatives. Surprisingly, we find that there is no general social ranking rule that satisfies both requirements. Furthermore, we show that our impossibility result holds under various restrictions on the class of ranking problems considered. Each of these axioms can be individually satisfied. Moreover, we show a complete axiomatization of approval voting using one of these axioms.

1 Introduction

The ranking of agents based on other agents' input is fundamental to multi-agent systems (see e.g. [Resnick *et al.*, 2000]). Moreover, it has become a central ingredient of a variety of Internet sites, where perhaps the most famous examples are Google's PageRank algorithm[Page *et al.*, 1998] and eBay's reputation system[Resnick and Zeckhauser, 2001].

This basic problem introduces a new social choice model. In the classical theory of social choice, as manifested by Arrow[1963], a set of agents/voters is called to rank a set of alternatives. Given the agents' input, i.e. the agents' individual rankings, a social ranking of the alternatives is generated. The theory studies desired properties of the aggregation of agents' rankings into a social ranking. In particular, Arrow's celebrated impossibility theorem[Arrow, 1963] shows that there is no aggregation rule that satisfies some minimal requirements, while by relaxing any of these requirements appropriate social aggregation rules can be defined. The novel feature of the ranking systems setting is that the set of agents and the set of alternatives **coincide**. Therefore, in such setting one may need to consider the transitive effects of voting. For example, if agent *a* reports on the importance of (i.e. votes

for) agent *b* then this may influence the credibility of a report by *b* on the importance of agent *c*; these indirect effects should be considered when we wish to aggregate the information provided by the agents into a social ranking.

Notice that a natural interpretation/application of this setting is the ranking of Internet pages. In this case, the set of agents represents the set of Internet pages, and the links from a page *p* to a set of pages *Q* can be viewed as a two-level ranking where agents in *Q* are preferred by agent(page) *p* to the agents(pages) which are not in *Q*. The problem of finding an appropriate social ranking in this case is in fact the problem of (global) page ranking. Particular approaches for obtaining a useful page ranking have been implemented by search engines such as Google[Page *et al.*, 1998].

The theory of social choice consists of two complementary axiomatic perspectives:

- The descriptive perspective: given a particular rule *r* for the aggregation of individual rankings into a social ranking, find a set of axioms that are sound and complete for *r*. That is, find a set of requirements that *r* satisfies; moreover, every social aggregation rule that satisfies these requirements should coincide with *r*. A result showing such an axiomatization is termed a *representation theorem* and it captures the exact essence of (and assumptions behind) the use of the particular rule.
- The normative perspective: devise a set of requirements that a social aggregation rule should satisfy, and try to find whether there is a social aggregation rule that satisfies these requirements.

Many efforts have been invested in the descriptive approach in the framework of the classical theory of social choice. In that setting, representation theorems have been presented to major voting rules such as the majority rule[May, 1952]. Recently, we have successfully applied the descriptive perspective in the context of ranking systems by providing a representation theorem[Altman and Tennenholtz, 2005] for the well-known PageRank algorithm [Page *et al.*, 1998], which is the basis of Google's search technology.

An excellent example for the normative perspective is Arrow's impossibility theorem mentioned above. In [Tennenholtz, 2004], we presented some preliminary results for ranking systems where the set of voters and the set of alternatives coincide. However, the axioms presented in that work consist

of several very strong requirements which naturally lead to an impossibility result.

In this paper we provide an extensive study of ranking systems. We introduce two fundamental axioms. One of these axioms captures the transitive effects of voting in ranking systems, and the other adapts Arrow’s well-known independence of irrelevant alternatives (IIA) axiom to the context of ranking systems. Surprisingly, we find that no general ranking system can simultaneously satisfy these two axioms! We further show that our impossibility result holds under various restrictions on the class of ranking problems considered. On the other hand, we show that each of these axioms can be individually satisfied. Moreover, we use our IIA axiom to present a positive result in the form of a representation theorem for the well-known approval voting ranking system, which ranks the agents based on the number of votes received. This axiomatization shows that when ignoring transitive effects, there is only one ranking system that satisfies our IIA axiom.

This paper is structured as follows: Section 2 formally defines our setting and the notion of ranking systems. Sections 3 and 4 introduce our axioms of Transitivity and Ranked Independence of Irrelevant Alternatives respectively. Our main impossibility result is presented in Section 5, and further strengthened in Section 6. Our positive result, in the form of an axiomatization for the Approval Voting ranking system is presented in Section 7. Finally, some concluding remarks are given in Section 8.

2 Ranking Systems

Before describing our results regarding ranking systems, we must first formally define what we mean by the words “ranking system” in terms of graphs and linear orderings:

Definition 2.1. Let A be some set. A relation $R \subseteq A \times A$ is called an *ordering* on A if it is reflexive, transitive, and complete. Let $L(A)$ denote the set of orderings on A .

Notation 2.2. Let $\preceq$ be an ordering, then $\simeq$ is the equality predicate of $\preceq$, and $\prec$ is the strict order induced by $\preceq$. Formally, $a \simeq b$ if and only if $a \preceq b$ and $b \preceq a$; and $a \prec b$ if and only if $a \preceq b$ but not $b \preceq a$.

Given the above we can define what a ranking system is:

Definition 2.3. Let $\mathbb{G}_V$ be the set of all graphs with vertex set V . A *ranking system* F is a functional that for every finite vertex set V maps graphs $G \in \mathbb{G}_V$ to an ordering $\preceq_G^F \in L(V)$. If F is a partial function then it is called a *partial ranking system*, otherwise it is called a *general ranking system*.

One can view this setting as a variation/extension of the classical theory of social choice as modeled by [Arrow, 1963]. The ranking systems setting differs in two main properties. First, in this setting we assume that the set of voters and the set of alternatives coincide, and second, we allow agents only two levels of preference over the alternatives, as opposed to Arrow’s setting where agents could rank alternatives arbitrarily.

3 Transitivity

A basic property one would assume of ranking systems is that if an agent a ’s voters are ranked higher than those of agent b ,

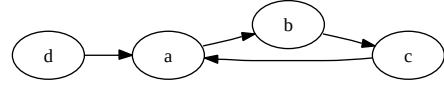


Figure 1: Example of Transitivity

then agent a should be ranked higher than agent b . This notion is formally captured below:

Definition 3.1. Let F be a ranking system. We say that F satisfies *strong transitivity* if for all graphs $G = (V, E)$ and for all vertices $v_1, v_2 \in V$: Assume there is a 1-1 mapping $f : P(v_1) \mapsto P(v_2)$ s.t. for all $v \in P(v_1)$: $v \preceq f(v)$. Further assume that either f is not onto or for some $v \in P(v_1)$: $v \prec f(v)$. Then, $v_1 \prec v_2$.

Consider for example the graph G in Figure 1 and any ranking system F that satisfies strong transitivity. F must rank vertex d below all other vertices, as it has no predecessors, unlike all other vertices. If we assume that $a \preceq_G^F b$, then by strong transitivity we must conclude that $b \preceq_G^F c$ as well. But then we must conclude that $b \prec_G^F a$ (as b ’s predecessor a is ranked lower than a ’s predecessor c , and a has an additional predecessor d), which leads to a contradiction. Given $b \prec_G^F a$, again by transitivity, we must conclude that $c \prec_G^F b$, so the only ranking for the graph G that satisfies strong transitivity is $d \prec_G^F c \prec_G^F b \prec_G^F a$.

In [Tennenholtz, 2004], we have suggested an algorithm that defines a ranking system that satisfies strong transitivity by iteratively refining an ordering of the vertices.

Note that the PageRank ranking system defined in [Page et al., 1998] does not satisfy strong transitivity. This is due to the fact that PageRank reduces the weight of links (or votes) from nodes which have a higher out-degree. Thus, assuming Yahoo! and Microsoft are equally ranked, a link from Yahoo! means less than a link from Microsoft, because Yahoo! links to more external pages than does Microsoft. Noting this fact, we can weaken the definition of transitivity to require that the predecessors of the compared agents have an equal out-degree:

Definition 3.2. Let F be a ranking system. We say that F satisfies *weak transitivity* if for all graphs $G = (V, E)$ and for all vertices $v_1, v_2 \in V$: Assume there is a 1-1 mapping $f : P(v_1) \mapsto P(v_2)$ s.t. for all $v \in P(v_1)$: $v \preceq f(v)$ and $|S(v)| = |S(f(v))|$. Further assume that either f is not onto or for some $v \in P(v_1)$: $v \prec f(v)$. Then, $v_1 \prec v_2$.

Indeed, an idealized version of the PageRank ranking system defined on strongly connected graphs satisfies this weakened version of transitivity. Furthermore, the result in the example above does not change when we consider weak transitivity in place of strong transitivity.

4 Ranked Independence of Irrelevant Alternatives

A standard assumption in social choice settings is that an agent’s relative rank should only depend on (some property of) their immediate predecessors. Such axioms are usually called independence of irrelevant alternatives (IIA) axioms.

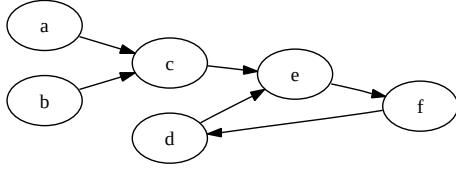


Figure 2: An example of RIIA.

In our setting, we require the relative ranking of two agents must only depend on the pairwise comparisons of the ranks of their predecessors, and not on their identity or cardinal value. Our IIA axiom, called *ranked* IIA, differs from the one suggested by [Arrow, 1963] in the fact that we do not consider the identity of the voters, but rather their relative rank.

For example, consider the graph in Figure 2. Furthermore, assume a ranking system F has ranked the vertices of this graph as following: $a \simeq b \prec c \simeq d \prec e \simeq f$. Now look at the comparison between c and d . c 's predecessors, a and b , are both ranked equally, and both ranked lower than d 's predecessor f . This is also true when considering e and f – e 's predecessors c and d are both ranked equally, and both ranked lower than f 's predecessor e . Therefore, if we agree with ranked IIA, the relation between c and d , and the relation between e and f must be the same, which indeed it is – both $c \simeq d$ and $e \simeq f$. However, this same situation also occurs when comparing c and f (c 's predecessors a and b are equally ranked and ranked lower than f 's predecessor e), but in this case $c \prec f$. So, we can conclude that the ranking system F which produced these rankings does not satisfy ranked IIA.

To formally define this condition, one must consider all possibilities of comparing two nodes in a graph based only on ordinal comparisons of their predecessors. We call these possibilities comparison profiles:

Definition 4.1. A *comparison profile* is a pair $\langle \mathbf{a}, \mathbf{b} \rangle$ where $\mathbf{a} = (a_1, \dots, a_n)$, $\mathbf{b} = (b_1, \dots, b_m)$, $a_1, \dots, a_n, b_1, \dots, b_m \in \mathbb{N}$, $a_1 \leq a_2 \leq \dots \leq a_n$, and $b_1 \leq b_2 \leq \dots \leq b_m$. Let $\mathcal{P}$ be the set of all such profiles.

A ranking system F , a graph $G = (V, E)$, and a pair of vertices $v_1, v_2 \in V$ are said to *satisfy* such a comparison profile $\langle \mathbf{a}, \mathbf{b} \rangle$ if there exist 1-1 mappings $f_1 : P(v_1) \mapsto \{1 \dots n\}$ and $f_2 : P(v_2) \mapsto \{1 \dots m\}$ such that given $f : (\{1\} \times P(v_1)) \cup (\{2\} \times P(v_2)) \mapsto \mathbb{N}$ defined as:

$$\begin{aligned} f(1, v) &= a_{f_1(v)} \\ f(2, u) &= b_{f_2(u)}, \end{aligned}$$

$$f(i, x) \leq f(j, y) \Leftrightarrow x \preceq_G^F y \text{ for all } (i, x), (j, y) \in (\{1\} \times P(v_1)) \cup (\{2\} \times P(v_2)).$$

For example, in the example considered above, all of the pairs (c, d) , (c, f) , and (e, f) satisfy the comparison profile $\langle (1, 1), (2) \rangle$.

We now require that for every such profile the ranking system ranks the nodes consistently:

Definition 4.2. Let F be a ranking system. We say that F satisfies *ranked independence of irrelevant alternatives (RIIA)* if there exists a mapping $f : \mathcal{P} \mapsto \{0, 1\}$ such that for every graph $G = (V, E)$ and for every pair of vertices $v_1, v_2 \in V$

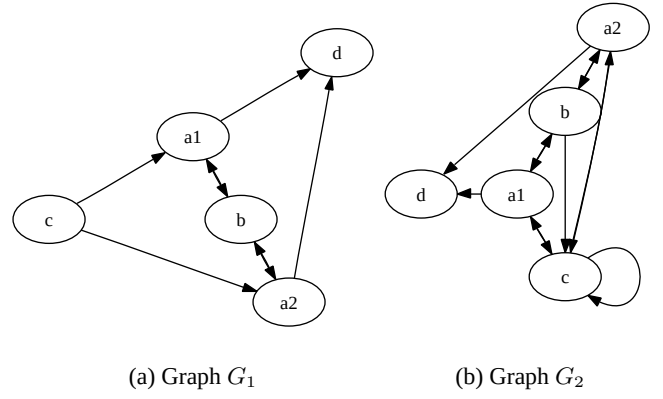


Figure 3: Graphs for the proof of Theorem 5.1

and for every comparison profile $p \in \mathcal{P}$ that v_1 and v_2 satisfy, $v_1 \preceq_G^F v_2 \Leftrightarrow f(p) = 1$.

As RIIA is an independence property, the ranking system $F_{\equiv}$, that ranks all agents equally, satisfies RIIA. A more interesting ranking system that satisfies RIIA is the approval voting ranking system, defined below.

Definition 4.3. The *approval voting ranking system* AV is the ranking system defined by:

$$v_1 \preceq_G^{AV} v_2 \Leftrightarrow |P(v_1)| \leq |P(v_2)|$$

A full axiomatization of the approval voting ranking system is given in section 7.

5 Impossibility

Our main result illustrates the impossibility of satisfying (weak) transitivity and RIIA simultaneously.

Theorem 5.1. *There is no general ranking system that satisfies weak transitivity and RIIA.*

Proof. Assume for contradiction that there exists a ranking system F that satisfies weak transitivity and RIIA. Consider first the graph G_1 in Figure 3(a). First, note that a_1 and a_2 satisfy some comparison profile $p_a = ((x, y), (x, y))$ because they have identical predecessors. Thus, by RIIA, $a_1 \preceq_{G_1}^F a_2 \Leftrightarrow a_2 \preceq_{G_1}^F a_1$, and therefore $a_1 \simeq_{G_1}^F a_2$. By weak transitivity, it is easy to see that $c \prec_{G_1}^F a_1$ and $c \prec_{G_1}^F b$. If we assume $b \preceq_{G_1}^F a_1$, then by weak transitivity, $a_1 \prec_{G_1}^F b$ which contradicts our assumption. So we conclude that $c \prec_{G_1}^F a_1 \prec_{G_1}^F b$.

Now consider the graph G_2 in Figure 3(b). Again, by RIIA, $a_1 \simeq_{G_2}^F a_2$. By weak transitivity, it is easy to see that $a_1 \prec_{G_2}^F c$ and $b \prec_{G_2}^F c$. If we assume $a_1 \preceq_{G_2}^F b$, then by weak transitivity, $b \prec_{G_2}^F a_1$ which contradicts our assumption. So we conclude that $b \prec_{G_2}^F a_1 \prec_{G_2}^F c$.

Consider the comparison profile $p = ((1, 3), (2, 2))$. Given F , a_1 and b satisfy p in G_1 (because $c \prec_{G_1}^F a_1 \simeq_{G_1}^F a_2 \prec_{G_1}^F b$) and in G_2 (because $b \prec_{G_2}^F a_1 \simeq_{G_2}^F a_2 \prec_{G_2}^F c$). Thus, by

RIIA, $a_1 \preceq_{G_1}^F b \Leftrightarrow a_1 \preceq_{G_2}^F b$, which is a contradiction to the fact that $a_1 \prec_{G_1}^F b$ but $b \prec_{G_2}^F a_1$. $\square$

This result is quite a surprise, as it means that every reasonable definition of a ranking system must either consider cardinal values for nodes and/or edges (like [Page *et al.*, 1998]), or operate ordinally on a global scale (like [Altman and Tennenholtz, 2005]).

6 Relaxing Generality

A hidden assumption in our impossibility result is the fact that we considered only general ranking systems. In this section we analyze several special classes of graphs that relate to common ranking scenarios.

6.1 Small Graphs

A natural limitation on a preference graph is a cap on the number of vertices (agents) that participate in the ranking. Indeed, when there are three or less agents involved in the ranking, strong transitivity and RIIA can be simultaneously satisfied. An appropriate ranking algorithm for this case is the one we suggested in [Tennenholtz, 2004].

However, when there are four or more agents, strong transitivity and RIIA cannot be simultaneously satisfied (the proof is similar to that of Theorem 5.1, but with vertex d removed in both graphs). When five or more agents are involved, even weak transitivity and RIIA cannot be simultaneously satisfied, as implied by the proof of Theorem 5.1.

6.2 Single Vote Setting

Another natural limitation on the domain of graphs that we might be interested in is the restriction of each agent(vertex) to exactly one vote(successor). For example, in the voting paradigm this could be viewed as a setting where every agent votes for exactly one agent. The following proposition shows that even in this simple setting weak transitivity and RIIA cannot be simultaneously satisfied.

Proposition 6.1. *Let $\mathbb{G}_1$ be the set of all graphs $G = (V, E)$ such that $|S(v)| = 1$ for all $v \in V$. There is no partial ranking system over $\mathbb{G}_1$ that satisfies weak transitivity and RIIA.*

Proof. Assume for contradiction that there is a partial ranking system F over $\mathbb{G}_1$ that satisfies weak transitivity and RIIA. Let $f : \mathcal{P} \mapsto \{0, 1\}$ be the mapping from the definition of RIIA for F .

Let $G_1 \in \mathbb{G}_1$ be the graph in Figure 4a. By weak transitivity, $x_1 \simeq_{G_1}^F x_2 \prec_{G_1}^F b \prec_{G_1}^F a$. (a, b) satisfies the comparison profile $\langle (1, 1, 2), (3) \rangle$, so we must have $f(\langle (1, 1, 2), (3) \rangle) = 0$. Now let $G_2 \in \mathbb{G}_1$ be the graph in Figure 4b. By weak transitivity $x_1 \simeq_{G_2}^F x_2 \prec_{G_2}^F y \prec_{G_2}^F a \prec_{G_2}^F b$. (b, a) satisfies the comparison profile $\langle (2, 3), (1, 4) \rangle$, so we must have $f(\langle (2, 3), (1, 4) \rangle) = 0$.

Let $G_3 \in \mathbb{G}_1$ be the graph in Figure 4c. By weak transitivity it is easy to see that $x_1 \simeq_{G_3}^F \dots \simeq_{G_3}^F x_7 \prec_{G_3}^F y_1 \simeq_{G_3}^F y_2 \prec_{G_3}^F c \prec_{G_3}^F d$. Furthermore, by weak transitivity we conclude that $a \prec_{G_3}^F b$ and $a' \prec_{G_3}^F b'$ from $c \prec_{G_3}^F d$; and

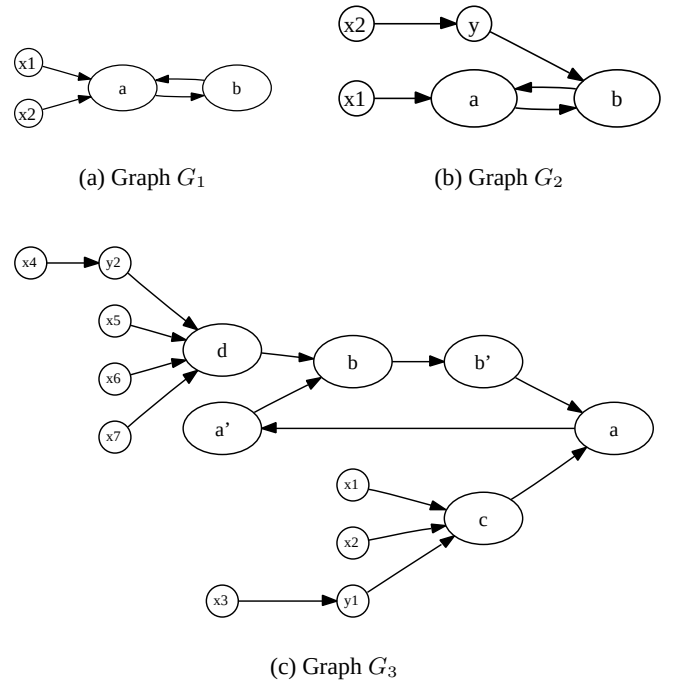


Figure 4: Graphs from the proof of proposition 6.1

$y_1 \prec_{G_3}^F b$ from $x_3 \prec_{G_3}^F d$. Now consider the vertex pair (c, b') . We have shown that $x_1 \simeq_{G_3}^F x_2 \prec_{G_3}^F y_1 \prec_{G_3}^F b$. So, (c, b') satisfies the comparison profile $\langle (1, 1, 2), (3) \rangle$, thus by RIIA $b' \prec_{G_3}^F c$. Now consider the vertex pair (b, a) . We have already shown that $a' \prec_{G_3}^F b' \prec_{G_3}^F c \prec_{G_3}^F d$. So, (a, b) satisfies the comparison profile $\langle (2, 3), (1, 4) \rangle$, thus by RIIA $b \prec_{G_3}^F a$. However, we have already shown that $a \prec_{G_3}^F b$ – a contradiction. Thus, the ranking system F cannot exist. $\square$

6.3 Bipartite Setting

In the world of reputation systems [Resnick *et al.*, 2000], we frequently observe a distinction between two types of agents such that each type of agent only ranks agents of the other type. For example buyers only interact with sellers and vice versa. This type of limitation is captured by requiring the preference graphs to be bipartite, as defined below.

Definition 6.2. A graph $G = (V, E)$ is called *bipartite* if there exist V_1, V_2 such that $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, and $E \subseteq (V_1 \times V_2) \cup (V_2 \times V_1)$. Let $\mathbb{G}_B$ be the set of all bipartite graphs.

Our impossibility result extends to the limited domain of bipartite graphs.

Proposition 6.3. *There is no partial ranking system over $\mathbb{G}_B \cap \mathbb{G}_1$ that satisfies weak transitivity and RIIA.*

6.4 Strongly Connected Graphs

The well-known PageRank ranking system is (ideally) defined on the set of strongly connected graphs. That is, the

set of graphs where there exists a directed path between any two vertices.

Let us denote the set of all strongly connected graphs $\mathbb{G}_{SC}$. The following proposition extends our impossibility result to strongly connected graphs.

Proposition 6.4. *There is no partial ranking system over $\mathbb{G}_{SC}$.*

7 Axiomatization of Approval Voting

In the previous sections we have seen mostly negative results which arise when trying to accommodate (weak) transitivity and RIIA. We have shown that although each of the axioms can be satisfied separately, there exists no general ranking system that satisfies both axioms.

We have previously shown [Tennenholtz, 2004] a non-trivial ranking system that satisfies (weak) transitivity, but have not yet shown such a ranking system for RIIA. In this section we provide a representation theorem for a ranking system that satisfies RIIA but not weak transitivity — the approval voting ranking system. This system ranks the agents based on the number of votes each agent received, with no regard to the rank of the voters. The axiomatization we provide in this section shows the power of RIIA, as it shows that there exists only one (interesting) ranking system that satisfies it without introducing transitive effects.

In order to specify our axiomatization, recall the following classical definitions from the theory of social choice:

The positive response axiom essentially means that if an agent receives additional votes, its rank must improve:

Definition 7.1. Let F be a ranking system. F satisfies *positive response* if for all graphs $G = (V, E)$ and for all $(v_1, v_2) \in (V \times V) \setminus E$, and for all $v_3 \in V$: Let $G' = (V, E \cup (v_1, v_2))$. If $v_3 \preceq_G^F v_2$, then $v_3 \prec_{G'}^F v_2$.

The anonymity and neutrality axioms mean that the names of the voters and alternatives respectively do not matter for the ranking:

Definition 7.2. A ranking system F satisfies *anonymity* if for all $G = (V, E)$, for all permutations $\pi : V \mapsto V$, and for all $v_1, v_2 \in V$: Let $E' = \{(\pi(v_1), v_2) | (v_1, v_2) \in E\}$. Then, $v_1 \preceq_{(V, E)}^F v_2 \Leftrightarrow v_1 \preceq_{(V, E')}^F v_2$.

Definition 7.3. A ranking system F satisfies *neutrality* if for all $G = (V, E)$, for all permutations $\pi : V \mapsto V$, and for all $v_1, v_2 \in V$: Let $E' = \{(v_1, \pi(v_2)) | (v_1, v_2) \in E\}$. Then, $v_1 \preceq_{(V, E)}^F v_2 \Leftrightarrow v_1 \preceq_{(V, E')}^F v_2$.

Arrow's classical Independence of Irrelevant Alternatives axiom requires that the relative rank of two agents be dependent only on the set of agents that preferred one over the other.

Definition 7.4. A ranking system F satisfies *Arrow's Independence of Irrelevant Alternatives (AIIA)* if for all $G = (V, E)$, for all $G' = (V, E')$, and for all $v_1, v_2 \in V$: Let $P_G(v_1) \setminus P_G(v_2) = P_{G'}(v_1) \setminus P_{G'}(v_2)$ and $P_G(v_2) \setminus P_G(v_1) = P_{G'}(v_2) \setminus P_{G'}(v_1)$. Then, $v_1 \preceq_G^F v_2 \Leftrightarrow v_1 \preceq_{G'}^F v_2$.

Our representation theorem states that together with positive response and RIIA, any one of the three independence

conditions above (anonymity, neutrality, and AIIA) are essential and sufficient for a ranking system being AV¹. In addition, we show that as in the classical social choice setting when only considering two-level preferences, positive response, anonymity, neutrality, and AIIA are an essential and sufficient representation of approval voting. This result extends the well known axiomatization of the majority rule due to [May, 1952]:

Proposition 7.5. (May's Theorem) *A social welfare functional over two alternatives is a majority social welfare functional if and only if it satisfies anonymity, neutrality, and positive response.*

We can now formally state our theorem:

Theorem 7.6. *Let F be a general ranking system. Then, the following statements are equivalent:*

1. F is the approval voting ranking system ($F = AV$)
2. F satisfies positive response, anonymity, neutrality, and AIIA
3. F satisfies positive response, RIIA, and either one of anonymity, neutrality, and AIIA

Proof. (Sketch) It is easy to see that AV satisfies positive response, RIIA, anonymity, neutrality, and AIIA. It remains to show that (2) and (3) entail (1) above.

To prove (2) entails (1), assume that F satisfies positive response, anonymity, neutrality, and AIIA. Let $G = (V, E)$ be some graph and let $v_1, v_2 \in V$ be some agents. By AIIA, the relative ranking of v_1 and v_2 depends only on the sets $P_G(v_1) \setminus P_G(v_2)$ and $P_G(v_2) \setminus P_G(v_1)$. We have now narrowed our consideration to a set of agents with preferences over two alternatives, so we can apply Proposition 7.5 to complete our proof.

To prove (3) entails (1), assume that F satisfies positive response, RIIA and either anonymity or neutrality or AIIA. As F satisfies RIIA we can limit our discussion to comparison profiles. Let $f : \mathcal{P} \mapsto \{0, 1\}$ be the function from the definition of RIIA. We will use the notation $\mathbf{a} \preceq \mathbf{b}$ to mean $f(\mathbf{a}, \mathbf{b}) = 1$, $\mathbf{a} \prec \mathbf{b}$ to mean $f(\mathbf{b}, \mathbf{a}) = 0$, and $\mathbf{a} \simeq \mathbf{b}$ to mean $\mathbf{a} \preceq \mathbf{b}$ and $\mathbf{b} \preceq \mathbf{a}$.

By the definition of RIIA, it is easy to see that $\mathbf{a} \simeq \mathbf{a}$ for all $\mathbf{a}$. By positive response it is also easy to see that $(\underbrace{1, 1, \dots, 1}_n) \preceq (\underbrace{1, 1, \dots, 1}_m)$ iff $n \leq m$. Let $P = \langle (a_1, \dots, a_n), (b_1, \dots, b_m) \rangle$ be a comparison profile. Let $G = (V, E)$ be the following graph:

$$\begin{aligned} V &= \{x_1, \dots, x_{\max\{a_n, b_m\}}\} \cup \\ &\quad \cup \{v_1, \dots, v_n, v'_1, \dots, v'_n, v\} \cup \\ &\quad \cup \{u_1, \dots, u_m, u'_1, \dots, u'_m, u\} \\ E &= \{(x_i, v_j) | i \leq a_j\} \cup \{(x_i, u_j) | i \leq b_j\} \cup \\ &\quad \cup \{(v_i, v) | i = 1, \dots, n\} \cup \{(u_i, u) | i = 1, \dots, m\}. \end{aligned}$$

¹In fact, an even weaker condition of *decoupling*, that in essence allows us to permute the graph structure while keeping the edges' names is sufficient in this case.

It is easy to see that in the graph G , v and u satisfy the profile P . Let π be the following permutation:

$$\pi(x) = \begin{cases} v'_i & x = v_i \\ v_i & x = v'_i \\ u'_i & x = u_i \\ u_i & x = u'_i \\ x & \text{Otherwise.} \end{cases}$$

The remainder of the proof depends on which additional axiom F satisfies:

- If F satisfies anonymity, let $E' = \{(\pi(x), y) | (x, y) \in E\}$. Note that in the graph (V, E') v and u satisfy the profile $\langle \underbrace{(1, 1, \dots, 1)}_n, \underbrace{(1, 1, \dots, 1)}_m \rangle$, and thus $v \preceq_{(V, E')}^F u$.
 $u \Leftrightarrow n \leq m$. By anonymity, $u \preceq_{(V, E)}^F v \Leftrightarrow u \preceq_{(V, E')}^F v$, thus proving that $f(P) = 1 \Leftrightarrow n \leq m$ for an arbitrary comparison profile P , and thus $F = AV$.
- If F satisfies neutrality, let $E' = \{(x, \pi(y)) | (x, y) \in E\}$. Note that in the graph (V, E') v and u satisfy the profile $\langle \underbrace{(1, 1, \dots, 1)}_n, \underbrace{(1, 1, \dots, 1)}_m \rangle$, and thus $v \preceq_{(V, E')}^F u$.
 $u \Leftrightarrow n \leq m$. By neutrality, $u \preceq_{(V, E)}^F v \Leftrightarrow u \preceq_{(V, E')}^F v$, again showing that $f(P) = 1 \Leftrightarrow n \leq m$ for an arbitrary comparison profile P , and thus $F = AV$.
- If F satisfies AIIA, let $E' = \{(x, \pi(y)) | (x, y) \in E\}$ as before. So, also $v \preceq_{(V, E')}^F u \Leftrightarrow n \leq m$. Note that $P_G(v) = P_{(V, E')}(v)$ and $P_G(u) = P_{(V, E')}(u)$, so by AIIA, $u \preceq_{(V, E)}^F v \Leftrightarrow u \preceq_{(V, E')}^F v$, and thus as before, $F = AV$. $\square$

8 Concluding Remarks

Reasoning about preferences and preference aggregation is a fundamental task in reasoning about multi-agent systems (see e.g. [Boutilier et al., 2004; Conitzer and Sandholm, 2002; LaMura and Shoham, 1998]). A typical instance of preference aggregation is the setting of ranking systems. Ranking systems are fundamental ingredients of some of the most famous tools/techniques in the Internet (e.g. Google's page rank and eBay's reputation systems, among many others).

Our aim in this paper was to treat ranking systems from an axiomatic perspective. The classical theory of social choice lay the foundations to a large part of the rigorous work on multi-agent systems. Indeed, the most classical results in the theory of mechanism design, such as the Gibbard-Satterthwaite Theorem (see [Mas-Colell et al., 1995]) are applications of the theory of social choice. Moreover, previous work in AI has employed the theory of social choice for obtaining foundations for reasoning tasks [Doyle and Wellman, 1989] and multi-agent coordination [Kfir-Dahav and Tennenholtz, 1996]. It is however interesting to note that ranking systems suggest a novel and new type of theory of social choice. We see this point as especially attractive, and as a main reason for concentrating on the study of the axiomatic foundations of ranking systems.

In this paper we identified two fundamental axioms for ranking systems, and conducted a basic axiomatic study of

such systems. In particular, we presented surprising impossibility results, and a representation theorem for the well-known approval voting scheme.

References

- [Altman and Tennenholtz, 2005] Alon Altman and Moshe Tennenholtz. Ranking systems: The pagerank axioms. In *Proceedings of the 6th ACM conference on Electronic commerce (EC-05)*, 2005.
- [Arrow, 1963] K.J. Arrow. *Social Choice and Individual Values (2nd Ed.)*. Yale University Press, 1963.
- [Boutilier et al., 2004] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, 21:135–191, 2004.
- [Conitzer and Sandholm, 2002] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the 18th conference on uncertainty in Artificial Intelligence (UAI-02)*, pages 103–110, 2002.
- [Doyle and Wellman, 1989] J. Doyle and M.P. Wellman. Impediments to Universal Preference-Based Default Theories. In *Proceedings of the 1st conference on principles of knowledge representation and reasoning*, 1989.
- [Kfir-Dahav and Tennenholtz, 1996] Noa E. Kfir-Dahav and Moshe Tennenholtz. Multi-Agent Belief Revision. In *Proceedings of the 6th conference on theoretical aspects of rationality and knowledge (TARK)*, 1996.
- [LaMura and Shoham, 1998] P. LaMura and Y. Shoham. Conditional, Hierarchical Multi-Agent Preferences. In *Proceedings of Theoretical Aspects of Rationality and Knowledge*, pages 215–224, 1998.
- [Mas-Colell et al., 1995] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [May, 1952] Kenneth O. May. A set of independent, necessary and sufficient conditions for simple majority decision. *Econometrica*, 20(4):680–84, 1952.
- [Page et al., 1998] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report, Stanford University, 1998.
- [Resnick and Zeckhauser, 2001] P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. Working Paper for the NBER workshop on empirical studies of electronic commerce, January 2001.
- [Resnick et al., 2000] P. Resnick, R. Zeckhauser, R. Friedman, and E. Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.
- [Tennenholtz, 2004] M. Tennenholtz. Reputation systems: An axiomatic approach. In *Proceedings of the 20th conference on uncertainty in Artificial Intelligence (UAI-04)*, 2004.

Sequential-Simultaneous Information Elicitation in Multi-Agent Systems

Gal Bahar and Moshe Tennenholtz

Faculty of Industrial Engineering and Management
Technion–Israel Institute of Technology
Haifa 32000, Israel

Abstract

We introduce a general setting for information elicitation in multi-agent systems, where agents may be approached both sequentially and simultaneously in order to compute a function that depends on their private secrets. We consider oblivious mechanisms for sequential-simultaneous information elicitation. In such mechanisms the ordering of agents to be approached is fixed in advance. Surprisingly, we show that these mechanisms, which are easy to represent and implement are sufficient for very general settings, such as for the classical uniform model, where agents' secret bits are uniformly distributed, and for the computation of the majority function and other classical threshold functions. Moreover, we provide efficient algorithms for the verification of the existence of the desired elicitation mechanisms, and for synthesizing such mechanisms.

1 Introduction

Information elicitation in multi-agent systems deals with the aggregation of information from agents in order to compute a desired function [Conen and Sandholm, 2001; Sunderam and Parkes, 2003; Parkes, 1999; Boutilier *et al.*, 2003; Smorodinsky and Tennenholtz, 2004; Shoham and Tennenholtz, 2002; McGrew *et al.*, 2003; Halpern and Teague, 2004]. In the general setting we consider in this paper there are n agents (players) who hold some private information (secrets). The agents would like to compute, jointly, a function which input is the vector of n secrets.¹ In addition, there exist a distinguished reliable agent (aka the center), who may be used for extracting information from the agents. The center's aim is to devise a mechanism that will yield to the desired computation. In this paper we consider a strategic setting. We assume that the agents are selfish and are driven by utility maximization considerations. On one hand, they would like to receive the value of the multi-party computation, but on the other hand they may not want to contribute to this computation, but rather free-ride on other agents' efforts. Hence, our

setting fits into the context of mechanism design (see [Mas-Colell *et al.*, 1995], Vol. 23), a subject of great interest to the synergy between AI and game theory (see [Monderer *et al.*, 2000]).

The particular (although general) setting we study is the following. Each agent has a secret which is accessible only to him. However, access to that secret is costly and agents may choose to access it or not. Thus accessing one's own private information becomes a strategic question. This approach generates a natural tension between the socially (and even privately) optimal action, which is to compute the joint function correctly, and agents' incentive to free ride. In order to overcome this tension one may need to design a mechanism to prevent (some or all) agents from free-riding, elicit agents' secrets and execute the desired computation.

As in previous work in this regard (e.g. [Smorodinsky and Tennenholtz, 2004]) the class of functions we consider is that of anonymous functions. An anonymous function is one where the function's value does not depend on the identity of the agents but on the secrets only. In other words, a permutation of the agents' secrets will not change the value of the function. This class of functions is quite elementary and often used in models. Among the anonymous functions are majority, consensus, average, variance, order statistic, percentile and more.

Previous work has shown the power of sequential information elicitation. By approaching the agents sequentially, we can in some cases incent agents to access their secrets, where they would not do so if approached simultaneously. The basic intuition behind this approach is that some agents may be willing to access their secrets since this is not too costly for them, while others will not agree to access their secrets unless they are pivotal (given that their costs for accessing their secrets is high). Sequential mechanisms take advantage of this structure, by carefully selecting the identity of the agent to be approached at each point, given the available information. In this paper we consider a more general approach: agents can be approached both simultaneously and sequentially. That is, we consider sequential mechanisms where at each point a set of agents is approached. The decision on the set of agents to be approached is based on the available information. As it can be shown (and is illustrated in Section 2) this kind of mechanisms, which we refer to as *sequential-simultaneous* mechanisms, allows more effective information

¹This setting is fundamental not only to AI, but to computer science in general; see the discussion in [Linial, 1994].

elicitation than either sequential or simultaneous information elicitation. The major aim of this paper is to identify conditions for the existence of successful sequential-simultaneous elicitation mechanisms, and devising algorithms which output the desired mechanism when exist.

As the reader would notice, our objective is quite ambitious. In particular, a sequential-simultaneous mechanism may be a complex decision tree, which maps any possible history observed to the set of agents to be approached given that history. This gives rise to the idea of considering a particular kind of such mechanisms, which we refer to as *Oblivious Mechanisms*. In an oblivious mechanism the ordering of the agents is fixed in advance and does not depend on previous history. Naturally, if successful elicitation can be established by such a mechanism, then it is also easy to represent and implement. This leads to two types of fundamental questions:

1. Can one provide an efficient algorithm for checking whether, given an information elicitation setting, a desired oblivious sequential-simultaneous elicitation mechanism exist? If one exists, can we provide an efficient procedure which will output the desired mechanism?
2. Are there natural and general settings where the restriction to oblivious mechanisms does not prevent us from the desired elicitation?

The surprising message of this paper is that we can provide highly positive and encouraging answers to both questions! We first prove that if there is an appropriate oblivious mechanism then there is also one which uses a semi-natural ordering, i.e. the mechanism will never approach an agent with cost c before it approaches an agent with cost $c' < c$ (although it may approach them simultaneously). Then we show an efficient algorithm for verifying whether a desired (oblivious) semi-natural ordering mechanism exists; if such mechanism exists, then the desired ordering is obtained as our algorithm's output. Then, we show that the restriction to (oblivious) semi-natural ordering mechanism does not restrict the power of elicitation in some of the most popular contexts. In particular, we show that this holds for the uniform setting (where each agent's secret bit is taken to be 0 or 1 with probability 0.5 each), as well as for the classical majority function and for more general threshold functions. Together, this provides a most powerful and general approach to information elicitation in multi-agent systems, backed up with efficient and effective algorithms.

In Section 2 we introduce the general model, and general sequential-simultaneous mechanisms. We also briefly illustrate the power that these mechanisms buy us when comparing to previous studies where purely sequential or purely simultaneous mechanisms have been considered. In Section 3 we introduce oblivious mechanisms. We then show an efficient algorithm for verifying the existence of an appropriate oblivious elicitation mechanism, and show that a semi-natural ordering mechanism can be computed as the desired result, if such oblivious mechanism exists. We also remark (by means of example) that there are cases where the restriction to oblivious mechanisms may prevent us from obtaining the desired computation; this leads to the study of settings where the ex-

istence of a desired elicitation process implies that it can be done by a corresponding oblivious mechanism, which we can efficiently compute. The uniform model is discussed in Section 4, and the majority function, as well as extensions to other threshold functions are discussed in Section 5. In all cases efficient algorithms are provided.

The proofs of many of our results are non-trivial, and are based on a series of propositions that are omitted from this version of the paper due to lack of space.

2 The General Model

Let $N = \{1, \dots, n\}$, $n \geq 2$, be a finite set of agents. Each agent j has a private secret $s_j \in \{0, 1\}$ that he may compute (i.e. access in order to learn its value). Let $0.5 \leq q < 1$ be the prior probability of $s_j = 1$ ² and assume these events are independent. Agents may compute their own secrets. However, computation is costly and agent j pays $c_j \geq 0$ for computing s_j . Without loss of generality we shall assume $c_1 \leq c_2 \leq \dots \leq c_n$ (in words, agents are ordered by their costs). We will also refer to the agents' costs, ordered from the lowest one to the highest one, as $c_{(0)}, c_{(1)}, \dots, c_{(n-1)}$.

Agents are interested in computing some joint binary parameter (e.g., the majority vote or whether they have a consensus) that depends on the vector of private inputs. Let $G : \{0, 1\}^n \rightarrow \{0, 1\}$ denote the desired computation. Each agent j has a utility of v_j from learning the value of G . We assume $v_j \geq c_j$, otherwise the agent face no dilemma.

It is also possible to use the convention that $v_j = 1$. This is done without loss of generality, as the more general case where $v_j > c_j > 0$, is equivalent to the case where the value of agent j is 1 but the cost is $\frac{c_j}{v_j}$.

A central designer, termed the *center*, elicits the agents' secrets, computes G and reports the computed value of G back to each agent. In this setup each agent faces a dilemma of whether to compute his private secret s_j , at a cost of c_j , or perhaps to submit a guess to the central designer. The desired property of a mechanism is the correct computation of G , which is done through the elicitation of secrets from sufficiently many agents.

One should note that the cost of each agent's computation, c_j , is lower than the gain from computing G , and therefore the socially optimal outcome is to compute. However, free riding of agents may undermine the ability to reach the social optimum.

2.1 Sequential-Simultaneous mechanisms

We now model mechanisms that approach players sequentially, with the possibility to approach simultaneously more than one player each time. We will assume that the communication between the center and the agents is fully revealed to all agents. This can be associated with having broadcast communication.

A sequential-simultaneous mechanism determines the set of agents to be approached at each stage, as a function of the information provided so far. Formally, let $H_i =$

²This is done wlog and for purposes of exposition. The case where $q < 0.5$ is treated similarly.

$\{(a_1, b_1), (a_2, b_2), \dots, (a_i, b_i) | a_i \leq n, \forall 1 \leq k < i : a_k < a_{k+1}, \forall 1 \leq k < i : b_k \leq b_{k+1}, \forall 1 \leq k < i : a_{k+1} - a_k \geq b_{k+1} - b_k\}$ where $v = (j, l)$ stands for the event where j agents have been approached and l 1's (and $j - l$ 0's) have been reported. H_i is the set of histories of length i , for $1 \leq i \leq n$. Let $H_0 = \Lambda$, where Λ is the empty (null) history. Let $H = \bigcup_{i=0}^n H_i$ be the set of all possible histories. A sequential-simultaneous mechanism is a triplet (u, g, f) where $u : H \rightarrow \{1, 2, \dots, n\}$ determines how many agents to approach simultaneously, $g : H \rightarrow 2^N$ determines the agents to be approached, and $f : H \rightarrow \{0, 1, *\}$ is a function that expresses a decision about whether to halt and output either 0 or 1, or continue the elicitation process (denoted by $*$). Note that each agent has 6 actions: Don't compute and report 0, Don't compute and report 1, Compute and report 0, Compute and report 1, Compute and report the true computed value, and Compute and report a false value. Let us denote by Γ the set of actions. A strategy for player j , $x_j : H \rightarrow \Gamma$, assigns an action to each possible history. Let $h_{(k)}(x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n, S)$ be the history that agent k sees when approached, given that $(x_1, \dots, x_{k-1}, x_{k+1}, \dots, x_n)$ is the vector of strategies of the other agents and S is the vector of secrets. Let $x_k(h_{(k)})$ be the strategy of agent k according to the history $h_{(k)}$ he sees when approached. Let $x_k^1(h_{(k)}) = 1$ if and only if the action of agent k according to the history $h_{(k)}$ is to compute the secret, otherwise $x_k^1(h_{(k)}) = 0$, and let $x_k^0(h_{(k)}) = 1$ if and only if agent k reports "1", otherwise $x_k^0(h_{(k)}) = 0$. We get that the utility for agent k according to the strategies and the secrets is: $u_k(x_1, \dots, x_n) = [Prob(G(x_1^0(h_{(1)}), \dots, x_n^0(h_{(n)}))) = G(S)] - [Prob(x_k^1(h_{(k)})) = 1] \cdot c_k$ ³. An equilibrium for the mechanism $\mathcal{A}$, is a vector of n strategies, $(x_1, \dots, x_n)$, such that each agent's strategy is the best response against the other agents' strategies. That is, for each agent k and for each strategy x'_k of agent k : $u_i(x_1, \dots, x_k, \dots, x_n) \geq u_i(x_1, \dots, x'_k, \dots, x_n)$. We seek mechanisms which can compute the true value of G in equilibrium. In fact, it is required that a mechanism computes G with certainty. Therefore we seek mechanisms that induce sufficiently many agents to access and reveal their secrets, in order for G to be computed. Note that in many cases G may be computed with partial information. For example, in the case of a consensus function it is sufficient to elicit information sequentially until we get two different replies, which are truthful.

Given a particular q and $c_1, \dots, c_n$, a mechanism $\mathcal{A}$ is *appropriate* for G , if there exists an equilibrium where G can surely be computed for all vector of agents' secrets. Such an equilibrium is referred to as a computing equilibrium. An algorithm is appropriate if it induces an appropriate mechanism.

Let G be an anonymous function. Consider the directed graph $G' = (V', E')$ where the set of nodes is $V' = \{(i, k) : i, k \in \mathbb{Z}_+, 0 \leq k \leq i \leq n-1\}$, and the set of edges is, $E' = \{((i, k), (i+1, k)) : 0 \leq i \leq n-2, k \leq i\} \cup \{((i, k), (i+1, k+1)) : 0 \leq i \leq n-2, k \leq i\}$.

³The probability distribution is induced by the distribution over S , and q .

Intuitively, this graph describes the possible states of information, when approaching agents while computing the value of an anonymous function, and the possible state transitions. Every state describes how many agents have been approached and how many 1's have been heard so far. Let $\bar{G} = (\bar{V}, \bar{E})$ be the graph induced by reducing G' to include only nodes where the value of the function G can not be determined yet. When referring to a node v , we will abuse notation and refer also to v as an event. Namely, the event $v = (j, l)$ stands for the case that j agents have been approached and l 1's (and $j - l$ 0's) have been reported.

For every node v in $\bar{G}$ we define the following. Let $piv(v, G) = Prob(Z_i | v)$, where Z_i is the event that agent i is pivotal for the function G . Notice that since G is an anonymous function $Prob(Z_i | v)$ is the same for each agent i and therefore $piv(v, G)$ is uniquely defined. When G is clear from the context then denote it $piv(v)$. Note that if $v \notin \bar{V}$, i.e. it is already possible to compute the value of G , then $piv(v) = 0$. Let $Cmax(v) \equiv Max\{c \mid 1 - c \geq piv(v) \cdot q + (1 - piv(v)) \cdot 1\}$. The left side of the above inequality is the utility for an agent with cost c from approaching his secret assuming all other agents approach their secrets. The right side is the utility for the agent when he doesn't approach the secret (assuming all other agents approach their secrets) and guess "1", which is the best choice for him in this case (since $q \geq 0.5$). This inequality implies that: $c \leq 1 - (1 - piv(v)) - q \cdot piv(v)$. This implies that $c \leq (1 - q) \cdot piv(v)$. Hence we get: $Cmax(v) = (1 - q) \cdot piv(v)$. Notice that $Cmax(v)$ is the maximal cost for which it is still rational to access the secret given the information v (assuming the others access their secrets).

The following example illustrates why is it useful to use sequential-simultaneous mechanisms. Consider the following setup. Let G be the majority function, $n = 3$, $q = 0.7$, and $c_{(0)} = 0.12$, $c_{(1)} = 0.12$, $c_{(2)} = 0.29$. We can show: given the above setup there exists a sequential-simultaneous mechanism which is appropriate. On the other hand, given the above setup there does not exist a sequential mechanism which is appropriate. It is also the case that approaching all agents simultaneously is not appropriate.

3 Oblivious mechanisms

An *oblivious mechanism* is a mechanism that partition the agents to groups and each time approach the agents from one group simultaneously without dependency on the history. Both the groups and the ordering between the groups are determined in advance. That is, the agents are partitioned into groups: $A_1, A_2, \dots, A_t$. such that: $\bigcup_{i=1}^t A_i = \{1, 2, \dots, n\}$, $\forall 1 \leq i \neq j \leq t : A_i \cap A_j = \emptyset$ where the group A_1 of agents is approached first, and afterwards the group A_2 of agents is approached, and so on until the group A_t of agents is approached. However, if after a group of agents is approached and reveal their secrets, the event is such that it is already possible to determine the value of the function, then the mechanism stops.

Given a function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , a *semi-natural ordering mechanism* is an oblivious mechanism, where the agents are also sorted accord-

ing to their cost for accessing their secrets. That is, $A_1 = \{c_{(0)}, c_{(1)}, \dots, c_{(|A_1|-1)}\}$,
 $A_2 = \{c_{(|A_1|)}, c_{(|A_1|+1)}, \dots, c_{(|A_1|+|A_2|-1)}\}, \dots,$
 $A_{t-1} = \{c_{(|A_1|+\dots+|A_{t-2}|)}, c_{(|A_1|+\dots+|A_{t-2}|+1)}, \dots,$
 $c_{(|A_1|+\dots+|A_{t-2}|+|A_{t-1}|-1)}\},$
 $A_t = \{c_{(|A_1|+\dots+|A_{t-1}|)}, c_{(|A_1|+\dots+|A_{t-1}|+1)}, \dots, c_{(n-1)}\}$

This section deals with the following: First we prove that whenever there exists an oblivious mechanism which is appropriate, there exists a semi-natural ordering mechanism which is appropriate. We then address the problem of existence: We provide an efficient algorithm, that given an anonymous function G , the c_i 's, and q , will decide if there exist an oblivious mechanism which is appropriate. Moreover, we provide an efficient algorithm that induces an appropriate oblivious mechanism whenever such a mechanism exist.

Given a function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , consider the following algorithm.

The Oblivious Mechanism Algorithm: (OMA)

- (A) Build the graph $G' = (V', E')$ of the function G
- (B) Compute for every node $v \in V' : piv(v)$ and $Cmax(v)$.
- (C) Induce the graph $\bar{G} = (\bar{V}, \bar{E})$ by reducing G' to include only nodes where the value of the function G can not be determined yet.
- (D) $\forall 0 \leq k \leq n-1 : \text{let } L_k = \text{Min}\{Cmax(k, j) | (k, j) \in \bar{V}\}$ i.e. L_k is the minimum $Cmax$ among all the nodes with the same depth k in $\bar{G}$.
- (E) Let $i_1 = 0$.
 $i_2 = \text{Min}\{j | i_1 < j \leq n-1, L_j > L_{i_1}\}.$
 $i_3 = \text{Min}\{j | i_2 < j \leq n-1, L_j > L_{i_2}\}.$
 $\dots$
 $i_r = \text{Min}\{j | i_{r-1} < j \leq n-1, L_j > L_{i_{r-1}}\}.$
 $\dots$
 $i_k = \text{Min}\{j | i_{k-1} < j \leq n-1, L_j > L_{i_{k-1}}\}$ where i_k satisfies $\forall j : i_k < j \leq n-1 : L_j \leq L_{i_k}$
- (F) Let $g_1 = i_2 - i_1, g_2 = i_3 - i_2, \dots, g_{r-1} = i_r - i_{r-1}, \dots, g_{k-1} = i_k - i_{k-1}, g_k = n - i_k.$
- (G) Sort the agents according to their costs of getting to their secrets and get: $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$
- (H) Check if the following set of conditions, marked as *Cond1*, is satisfied: $c_{(0)}, c_{(1)}, \dots, c_{(g_1-1)} \leq L_{i_1}$
 $c_{(g_1)}, c_{(g_1+1)}, \dots, c_{(g_1+g_2-1)} \leq L_{i_2}$
 $c_{(g_1+g_2)}, c_{(g_1+g_2+1)}, \dots, c_{(g_1+g_2+g_3-1)} \leq L_{i_3}$
 $\dots$
 $c_{(g_1+g_2+\dots+g_{k-2})}, \dots, c_{(g_1+g_2+\dots+g_{k-1}-1)} \leq L_{i_{k-1}}$
 $c_{(g_1+g_2+\dots+g_{k-2}+g_{k-1})}, \dots, c_{(g_1+g_2+\dots+g_{k-1}+g_k-1)} \leq L_{i_k}$

Consider the set of conditions *Cond1* in step (H) of the OMA above. We can now prove the following general results:

Theorem 3.1: Consider an anonymous function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q . If there exists an oblivious mechanism which is appropriate then *Cond1* is satisfied.

Theorem 3.2: Consider an anonymous function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , if *Cond1* is satisfied

then there exists a semi-natural ordering mechanism which is appropriate.

Note that given an anonymous function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , if there exists a semi-natural ordering mechanism which is appropriate then there exists an oblivious mechanism which is appropriate. Therefore, combining the above theorems and observations we get:

Theorem 3.3: Consider an anonymous function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q . The three following claims are equivalent: (1) There exists an oblivious mechanism which is appropriate. (2) *Cond1* is satisfied. (3) There exists a semi-natural ordering mechanism which is appropriate.

Now we can provide a complete efficient procedure that given a function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , will check if there exists an oblivious mechanism which is appropriate. Moreover, if there exist an oblivious mechanism which is appropriate, this procedure will induce an appropriate semi-natural ordering mechanism. In order to do so we repeat steps (A) - (G) of the OMA. Then, as in step (H) of of OMA we check whether *Cond1* is satisfied; if *Cond1* is not satisfied then the procedure will return "there isn't any oblivious algorithm which is appropriate" and finish; otherwise, if *Cond1* is satisfied then the mechanism will approach the agents sequentially, in the following order, where at each phase we approach simultaneously a set of agents as described below (the agents are referred to by their costs):

- $c_{(0)}, c_{(1)}, \dots, c_{(g_1-1)}.$
- $c_{(g_1)}, c_{(g_1+1)}, \dots, c_{(g_1+g_2-1)}.$
- $c_{(g_1+g_2)}, c_{(g_1+g_2+1)}, \dots, c_{(g_1+g_2+g_3-1)}.$
- $\dots$
- $c_{(g_1+\dots+g_{k-2})}, c_{(g_1+\dots+g_{k-2}+1)}, \dots,$
 $c_{(g_1+\dots+g_{k-2}+g_{k-1}-1)}$
- $c_{(g_1+\dots+g_{k-1})}, c_{(g_1+\dots+g_{k-1}+1)}, \dots,$
 $c_{(g_1+\dots+g_{k-1})} = c_{(n-1)}.$

To sum up, given a function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , the OMA algorithm verifies the existence of an appropriate oblivious mechanism. Such a mechanism, if exists, is induced by the above procedure.

Oblivious mechanisms are natural and can be easily implemented. It can be shown (by means of an example; details omitted due to lack of space) that the existence of an appropriate mechanism does not necessary imply the existence of an oblivious mechanism which is appropriate. However, surprisingly, in the next sections we show that there are some important and central setups where oblivious mechanisms are sufficient for efficient and effective information elicitation. Given the convenient and tractable structure of these mechanisms, this provides powerful tools for information elicitation. Namely, this turned out to be the case for the classical uniform model discussed in Section 4, and for the majority function, and moreover in the context of threshold functions, as discussed in Section 5.

4 The Uniform Model

Consider the case where $q = 0.5$ and G is any anonymous function. The assumption that the distribution is uniform is

most standard in many basic studies in this context. Consider the following algorithm:

The Uniform Semi-Natural Ordering Algorithm (USNO):

- (A) Build the graph $G' = (V', E')$ of the function G .
- (B) Compute $\forall v \in V' : piv(v)$ and $Cmax(v)$.
- (C) Induce the graph $\bar{G} = (\bar{V}, \bar{E})$ by reducing G' to include only nodes where the value of the function G can not be determined.
- (D) Denote as Υ the following path in $\bar{G}$: $(0, 0) = v'_0 \rightarrow v'_1 \rightarrow v'_2 \rightarrow \dots \rightarrow v'_j = (j, l) \rightarrow v'_{j+1} \rightarrow \dots \rightarrow v'_{n-1}$ where $(0, 0) = v'_0 \rightarrow v'_1 \rightarrow v'_2 \rightarrow \dots \rightarrow v'_j = (j, l)$ satisfies that $v'_j = (j, l)$ has only one son in $\bar{G}$, and $\forall 0 \leq i \leq j-1 : piv(v'_i) \geq piv(v'_{i+1})$. Let $v'_j = (j, l) \rightarrow v'_{j+1} \rightarrow \dots \rightarrow v'_{n-1}$ be the suffix from v'_j to the only leaf in $\bar{G}$ reachable from v'_j . The existence of such a path is induced by a series of propositions omitted from the paper due to lack of space.
- (E) Along Υ , $\forall 0 \leq i \leq n-1$: We will mark $\Psi_i \equiv Cmax(v'_i)$
- (F) Check which is the first node on Υ that satisfies $\Psi_t > \Psi_0$ ⁴. That is, $t = \min\{k | 1 \leq k \leq n-1, \Psi_k > \Psi_0\}$. (the fact that $t > j$ is implied by the above mentioned propositions).
- (G) Sort the agents according to their costs of getting to their secrets and get: $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$
- (H) Check if the following is satisfied: $c_{(0)} \leq \Psi_0, c_{(1)} \leq \Psi_0, \dots, c_{(t-1)} \leq \Psi_0, c_{(t)} \leq \Psi_t, c_{(t+1)} \leq \Psi_{t+1}, \dots, c_{(n-1)} \leq \Psi_{n-1}$. If the above is not satisfied then finish and return "there is no appropriate mechanism for the problem", else continue to phase (I).
- (I) First approach simultaneously the t agents with the lowest costs, i.e. $c_{(0)}, c_{(1)}, \dots, c_{(t-1)}$. Afterwards approach each time one agent with the lowest cost among those left, i.e. first approach $c_{(t)}$, then approach $c_{(t+1)}$, etc, when $c_{(n-1)}$ is approached last.

We can now prove:

Theorem 4.1: Consider an anonymous function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and $q = 0.5$, and let $\Psi_i \equiv Cmax(v'_i)$ for every $0 \leq i \leq n-1$ along the path Υ . Let $t = \min\{k | 1 \leq k \leq n-1, \Psi_k > \Psi_0\}$. Then there exists an appropriate algorithm, if and only if: $c_{(0)} \leq \Psi_0, c_{(1)} \leq \Psi_0, \dots, c_{(t-1)} \leq \Psi_0, c_{(t)} \leq \Psi_t, c_{(t+1)} \leq \Psi_{t+1}, \dots, c_{(n-1)} \leq \Psi_{n-1}$.

Theorem 4.2: Consider an anonymous function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and $q = 0.5$, if there exists an appropriate mechanism then the USNO algorithm induces an appropriate oblivious mechanism.

Corollary 4.1: Consider an anonymous function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and $q = 0.5$. The USNO handles the problem of existence, i.e the USNO

⁴It is guaranteed that there exists such a node if G is not the parity or the $\neg$ parity function. These functions are treated separately, leading to similar results.

will efficiently compute whether there exists any appropriate mechanism. If such a mechanism exists, the USNO will output an appropriate semi-natural ordering mechanism.

5 Computing the majority function

The majority function is central to voting theory, and as a result to the related multi-agent systems and distributed computing literature. Consider the case where n is odd, and $q > 0.5$ and the function we would like to compute is the majority function,⁵ i.e. if $\lceil \frac{n-1}{2} \rceil + 1$ or more of the agents' secrets are 1's then the value of the function will be 1, else the value will be 0.

We will now present an efficient algorithm that induces an appropriate mechanism if such a mechanism exists. The output mechanism will be an oblivious, semi-natural ordering mechanism. Denote by Γ the following path in the graph associated with the majority function:

$$v_0^* = (0, 0) \rightarrow (1, 1) \rightarrow (2, 2) \rightarrow \dots \rightarrow (\frac{n-1}{2}, \frac{n-1}{2}) \rightarrow (\frac{n-1}{2} + 1, \frac{n-1}{2}) \rightarrow (\frac{n-1}{2} + 2, \frac{n-1}{2}) \rightarrow \dots \rightarrow (n-1, \frac{n-1}{2}) = v_{n-1}^*$$

The Semi-Natural Ordering Algorithm (SNO):

- (A) Compute: $\forall 0 \leq i \leq n-1 : piv(v_i^*)$ and $Cmax(v_i^*)$, along Γ . $\forall 0 \leq i \leq n-1$: We will mark $\Psi_i \equiv Cmax(v_i^*)$
- (B) Check which is the first node on Γ that satisfies $\Psi_j > \Psi_0$. That is, $j = \min\{k | 1 \leq k \leq n-1, \Psi_k > \Psi_0\}$ ⁶.
- (C) Sort the agents according to their costs of getting to their secrets and get: $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$
- (D) Check if the following is satisfied: $c_{(0)} \leq \Psi_0, c_{(1)} \leq \Psi_0, \dots, c_{(j-1)} \leq \Psi_0, c_{(j)} \leq \Psi_j, c_{(j+1)} \leq \Psi_{j+1}, \dots, c_{(n-1)} \leq \Psi_{n-1}$. If the above is not satisfied then finish and output "there is no appropriate mechanism for the problem", else continue to phase (E).
- (E) First approach simultaneously the j agents with the lowest costs, i.e. $c_{(0)}, c_{(1)}, \dots, c_{(j-1)}$. Afterwards approach each time one agent with the lowest cost among those left, i.e. first approach $c_{(j)}$, then approach $c_{(j+1)}$, etc, when $c_{(n-1)}$ is approached last.

We can now show:

Theorem 5.1: Consider the majority function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , and let $\Psi_i \equiv Cmax(v_i^*)$ for every $0 \leq i \leq n-1$ along the path Γ . Let $j = \min\{k | 1 \leq k \leq n-1, \Psi_k > \Psi_0\}$. Then there exists an appropriate algorithm, if and only if: $c_{(0)} \leq \Psi_0, c_{(1)} \leq \Psi_0, \dots, c_{(j-1)} \leq \Psi_0, c_{(j)} \leq \Psi_j, c_{(j+1)} \leq \Psi_{j+1}, \dots, c_{(n-1)} \leq \Psi_{n-1}$.

Theorem 5.2: Consider the majority function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , if there exists an appropriate mechanism then the SNO algorithm induces an appropriate oblivious mechanism.

⁵Our results in this section can be extended in a straightforward manner to the case where $q \leq 0.5$ but not hold to the case where n is even; the case where n is even will be discussed later, when we extend the discussion to general threshold functions.

⁶As before, the fact that such a j exists is implied by a series of propositions characterizing the properties of Γ

Corollary 5.1: Consider the majority function G , agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$, and q , the SNO handles the problem of existence, i.e. the SNO will efficiently compute whether there exists any appropriate mechanism. If such a mechanism exists, the SNO will output an appropriate semi-natural ordering mechanism.

The computation of majority when $q < 0.5$: Consider the case where n is odd, and the function that we would like to compute is the majority function where $q < 0.5$. In this case we can use similar techniques to the ones we used for the computing of majority where $q > 0.5$. The main difference is that instead of the path Γ we will consider the path Γ' : $v_0 = (0, 0) \rightarrow (1, 0) \rightarrow (2, 0) \rightarrow \dots \rightarrow (\frac{n-1}{2}, 0) \rightarrow (\frac{n-1}{2} + 1, 1) \rightarrow (\frac{n-1}{2} + 2, 2) \rightarrow \dots \rightarrow (n-1, \frac{n-1}{2}) = v'_{n-1}$. Note that the case where $q = 0.5$ is covered by the uniform model section.

5.1 Threshold functions

Given an anonymous function G . Let $G(T)$ be the value of the function G when there are T "1"s and $n - T$ "0"s. G is considered as a *threshold function* if there exists $1 \leq r \leq n$ such that: $G(0) = G(1) = \dots = G(r-1) \neq G(r) = G(r+1) = \dots = G(n)$, i.e. whenever there are less than r 1's among the agents then the function gets a particular value, and where there are r or more 1's the value of the function will be the other value. The majority function for example is a threshold function with $r = \lceil \frac{n+1}{2} \rceil$ where $G(0) = 0$.

We can now prove the following general theorems:

Theorem 5.3: Consider an anonymous threshold function G with threshold r where $\lceil \frac{n+1}{2} \rceil \leq r \leq n$, $q \leq 0.5$ and agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$. There exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

Theorem 5.4: Consider an anonymous threshold function G with threshold r where $\lceil \frac{n+1}{2} \rceil \leq r \leq n$, $q \geq \frac{r-1}{n-1}$ and agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$. There exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

Similarly to theorems 5.3 and 5.4, we can also show:

Theorem 5.5: Consider an anonymous threshold function G with threshold r where $1 \leq r \leq \lfloor \frac{n+1}{2} \rfloor$, and $q \leq \frac{n-r}{n-1}$ or $q \geq 0.5$ and agents' costs $c_{(0)} \leq c_{(1)} \leq \dots \leq c_{(n-1)}$. Then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

Further discussion about majority: The majority function is a particular case of a threshold function, where $r = \lceil \frac{n+1}{2} \rceil$. However when n is odd, we get $r = \frac{n+1}{2}$. Theorem 5.3 implies that if $q \leq 0.5$ then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate. Theorem 5.4 implies that if $q \geq \frac{r-1}{n-1} = \frac{\frac{n+1}{2}-1}{n-1} = \frac{n-1}{2(n-1)} = 0.5$ then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate. Hence when n is odd the above holds for every q (as we have already shown before).

Consider the case where G is the majority function and n is even. This implies that $r = \frac{n+2}{2}$. Theorem 5.3 implies that if $q \leq 0.5$ then there exist an appropriate mechanism

if and only if there exist a semi-natural ordering mechanism which is appropriate. Theorem 5.4 implies that if $q \geq \frac{r-1}{n-1} = \frac{\frac{n+2}{2}-1}{n-1} = \frac{\frac{n}{2}}{n-1} = 0.5 \cdot \frac{n}{n-1}$ then there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate. However, when n is large this implies that for almost every q there exist an appropriate mechanism if and only if there exist a semi-natural ordering mechanism which is appropriate.

References

- [Boutilier et al., 2003] C. Boutilier, R. Das, J.O. Kephart, G. Tesauro, and W.E. Walsh. Cooperative negotiation in autonomic systems using incremental utility elicitation. In *Proceedings of UAI-03*, pages 89–97, 2003.
- [Conen and Sandholm, 2001] W. Conen and T. Sandholm. Minimal preference elicitation in combinatorial auctions. In *IJCAI Workshop on Economic Agents, Models, and Mechanisms*, Seattle, WA, August 6th., 2001.
- [Halpern and Teague, 2004] J. Halpern and V. Teague. Rational secret sharing and multiparty computation. In *Proc. of STOC 2004*, 2004.
- [Linial, 1994] N. Linial. Game-Theoretic Aspects of Computer Science. In R.J. Aumann and S. Hart, editors, *Handbook of Game Theory with Economic Applications*, Vol. II, Chapter 38, pages 1340–1395. North-Holland, 1994.
- [Mas-Colell et al., 1995] A. Mas-Colell, M.D. Whinston, and J.R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [McGrew et al., 2003] R. McGrew, R. Porter, and Y. Shoham. Towards a general theory of non-cooperative computation. In *Proc. of the 9th conference of theoretical aspects of rationality and knowledge (TARK 2003)*, pages 59–71, 2003.
- [Monderer et al., 2000] D. Monderer, M. Tennenholtz, and H. Varian. Game theory and artificial intelligence. Special issue of Games and Economic behavior, 2000.
- [Parkes, 1999] D. C. Parkes. Optimal auction design for agents with hard valuation problems. In *Proc. IJCAI-99 Workshop on Agent Mediated Electronic Commerce*, pages 206–219, 1999.
- [Shoham and Tennenholtz, 2002] Y. Shoham and M. Tennenholtz. Non-cooperative computation: Boolean functions with correctness and exclusivity. Technical Report, Stanford & Technion; To appear in Theoretical Computer Science, 2002.
- [Smorodinsky and Tennenholtz, 2004] Rann Smorodinsky and Moshe Tennenholtz. Sequential information elicitation in multi-agent systems. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI-04)*, pages 528–535, 2004.
- [Sunderam and Parkes, 2003] Aditya V Sunderam and David C. Parkes. Preference elicitation in proxied multi-attribute auctions. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, pages 214–215, 2003.

Regret-based Utility Elicitation in Constraint-based Decision Problems

Craig Boutilier

Dept. of Computer Science
University of Toronto
cebly@cs.toronto.edu

Relu Patrascu

Dept. of Computer Science
University of Toronto
relu@cs.toronto.edu

Pascal Poupart

School of Computer Science
University of Waterloo
ppoupart@cs.uwaterloo.ca

Dale Schuurmans

Dept. of Computing Science
University of Alberta
dale@cs.ualberta.ca

Abstract

We propose new methods of preference elicitation for constraint-based optimization problems based on the use of minimax regret. Specifically, we assume a constraint-based optimization problem (e.g., product configuration) in which the objective function (e.g., consumer preferences) are unknown or imprecisely specified. Assuming a graphical utility model, we describe several elicitation strategies that require the user to answer only binary (bound) queries on the utility model parameters. While a theoretically motivated algorithm can provably reduce regret quickly (in terms of number of queries), we demonstrate that, in practice, heuristic strategies perform much better, and are able to find optimal (or near-optimal) configurations with far fewer queries.

1 Introduction

The development of automated decision software is a key focus within decision analysis [21; 16] and AI [7; 3]. To deal with different users, some form of preference elicitation must be undertaken in order to capture specific user preferences to a sufficient degree to allow an (approximately) optimal decision to be taken. In this work, we study the problem of preference elicitation in *constraint-based optimization* (CBO). CBO provides a natural framework for specifying and solving many decision problems, such as configuration tasks [15], in which hard constraints capture options available to a customer and an objective (or utility) function reflects customer preferences. Explicit formulation as a mathematical program or using a soft constraint framework [17; 2] has been successfully used to model such problems.

Unfortunately, the requirement of complete utility information demanded by CBO is often problematic. For instance, users may have neither the ability nor the patience to provide full utility information to a system. Furthermore, in many if not most instances, an optimal decision (or some approximation thereof) can be determined with a very partial specification of the user's utility function. As such, it is imperative that preference elicitation procedures be designed that focus on the relevant aspects of the problem (e.g., by ignoring infeasible parts of utility space, or utility for outcomes provably dominated given current information).

Our framework is as follows. We assume a set of (hard) constraints together with a graphical utility model [1; 4] cap-

turing user preferences. While the structure of the utility model is known, the parameters of this utility model are imprecise, given by upper and lower bounds. Adopting the minimax regret model of [5], a robust decision can be made with respect to this utility uncertainty, by choosing the *minimax optimal configuration*. This is the solution the user would regret the least should an adversary impose a utility function consistent with our knowledge of the user's preferences. If regret is unacceptably high, we query the user for more information about their preferences, until the worst-case error (regret) is small enough (zero if optimality is required).

In this work, we elicit preferences using *bound queries* (a local form of *standard gamble queries* [9])—that provide tighter upper or lower bounds on the utility parameters—since these are reasonably easy for users to assess and have been studied extensively in the decision analysis literature. We develop several new strategies for elicitation using bound queries whose aim is to reduce the worst-case error (i.e., get guaranteed improvement in decision quality) with as few queries as possible. These include strategies with good theoretical guarantees (related to polyhedral methods in conjoint analysis [18; 11]), as well as several heuristic methods that perform better empirically. We also show that one of these strategies is largely unaffected by computational approximation of the required minimax solutions.

2 Problem Formulation

We assume some system is charged with making or recommending a decision on behalf of a user, for example, configuring a (multiattribute) product for a consumer (e.g., the choice of a car and options). However, since user preferences vary, an appropriate configuration requires that the system interact with the user to determine enough about her preferences over feasible alternatives to make a good (or possibly optimal) choice. Thus, the system must engage in some form of *preference elicitation*. This basic problem lies at the heart of considerable work in multiattribute utility theory [12; 20; 14] and the theory of consumer choice (such as conjoint analysis [18; 11]). Our approach differs from classic approaches in several important respects, as we emphasize below. Most importantly, our decision model and elicitation strategies will be driven by the minimax regret criterion. To present our results, we first need to review previous work on minimax regret and the framework of [5] in more detail.

2.1 Optimization with Graphical Utility Models

Following [5], we assume a finite set of *attributes* $\mathbf{X} = \{X_1, X_2, \dots, X_N\}$ with finite domains, characterizing a set of choices available to a decision maker (or consequences thereof). These might be, say, car options such as *make*, *engine size*, *transmission type*, etc. An assignment $\mathbf{x} \in \text{Dom}(\mathbf{X})$ is often referred to as a *state*. For ease of presentation, we assume attributes are boolean. We also have a set of *hard constraints* $\mathcal{C}$ over these attributes. Each constraint C_ℓ , $\ell = 1, \dots, L$, is defined over a subset of attributes $\mathbf{X}[\ell] \subset \mathbf{X}$, and thus induces a set of legal configurations of attributes in the subset $\mathbf{X}[\ell]$ (e.g., the set of products that can be proposed). We assume that the constraints C_ℓ are represented in some logical form and can be expressed compactly (e.g., $X_1 \wedge X_2 \supset \neg X_3$). For instance, we might have that model *Passat* and engine *2.8T* do not allow transmission *5Speed-Man*. We let $\text{Feas}(\mathbf{X})$ denote the subset of *feasible states* satisfying $\mathcal{C}$.

Suppose the system had access to the user's utility function $u : \text{Dom}(\mathbf{X}) \rightarrow \mathbf{R}$, representing a user's strength of preference for various configurations (e.g., we might view this as what they are willing to pay). The constraint-based optimization (CBO) problem is to find an optimal feasible state $\mathbf{x}^*$:

$$\mathbf{x}^* \in \arg \max_{\mathbf{x} \in \text{Feas}(\mathbf{X})} u(\mathbf{x}).$$

For multiattribute problems of this type, one generally assumes some utility function structure [12; 8]. In this work, we adopt a *generalized additive independence (GAI)* model [8; 1]. Specifically, we assume that the utility function can be written as the sum of K local utility functions, or *factors*, over small sets of variables:

$$u(\mathbf{x}) = \sum_{k \leq K} f^k(\mathbf{x}[k]). \quad (1)$$

Here each function f^k depends only on a local family of attributes $\mathbf{X}[k] \subset \mathbf{X}$. We denote by $\mathbf{x}[k]$ the restriction of state $\mathbf{x}$ to the attributes in $\mathbf{X}[k]$. This model is attractive due to its naturalness and generality (encompassing both linear models [12] and UCP-nets [4] as special cases). The problem of finding an optimal configuration can be formulated as an integer program (IP), and can often be solved very effectively.

2.2 Minimax Regret

While many approaches to elicitation focus on obtaining full utility information, it will often be infeasible, undesirable, or unnecessary to extract a complete utility function from the user [20]. As a consequence, a system must make decisions in the face of *incompletely* specified utility functions. Several approaches have been proposed for representing utility uncertainty, as well as making decisions given this uncertainty. For example, Bayesian methods quantify uncertainty about preferences probabilistically [7; 3]. Other techniques simply pose constraints on the set of possible utility functions and use various criteria to find or reduce the set of decisions or otherwise direct elicitation; for example, one can identify Pareto optimal [21] or dominant alternatives [20; 13; 14], or decisions that minimize regret [4; 16; 19; 5].

In this paper we adopt the minimax regret criterion, following the formulation of [5]. Unlike Bayesian methods, minimax regret recommends decisions given only a *set* of possible utility functions rather than a probabilistic prior. Thus it is suitable when reasonable priors are hard to formulate and does not require the computational approximations often needed in reasoning with complex priors [7; 3]. Minimax regret also allows recommendation of decisions that are robust in the face of utility function uncertainty since it provides tight bounds on the worst-case error, which is appealing in many circumstances (and unlike many schemes proffers a specific decision rather than, say, a Pareto optimal set). Previously this has been found to be an attractive criterion in real-world procurement settings [6]. As we will see, regret is also very effective in focusing elicitation effort even when priors are available [19].

Suppose the utility function for a CBO problem is unknown, but constraints on its parameters (e.g., in the form of bounds) are available and some decision must be recommended.¹ The minimax regret decision criterion suggests that one adopt the (feasible) assignment $\mathbf{x}$ that obtains minimum max-regret, where max-regret is the largest quantity by which one could "regret" choosing $\mathbf{x}$ (while allowing the utility function to vary within the bounds). More formally, let $\mathcal{U}$ denote the set of feasible utility functions. We refer to a pair $\langle \mathcal{C}, \mathcal{U} \rangle$, where $\mathcal{C}$ is a set of configuration constraints, as an *imprecise CBO problem*. The *pairwise regret* of state $\mathbf{x}$ with respect to $\mathbf{x}'$ over feasible utility set $\mathcal{U}$ is defined as

$$R(\mathbf{x}, \mathbf{x}', \mathcal{U}) = \max_{u \in \mathcal{U}} \{u(\mathbf{x}') - u(\mathbf{x})\}, \quad (2)$$

which is the most our system could regret choosing $\mathbf{x}$ for the user instead of $\mathbf{x}'$ (e.g., if an adversary could impose any utility function in $\mathcal{U}$). The *maximum regret* of decision $\mathbf{x}$ is:

$$MR(\mathbf{x}, \mathcal{U}) = \max_{\mathbf{x}' \in \text{Feas}(\mathbf{X})} R(\mathbf{x}, \mathbf{x}', \mathcal{U}) \quad (3)$$

The *minimax regret* of feasible utility set $\mathcal{U}$ is:

$$MMR(\mathcal{U}) = \min_{\mathbf{x} \in \text{Feas}(\mathbf{X})} MR(\mathbf{x}, \mathcal{U}) \quad (4)$$

If the only information we have about a user's utility function is that it lies in the set $\mathcal{U}$, then a minimax-optimal decision $\mathbf{x}^*$ (i.e., $\mathbf{x}^*$ s.t. $MR(\mathbf{x}^*, \mathcal{U}) = MMR(\mathcal{U})$) minimizes the worst-case loss w.r.t. possible realizations of utility $u \in \mathcal{U}$.

Computation of minimax regret in CBO problems requires care; the explicit minimization in Eq. 4 is infeasible. Fortunately, one can formulate it in a manner that exploits the graphical structure of the utility model, thereby admitting (in practice) computationally tractable solution [5]. The procedure of [5] assumes an imprecise CBO problem with factors f^k , $k \leq K$, defined over local families $\mathbf{X}[k]$. The parameters of this utility function are denoted by $u_{\mathbf{x}[k]} = f^k(\mathbf{x}[k])$, where $\mathbf{x}[k]$ ranges over $\text{Dom}(\mathbf{X}[k])$. Upper and lower bounds on each of these parameters are assumed, denoted by $u_{\mathbf{x}[k]}^\uparrow$ and $u_{\mathbf{x}[k]}^\downarrow$, respectively. Effective computation of pairwise regret, max regret and minimax regret is possible by exploiting structure in the constraints and graphical utility model.

¹These constraints reflect knowledge of the user's utility function, generally obtained through elicitation as we elaborate below.

Input: imprecise CBO problem, worst-case error tolerance τ .

1. Compute minimax regret value mmr
2. Repeat until $mmr < \tau$
 - (a) Ask bound query q about some utility parameter $u(\mathbf{x}[k])$.
 - (b) If $u(\mathbf{x}[k]) \leq q$ then lower $u_{\mathbf{x}[k]} \uparrow$ to q .
 - (c) Otherwise raise $u_{\mathbf{x}[k]} \downarrow$ to q
 - (d) recompute mmr

Table 1: General form of the interactive elicitation procedure.

In particular, the minimax optimization in Eq. 4 is rewritten as as minimization with an infinite number of constraints. A constraint generation procedure is used to generate constraints incrementally until all (finitely many) active constraints are enumerated. The mixed integer programs (MIPs) required to solve both the main minimization and the generation of the most violated constraint are compact, with a number of variables linear in the size (number of parameters) of the GAI model. An important property of this procedure is that it generates both an optimal solution $\mathbf{x}^*$ and an adversarial *witness* $\mathbf{x}^w$ for the current $\mathcal{U}$: $\mathbf{x}^w$ is the assignment that the maximizes regret of $\mathbf{x}^*$ in Eq. 4 (as the $\mathbf{x}'$ variable in Eq. 3). We refer to [5] for further algorithm details. The procedure was shown to handle significant practical problems, generally generating very few constraints, with solution times ranging from fractions of a second to 1000 seconds [5].

In practice, since minimax regret will be computed between elicitation queries, it is critical that minimax regret be estimated in a relatively short period of time (say 5 seconds). For this reason, we propose several improvements to the procedure of [5] that can speed up regret computation with elicitation in mind: (1) The constraint generation procedure for solving the MIP can be accelerated by simply finding a feasible $\mathbf{x}$ given the current set of constraints, rather than explicitly searching for a minimax optimal $\mathbf{x}$ given current constraints. (2) Since minimax regret is computed incrementally by generating constraints, it has an anytime nature and early stopping can be used. This has the effect that some violated constraints may not have been generated, but the “early” solution provides a lower bound on true minimax regret. We can also compute an upper bound by computing the max regret of the $\mathbf{x}$ found for the last MIP solved. These bounds are often tight enough to provide good elicitation guidance. (3) The minimax regret problem solved after receiving a response to one query is very similar to that solved before posing the query. As such, one can “seed” the minimax procedure invoked after a query with the constraints generated at the previous step. In this way, typically, only a few extra constraints are generated during each minimax computation.

While we focus on the use of upper and lower bounds on utility parameters, the procedures described here can be adapted to problems with arbitrary linear constraints over utility parameters. Handling such constraints is important when *comparison queries* are used (see below). With this background in place, we can now turn to elicitation.

3 Elicitation Strategies

We consider an interactive process in which the decision software queries the user for information about her utility func-

tion, refining initial bounds on the parameters, until minimax regret reaches an acceptable level τ .² Table 1 summarizes the general form of the interactive elicitation procedure.

3.1 Bound Queries

The types of queries we consider are *bound queries* in which we ask the user whether one of her utility parameters lies above a certain value. A positive response raises the lower bound on that parameter, while a negative response lowers the upper bound: in both cases, uncertainty is reduced.³

While users often have difficulty assessing numerical parameters, they are typically better at comparing outcomes [12; 9]. However, a bound query can be viewed as a local form of a *standard gamble query* (SGQ), commonly used in decision analysis; these, in fact, ask for comparisons. An SGQ for a specific state $\mathbf{x}$ asks the user if she prefers $\mathbf{x}$ to a gamble in which the best outcome $\mathbf{x}_\top$ occurs with probability l and the worst $\mathbf{x}_\perp$ occurs with probability $1 - l$ [12]. A positive response puts a lower bound on the utility of $\mathbf{x}$, and a negative response puts an upper bound. Calibration is attained by the use of common best and worst outcomes across all queries (and numerical assessment is restricted to evaluating probabilities). The foundations of such queries can be made precise using results of Fishburn [8]; we defer details for space reasons (see [10] for more on elicitation in GAI networks).⁴

Our query strategies rely on the following definitions.

Defn 1 Let $\langle \mathcal{C}, \mathcal{U} \rangle$ be an imprecise CBO problem. An *optimistic state* $\mathbf{x}^o$, a *pessimistic state* $\mathbf{x}^p$, and a *most uncertain state* $\mathbf{x}^{mu}$ are any states satisfying:

$$\mathbf{x}^o \in \arg \max_{\mathbf{x} \in \text{Feas}(\mathbf{X})} \max_{u \in \mathcal{U}} u(\mathbf{x})$$

$$\mathbf{x}^p \in \arg \max_{\mathbf{x} \in \text{Feas}(\mathbf{X})} \min_{u \in \mathcal{U}} u(\mathbf{x})$$

$$\mathbf{x}^{mu} \in \arg \max_{\mathbf{x} \in \text{Feas}(\mathbf{X})} \max_{u, u' \in \mathcal{U}} \{u(\mathbf{x}) - u'(\mathbf{x})\}$$

An optimistic state is a feasible state with the greatest upper bound on utility. A pessimistic state has the greatest lower bound on utility. A most uncertain state has the greatest difference between its upper and lower bounds. Each of these states can be computed in a single optimization by setting the parameters of the utility model to their upper bounds, their lower bounds, or their difference, and solving the corresponding (precise) CBO problem.

In this framework, the goal of an elicitation strategy is to reduce minimax regret using as few queries as possible. The challenge is to select such queries efficiently—avoiding intractabilities inherent in outcome enumeration and lookahead, while nevertheless reducing minimax regret effectively.

²We could insist that regret reaches zero (i.e., a provably optimal solution), or stop when regret reaches a point where further improvement is outweighed by the cost of further interaction.

³If the user’s true value is close to the query point, she may feel “roughly indifferent;” in this case we could impose a constraint that the true value is “close” (e.g., within some ϵ) to this point.

⁴While we focus on bound queries, other queries are quite natural. Comparison queries ask if one state $\mathbf{x}$ is preferred to another $\mathbf{x}'$. A response imposes a linear constraint on utility parameters. Regret computation must then take the general form alluded to above.

3.2 The Halve Largest Gap Strategy

The first query strategy we consider is the *halve largest gap (HLG)* strategy. It asks a query at the midpoint of the largest interval, i.e., corresponding to the parameter $\mathbf{x}[k]$ with the largest gap between its upper and lower bounds. This is motivated by theoretical considerations based on simple worst-case bounds on minimax regret. Define, respectively, the *gap* of a utility parameter $u(\mathbf{x}[k])$, the *span* of factor f^k and *maxspan* of a utility model $\mathcal{U}$ as follows:⁵

$$\text{gap}(\mathbf{x}[k]) = u_{\mathbf{x}[k]}^{\uparrow} - u_{\mathbf{x}[k]}^{\downarrow} \quad (5)$$

$$\text{span}(f^k) = \max_{\mathbf{x}[k] \in \text{Dom}(\mathbf{X}[k])} \text{gap}(\mathbf{x}[k]) \quad (6)$$

$$\text{maxspan}(\mathcal{U}) = \sum_k \text{span}(f^k) \quad (7)$$

The quantity *maxspan* measures the largest gap between the upper and lower utility bound, regardless of feasibility. We can show that this quantity bounds minimax regret:

Proposition 1 For any $\langle \mathcal{C}, \mathcal{U} \rangle$, $\text{MMR}(\mathcal{U}) \leq \text{maxspan}(\mathcal{U})$.

Since $\text{MMR}(\mathcal{U}) \leq \text{MR}(\mathbf{x}^o, \mathcal{U})$ and for any optimistic state $\mathbf{x}^o$ we have $\text{MR}(\mathbf{x}^o, \mathcal{U}) \leq \text{maxspan}(\mathcal{U})$, the result follows.⁶

This suggests an obvious query strategy, the HLG method, in which a bound query is asked of the parameter p with the largest gap, at the midway point of its interval, $(p^{\uparrow} - p^{\downarrow})/2$. This method ensures rapid reduction in max regret:

Proposition 2 Let $\mathcal{U}$ be an uncertain utility model with n parameters and let $m = \text{maxspan}(\mathcal{U})$. After $n \log(m/\varepsilon)$ queries by HLG, minimax regret is no greater than ε .

In the worst case, no query strategy can reduce regret more quickly than HLG. Furthermore, there are classes of utility functions for which the bound is tight, so worst-case $\mathcal{U}$ and configuration constraints $\mathcal{C}$ exist that ensure regret will never be reduced to zero in finitely many queries.⁷ This strategy is similar to heuristically motivated polyhedral methods in conjoint analysis used in product design and marketing [18; 11]). In fact, HLG can be viewed as a special case of the method of [18] in which our polyhedra are hyper-rectangles.

3.3 The Current Solution Strategy

While HLG allows one to provide strong worst-case guarantees on regret improvement, it is “undirected” in that considerations of feasibility play no role in determining which queries to ask. An alternative strategy is to focus attention on parameters that participate in defining the max regret, namely, the minimax optimal $\mathbf{x}^*$ and the adversarial witness $\mathbf{x}^w$ for the current $\mathcal{U}$ (recall that the witness maximizes the regret of $\mathbf{x}^*$). The *current solution (CS) query strategy* asks about

⁵We denote the upper and lower bounds of any parameter p by $p^{\uparrow}$ and $p^{\downarrow}$, respectively.

⁶The definition of *maxspan* can be tightened to account for logical consistency of the assignments to different factors, or by restricting attention to feasible states (w.r.t. $\mathcal{C}$). The result still holds.

⁷The bound is not generally tight if there is overlap in factors; but is tight if *maxspan* accounts for logical consistency.

the utility parameter in the set $\{\mathbf{x}^*[k] : k \leq K\} \cup \{\mathbf{x}^w[k] : k \leq K\}$ with largest $\text{gap}(\mathbf{x}[k])$ and queries the midpoint of the corresponding utility interval. Intuitively, should the answer to a query raise the lower bound on some $u(\mathbf{x}^*[k])$ or lower the upper bound on some $u(\mathbf{x}^w[k])$, then the pairwise regret $R(\mathbf{x}^*, \mathbf{x}^w)$ will be reduced, and usually minimax regret as well. Of course, if the answer lowers the upper bound on some $u(\mathbf{x}^*[k])$ or raises the lower bound on some $u(\mathbf{x}^w[k])$, then pairwise regret $R(\mathbf{x}^*, \mathbf{x}^w)$ remains unchanged. (Note that minimax regret might still be reduced.)

We have also experimented with a variant of the CS strategy in which regret is computed approximately to ensure fast interactive response by imposing a time bound on computation (as described above). While we can’t be sure we have the minimax optimal solution with early termination, the solution may be good enough to guide the querying process. Furthermore, since we can compute the max regret of the anytime solution, we know the quality of the solution, and we have an upper bound on minimax regret which can be used as a natural termination criterion for the querying process.

3.4 Alternative Strategies

Finally, we have experimented with several other strategies, which we describe briefly. The *optimistic query strategy* computes an optimistic state $\mathbf{x}^o$ and queries (at the midpoint of the interval) the utility parameter in $\mathbf{x}^o$ with the largest gap. Intuitively, an optimistic $\mathbf{x}^o$ is a useful adversarial choice, so refining information about it can help reduce regret. The *pessimistic query strategy* is analogous, relying on the intuition that a pessimistic choice is useful in preventing the adversary from making us regret our decision too much. The *optimistic-pessimistic (OP) strategy* combines the two: it queries the parameter that has the largest gap among both states. These strategies are computationally appealing since they require only standard CBO, not minimax optimization.

The *most uncertain state (MUS) strategy* is a variant of HLG that accounts for feasibility: we compute a most uncertain state $\mathbf{x}^{mu}$ and query (at the midpoint) the parameter in $\mathbf{x}^{mu}$ with the largest gap. Finally, the *second-best (SB) strategy* is based on the following intuition: suppose we have the optimistic state $\mathbf{x}^o$ and the second-best optimistic state $\mathbf{x}^{2o}$ (i.e., that state with the second-highest upper bound—this is computable with a single optimization). If we could ask a query which reduced the upper bound utility of $\mathbf{x}^o$ to lower than that of $\mathbf{x}^{2o}$, we ensure that regret is reduced (since the adversary can no longer attain this most optimistic value); if the lower bound of $\mathbf{x}^o$ were raised to the level of $\mathbf{x}^{2o}$ ’s upper bound, then we could terminate—knowing that $\mathbf{x}^o$ is *optimal*. Thus we would like to query $\mathbf{x}^o$ at $\mathbf{x}^{2o}$ ’s upper bound: a negative response will reduce regret, a positive response ensures $\mathbf{x}^o$ is optimal. The SB method “distributes” this query across the relevant factor parameters, asking several bound queries.

The *myopically optimal (MY) strategy* computes the average regret reduction of the midpoint query for *each* utility parameter by solving the minimax problem for each response to each query; it then asks the query with the largest average reduction. This approach is generally infeasible, but we test

it on small problems to see how the other methods compare.⁸

4 Empirical Results

We tested the effectiveness of our query strategies on a variety of problems. For each problem we tested: halve largest gap (HLG), current solution (CS), current solution with a computation-time bound of five seconds per query (CS-5), optimistic-pessimistic (OP), second-best (SB), and most uncertain state (MUS). We also compared these against the more computationally demanding myopically optimal method (MY) on small problems. We implemented the constraint generation approach of [5] and used CPLEX as the generic IP solver. Our experiments considered two realistic domains—car rentals and real estate—as well as randomly generated synthetic problems.

First, we tested small synthetic problems to allow a comparison of our heuristics with the MY strategy. Fig. 1 reports the average minimax regret over 45 small synthetic problems constructed by randomly setting the utility bounds and the variables on which each utility factor depends. Each problem has 10 attributes that can take at most 4 values and 10 factors that depend on at most 3 attributes. We simulate user responses by drawing a random utility function u for each trial, consistent with the bounds, representing a user's preferences. Responses to queries are generated using u (assuming the user accurately answers queries relative to u). Results are shown for two cases: utility parameters drawn from a uniform distribution over each interval, and those drawn from a (truncated) Gaussian centered at the midpoint of the interval (reflecting that a user is somewhat more likely to have a true value near the middle of the range).⁹ In both cases, we observe that the OP, CS and CS-5 heuristics decrease minimax-regret at a rate very close to MY. This suggests that OP, CS and CS-5 are computationally feasible, yet promising alternatives to the computationally prohibitive MY strategy.

We report on further experiments using all strategies except MY with larger synthetic problems, a real-estate problem and a car rental problem taken from [5], drawing users from uniform distributions (Gaussian results are very similar both in shape and magnitude). As above, all results are averaged over 45 trials. The car-rental problem is modeled with 26 (multivalued) variables that specify various attributes of a car relevant to typical rental decisions, resulting in 61,917,360,000 possible configurations. The factored utility model consists of 36 local factors, each defined on at most five variables, with 150 utility parameters. Performance of the various query strategies is depicted in Fig. 2(a). Initial utility bounds are set to give minimax regret of roughly 18% of the optimal solution. Both CS and CS-5 perform extremely well: regret is reduced to almost zero within 160 queries on average.¹⁰ More importantly, these methods show

excellent anytime performance: after only 80 queries, average minimax regret has dropped from 18% to under 2%. Interestingly, the time bound of 5 seconds imposed by CS-5, while leading to approximately minimax optimal solutions, does not affect query quality: the approximate solutions give rise to queries that are virtually as effective as the optimal solutions. The CS strategy requires on average at most 83s per query. The OP strategy works very well too, and requires less computation time (0.1s per query) since it does not need to solve minimax problems (except to verify termination “periodically,” which is not reflected in query computation time). However, both OP and CS-5 are fast enough to be used interactively on problems of this size. MUS, HLG, and SB do not work nearly as well. Note the HLG performs poorly since it fails to account for the feasibility of options, thus directing its attention to parts of utility space for which no product exists (hence polyhedral methods alone [18; 11] will not offer reasonable elicitation in our setting). MUS significantly outperforms HLG for just this reason.

The real-estate problem is modeled with 20 (multivalued) variables, with 47,775,744 possible configurations. The factored utility model consists of 29 local factors, giving rise to 100 utility parameters. Query performance is shown in Fig. 2(b), using the same regime as above. Again, both CS and CS-5 perform best, and the time bound of CS-5 has no effect on the quality of the CS strategy. Interestingly, OP performs almost identically to these, with somewhat lower computational cost (CS takes 14s/query, CS-5 5s, and OP 0.1s). Each of these methods reduces minimax regret from 40% of optimal to under 5% in about 120 queries. As above, SB fails to make progress, while HLG and MUS do somewhat better. We also tested the query strategies on larger randomly generated problems (with 25 variables of domain size no more than four, and 20 utility factors with no more than three variables each). The same performance patterns as in real-estate emerge, with CS, CS-5 and OP all performing much better than the others (see Fig. 2(c)). Although OP performs slightly better than CS/CS-5, the difference is not significant.

5 Concluding Remarks

We have developed a number of query strategies for eliciting bounds on the parameters of graphical utility models for the purpose of solving imprecise constraint-based optimization problems. The most promising of these strategies, CS and OP, perform extremely well, requiring very few queries (relative to the model size) to provide dramatic reductions in regret. We have shown that using approximation of minimax regret reduces interactive computation time to levels required for real-time response without a noticeable effect on the performance of CS. OP also can be executed in real-time, since it does not require the same intensive minimax computation.

We plan to extend this research a number of directions. We would like to consider additional query types, such as comparisons of outcomes and tackle the associated computational

⁸By doing lookahead of this type for k stages, we could in fact compute the optimal query plan of k -steps.

⁹All experiments show a reasonably small variance so we exclude error bars for legibility.

¹⁰Though this may seem like a lot of queries, recall that the problem is very large, with a utility model with 150 parameters. We intentionally choose problems this large to push the compu-

tational boundaries of regret-based elicitation. Furthermore, while 160 queries may be large for typical consumer choice problems, it is more than reasonable for high stakes configuration applications.

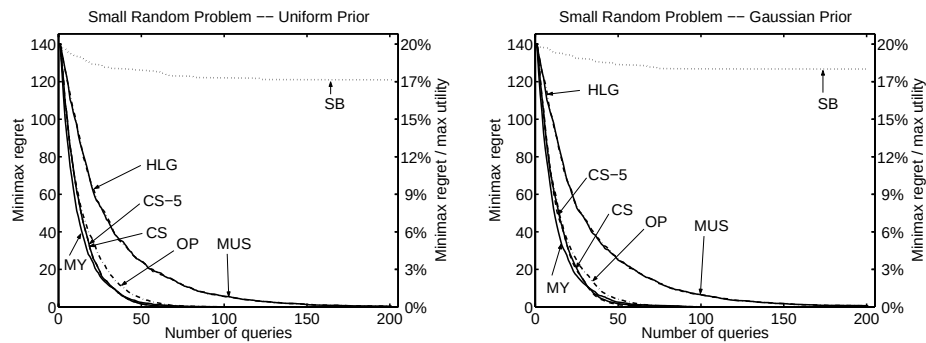


Figure 1: Avg. max regret on small random problems (45 instances) as a function of number of queries given (a) uniform and (b) Gaussian distributed utilities.

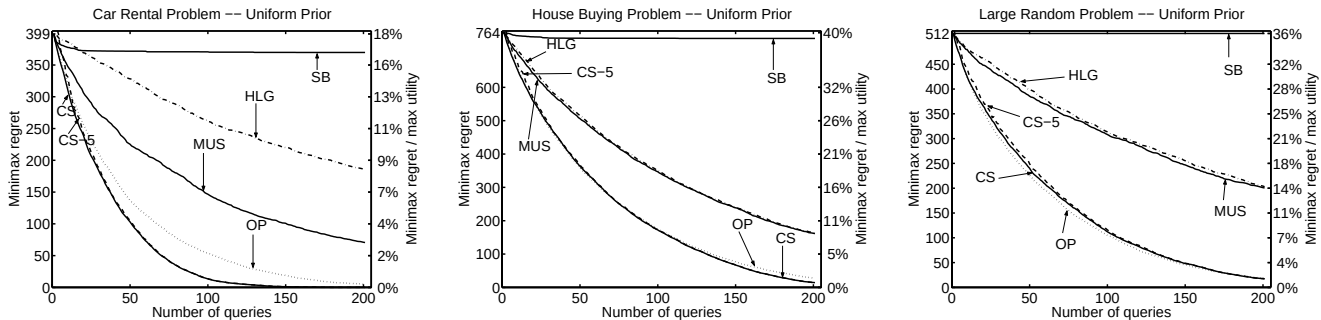


Figure 2: Avg. max regret (45 instances, uniform density) as a function of number of queries: (a) car; (b) real estate; (c) large random problems.

problems. We also plan to explore new query strategies, including those that exploit probabilistic information to compute queries yet still use (distribution-free) regret to make decisions [19]. Non-myopic models are also of considerable interest. Finally, human factors must be addressed, including framing and presentation, and dealing with the fact that users often “construct” preferences during analysis, rather than “reveal” existing preferences—this can often lead to inconsistency. While our current approach will never ask a query for which a response could be inconsistent with prior responses, allowing a user to backtrack still may be important.

Acknowledgements: This research was supported by the the Institute for Robotics and Intelligent Systems (IRIS) and the Natural Sciences and Engineering Research Council (NSERC). Poupart was supported by a scholarship provided by Precarn Incorporated through IRIS.

References

- [1] F. Bacchus and A. Grove. Graphical models for preference and utility. *UAI-95*, pp.3–10, Montreal, 1995.
- [2] S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint satisfaction and optimization. *JACM*, 44:201–236, 1997.
- [3] C. Boutilier. A POMDP formulation of preference elicitation problems. *AAAI-02*, pp.239–246, Edmonton, 2002.
- [4] C. Boutilier, F. Bacchus, and R. I. Brafman. UCP-Networks: A directed graphical representation of conditional utilities. *UAI-01*, pp.56–64, Seattle, 2001.
- [5] C. Boutilier, R. Patrascu, P. Poupart, D. Schuurmans. Constraint-based optimization with the minimax decision criterion. *CP-2003*, pp.168–182, Kinsale, 2003.
- [6] C. Boutilier, T. Sandholm, and R. Shields. Eliciting bid taker non-price preferences in (combinatorial) auctions. *AAAI-04*, pp.204–211, San Jose, CA, 2004.
- [7] U. Chajewska, D. Koller, and R. Parr. Making rational decisions using adaptive utility elicitation. *AAAI-00*, pp.363–369, Austin, TX, 2000.
- [8] P. C. Fishburn. Interdependence and additivity in multivariate, unidimensional expected utility theory. *Intl. Econ. Rev.*, 8:335–342, 1967.
- [9] S. French. *Decision Theory*. Halsted, New York, 1986.
- [10] C. Gonzales and P. Perny. GAI networks for utility elicitation. *KR-04*, pp.224–234, Whistler, BC, 2004.
- [11] V. S. Iyengar, J. Lee, and M. Campbell. Q-Eval: Evaluating multiple attribute items using queries. *ACM EC-01*, pp.144–153, Tampa, FL, 2001.
- [12] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-offs*. Wiley, NY, 1976.
- [13] C. W. Kirkwood, R. K. Sarin. Ranking with partial information: A method and an application. *Op. Res.*, 33:38–48, 1985.
- [14] A. M. Mármol, J. Puerto, F. R. Fernández. The use of partial information on weights in multicriteria decision problems. *J. Multicrit. Dec. Anal.*, 7:322–329, 1998.
- [15] D. Sabin and R. Weigel. Product configuration frameworks—a survey. *IEEE Intel. Sys. and their Appl.*, 13(4):42–49, 1998.
- [16] A. Salo and R. P. Hämäläinen. Preference ratios in multiattribute evaluation (PRIME)—elicitation and decision procedures under incomplete information. *IEEE SMC*, 31(6):533–545, 2001.
- [17] T. Schiex, H. Fargie, and G. Verfaillie. Valued constraint satisfaction problems: Hard and easy problems. *IJCAI-95*, pp.631–637, Montreal, 1995.
- [18] O. Toubia, J. Hauser, and D. Simester. Polyhedral methods for adaptive choice-based conjoint analysis. *TR 4285-03*, Sloan School of Mgmt., MIT, 2003.
- [19] T. Wang and C. Boutilier. Incremental utility elicitation with the minimax regret decision criterion. *IJCAI-03*, pp.309–316, Acapulco, 2003.
- [20] M. Weber. Decision making with incomplete information. *Eur. J. Op. Res.*, 28:44–57, 1987.
- [21] C. C. White, III, A. P. Sage, and S. Dozono. A model of multiattribute decisionmaking and trade-off weight determination under uncertainty. *IEEE SMC*, 14(2):223–229, 1984.

Efficiency and envy-freeness in fair division of indivisible goods: logical representation and complexity

Sylvain Bouveret

IRIT (UPS-CNRS) and

ONERA

31055 Toulouse Cedex 4, France

sylvain.bouveret@cert.fr

Jérôme Lang

IRIT

Université Paul Sabatier - CNRS

31062 Toulouse Cedex, France

lang@irit.fr

Abstract

We study fair division of indivisible goods among agents from the point of view of compact representation and computational complexity. We identify the complexity of several problems, including that of deciding whether there exists an efficient and envy-free allocation when preferences are represented in a succinct way. We also draw connections to nonmonotonic reasoning.

1 Introduction

Allocation of goods among agents has been considered from different perspectives in social choice theory and AI. In this paper we focus on *fair division of indivisible goods without money transfers*. Fair division makes a prominent use of fairness criteria such as *equity* and *envy-freeness*, and on this point totally depart from auctions, where only *efficiency* is relevant (and moreover a specific form of efficiency, since the criterion to be maximized is the total revenue of the auctioneer). Envy-freeness is a key concept in the literature on fair division: an allocation is envy-free if and only if each agent likes her share at least as much as the share of any other agent. Ensuring envy-freeness is considered as crucial; however, envy-freeness alone does not suffice as a criterion for finding satisfactory allocations, therefore it has to be paired with some efficiency criterion, such as Pareto optimality. However, it is known that for any reasonable notion of efficiency, there are profiles for which no efficient and envy-free allocation exists (see [Brams *et al.*, 2000])¹.

Whereas social choice theory has developed an important literature on fair division, computational issues have rarely been considered. On the other hand, artificial intelligence has studied these issues extensively, but until now has focused mainly on combinatorial auctions and related problems, investigating issues such as compact representation as well as complexity and algorithms. Complexity issues for negotiation (where agents exchange goods by means of deals) have also been studied (e.g. [Dunne *et al.*, 2005; Chevaleyre *et al.*, 2004]). See also [Bouveret *et al.*, 2005] for a preliminary complexity study of fair division problems.

¹This is even trivial if every good must be assigned to someone: in this case, there are profiles for which not even an envy-free allocation exists.

The above discussion reveals the existence of a gap: compact representation and complexity issues for fair division have received very little attention until now, apart of the recent work [Lipton *et al.*, 2004] which studies approximation schemes for envy-freeness. The need for compact representation arises from the following dilemma, formulated by several social choice theorists: either (a) allow agents to express any possible preference relation on the set of all subsets of items, and end up with an exponentially large representation (such as in [Herreiner and Puppe, 2002]); or (b) severely restrict the set of expressible preferences, typically by assuming additive independence between items, and then design procedures where agents express preferences between single items, thus giving up the possibility of expressing preferential dependencies such as complementarity and substitutability effects among items; this is the path followed by [Brams *et al.*, 2000] and [Demko and Hill, 1998]. Yet, as we advocate in this paper, conciliating conciseness and expressivity is possible, by means of *compact representation*.

As in most works on fair allocation of indivisible items we focus on the joint search for envy-freeness and efficiency. The impossibility to guarantee the existence of an efficient envy-free allocation implies that determining whether there exists such an allocation is a crucial task, since a positive answer leads to choose such an allocation whereas a negative answer calls for a relaxation of one of the criteria. We consider this problem from the point of view of compact representation and computational complexity. We focus first in the simple case where agents have *dichotomous preferences*, that is, they simply express a partition between satisfactory and unsatisfactory shares. The interest of such a restriction is that in spite of the expressivity loss it imposes, it will be shown to be no less complex than the general case, while being much simpler to expose. The most natural representation of a dichotomous preference is by a single propositional formula, where variables correspond to goods. Expressing envy-freeness and efficiency within this logical representation reveals unexpected connections to nonmonotonic reasoning. We identify the complexity of the key problem of the existence of an envy-free and Pareto-efficient allocation, which turns out to be Σ_2^P -complete; we also identify the complexity of several other problems obtained either by imposing some restrictions on the latter or by replacing Pareto-efficiency by other criteria. In Section 5 we extend this result to the case of non-dichotomous, compactly represented preferences.

2 Background

2.1 Fair division problems

Definition 1

A **fair division problem** is a tuple $\mathcal{P} = \langle I, X, \mathcal{R} \rangle$ where

- $I = \{1, \dots, N\}$ is a set of agents;
- $X = \{x_1, \dots, x_p\}$ is a set of indivisible goods;
- $\mathcal{R} = \langle R_1, \dots, R_N \rangle$ is a preference profile, where each R_i is a reflexive, transitive and complete relation on 2^X .

R_i is the preference relation of agent i . AR_iB is alternatively denoted by $R_i(A, B)$ or by $A \succeq_i B$; we note $A \succ_i B$ (strict preference) for $A \succeq_i B$ and not $B \succeq_i A$ and $A \sim_i B$ (indifference) for $A \succeq_i B$ and $B \succeq_i A$.

In addition, R_i is said to be *monotonous* if and only if for all $A, B, A \subseteq B \subseteq X$ implies $B \succeq_i A$. $\mathcal{R} = \langle R_1, \dots, R_N \rangle$ is *monotonous* if and only if R_i is *monotonous* for every i .

Definition 2

- An **allocation** for $\mathcal{P} = \langle I, X, \mathcal{R} \rangle$ is a mapping $\pi : I \rightarrow 2^X$ such that for all i and $j \neq i$, $\pi(i) \cap \pi(j) = \emptyset$. If for every $x \in X$ there exists a i such that $x \in \pi(i)$ then π is a complete allocation.
- Let π, π' two allocations. π dominates π' if and only if (a) for all i , $\pi(i) \succeq_i \pi'(i)$ and (b) there exists an i such that $\pi(i) \succ_i \pi'(i)$. π is **(Pareto-) efficient** if and only if there is no π' such that π' dominates π .
- An allocation π is **envy-free** if and only if $\pi(i) \succeq_i \pi(j)$ holds for all i and all $j \neq i$.

2.2 Propositional logic

Let V be a finite set of *propositional variables*. L_V is the propositional language generated from V , the usual connectives $\neg, \wedge$ and $\vee$ and the Boolean constants $\top$ and $\perp$ in the usual way². An *interpretation* M for L_V is an element of 2^V , i.e., a truth assignment to symbols: for all $x \in V$, $x \in M$ (resp. $x \notin M$) means that M assigns x to true (resp. to false). $\text{Mod}(\varphi) = \{M \in 2^V \mid M \models \varphi\}$ is the set of all models of φ (the satisfaction relation $\models$ is defined as usual, as well as satisfiability and logical consequence).

A *literal* is a formula of L_V of the form x or of the form $\neg x$, where $x \in V$. A formula φ is under *negative normal form* (or NNF) if and only if any negation symbol in φ appears only in literals. Any formula can be turned in polynomial time into an equivalent NNF formula. For instance, $a \wedge \neg(b \wedge c)$ is not under NNF but is equivalent to the NNF formula $a \wedge (\neg b \vee \neg c)$.

A formula is *positive* if it contains no occurrence of the negation symbol. For instance, $a \wedge (b \vee \neg c)$ and $a \vee (\neg a \wedge b)$ are not positive, whereas $a \wedge (b \vee c)$ and $(a \wedge c) \vee (a \wedge b)$ are. $\top$ and $\perp$ are considered positive as well.

Let $\varphi \in L_V$. $\text{Var}(\varphi) \subseteq V$ is the set of propositional variables appearing in φ . For instance, $\text{Var}((a \wedge c) \vee (a \wedge b)) = \{a, b, c\}$ and $\text{Var}(\top) = \emptyset$.

Lastly, if $S = \{\varphi_1, \dots, \varphi_n\}$ is a finite set of formulas then $\bigwedge S = \varphi_1 \wedge \dots \wedge \varphi_n$ is the conjunction of all formulas of S .

²Note that connectives $\rightarrow$ and $\leftrightarrow$ are not allowed; this is important for the definition of positive formulas (to come).

2.3 Computational complexity

In this paper we will refer to some complexity classes located in the polynomial hierarchy. We assume the reader to be familiar with the classes NP and coNP. BH_2 (also referred to as DP) is the class of all languages of the form $L_1 \cap L_2$ where L_1 is in NP and L_2 in coNP. $\Delta_2^P = \text{P}^{\text{NP}}$ is the class of all languages recognizable by a deterministic Turing machine working in polynomial time using NP oracles. Likewise, $\Sigma_2^P = \text{NP}^{\text{NP}}$. $\Theta_2^P = \Delta_2^P[\mathcal{O}(\log n)]$ is the subclass of Δ_2^P of problems that only need a logarithmic number of oracles. See for instance [Papadimitriou, 1994] for further details.

3 Fair division problems with dichotomous preferences: logical representation

We start by considering in full detail the case where preferences are dichotomous.

Definition 3 R_i is *dichotomous* if and only if there exists a subset Good_i of 2^X such that for all $A, B \subseteq X$, $A \succeq_i B$ if and only if $A \in \text{Good}_i$ or $B \notin \text{Good}_i$. $\mathcal{R} = \langle R_1, \dots, R_N \rangle$ is *dichotomous* if and only if every R_i is *dichotomous*.

There is an obvious way of representing dichotomous preferences compactly, namely by a propositional formula φ_i (for each agent i) of the language L_X (a propositional symbols for each good) such that $\text{Mod}(\varphi) = \text{Good}_i$. Formally:

Definition 4 Let R_i be a dichotomous preference on 2^X , with Good_i its associated subset of 2^X , and φ_i a propositional formula on the propositional language L_X . We say that φ_i represents R_i if and only if $\text{Mod}(\varphi_i) = \text{Good}_i$.

Clearly, for any dichotomous preference R_i there is a formula φ_i representing R_i ; furthermore, this formula is unique up to logical equivalence.

Example 1 $X = \{a, b, c\}$ and $\text{Good}_i = \{\{a, b\}, \{b, c\}\}$. Note that R_i is not *monotonous*. Then $\varphi_i = (a \wedge b \wedge \neg c) \vee (\neg a \wedge b \wedge c)$ represents R_i .

An easy but yet useful result (whose proof is omitted):

Proposition 1 Let R_i be a dichotomous preference on 2^X . The following statements are equivalent:

1. R_i is *monotonous*;
2. Good_i is *upward closed*, that is, $A \in \text{Good}_i$ and $B \supseteq A$ imply $B \in \text{Good}_i$.
3. R_i is *representable by a positive propositional formula*.

From now on, we assume that allocation problems $\mathcal{P}$ are represented in propositional form, namely, instead of I, X and $\mathcal{R}$ we only specify $\langle \varphi_1, \dots, \varphi_N \rangle$. I and X are obviously determined from $\langle \varphi_1, \dots, \varphi_N \rangle$.

Let $\mathcal{P} = \langle \varphi_1, \dots, \varphi_N \rangle$ be an allocation problem with dichotomous preferences; then for each $i \leq N$, we rewrite φ_i into φ_i^* obtained from φ_i by replacing every variable x^* by the new symbol x_i . For instance, if $\varphi_1 = a \wedge (b \vee c)$ and $\varphi_2 = a \wedge d$ then $\varphi_1^* = a_1 \wedge (b_1 \vee c_1)$ and $\varphi_2^* = a_2 \wedge d_2$.

For all $i \leq N$, let $X_i = \{x_i, x \in X\}$. An allocation for a standard allocation problem $\mathcal{P}$ corresponds to a model of $V = X_1 \cup \dots \cup X_N$ satisfying at most one x_i for each $x \in X$. In other terms, there is a bijective mapping between the set of

possible allocations and the models of the following formula

$$\Gamma_P = \bigwedge_{x \in X} \bigwedge_{i \neq j} \neg(x_i \wedge x_j)$$

If allocation are required to be complete, then Γ_P is replaced by $\Gamma_P^C = \Gamma_P \wedge \bigwedge_{x \in X} (x_1 \vee \dots \vee x_n)$. The rest is unchanged.

Let $V = \{x_i \mid i = 1, \dots, N, x \in X\}$. Any interpretation M of $\text{Mod}(\Gamma_P)$ is such that it is never the case that x_i and x_j are simultaneously true for $i \neq j$, therefore we can map $M \in \text{Mod}(\Gamma_P)$ to an allocation $F(\pi) = M$ simply defined by $\pi(i) = \{x \mid M \models x_i\}$. This mapping is obviously bijective, and we denote $F^{-1}(M)$ the allocation corresponding to an interpretation M of $\text{Mod}(\Gamma_P)$.

3.1 Envy-freeness

We now show how the search for an envy-free allocation can be mapped to a satisfiability problem. Let $\varphi_{j|i}$ be the formula obtained from φ_i^* by substituting every symbol x_i in φ_i^* by x_j : for instance, if $\varphi_1^* = a_1 \wedge (b_1 \vee c_1)$ then $\varphi_{2|1}^* = a_2 \wedge (b_2 \vee c_2)$. (Obviously, $\varphi_{i|i}^* = \varphi_i^*$.)

Proposition 2 Let $\mathcal{P} = \langle \varphi_1, \dots, \varphi_N \rangle$ be an allocation problem with dichotomous preferences under propositional form, and the formulas $\varphi_{j|i}^*$ and mapping F as defined above. Let

$$\Lambda_P = \bigwedge_{i=1, \dots, N} \left[\varphi_i^* \vee \left(\bigwedge_{j \neq i} \neg \varphi_{j|i}^* \right) \right]$$

Then π is envy-free if and only if $F(\pi) \models \Lambda_P$.

The proof is simple, so we omit it. The search for envy-free allocations can thus be reduced to a satisfiability problem: $\{F^{-1}(M) \mid M \models \Gamma_P \wedge \Lambda_P\}$ is the set of envy-free allocations for $\mathcal{P}$. Note that, importantly, $\Gamma_P \wedge \Lambda_P$ has a polynomial size (precisely, quadratic) in the size of the input data.

Example 2

$\varphi_1 = a \vee (b \wedge c)$; $\varphi_2 = a$; $\varphi_3 = a \vee b$.
 $\Lambda_P = ((a_1 \vee (b_1 \wedge c_1)) \vee (\neg(a_2 \vee (b_2 \wedge c_2)) \wedge \neg(a_3 \vee (b_3 \wedge c_3))))$
 $\wedge (a_2 \vee (\neg a_1 \wedge \neg a_3)) \wedge ((a_3 \vee b_3) \vee (\neg(a_1 \vee b_1) \wedge \neg(a_2 \vee b_2)))$;
 $\text{Mod}(\Gamma_P \wedge \Lambda_P) = \{\{c_1\}, \{c_1, b_3\}, \{c_2, b_3\}, \{c_3\}, \{b_3\}, \{c_3\}, \emptyset\}$.
 There are therefore 7 envy-free allocations, namely $(c, -, -)$, $(c, -, b)$, $(-, c, b)$, $(-, c, -)$, $(-, -, b)$, $(-, -, c)$ and $(-, -, -)$. Note that none of them is complete.

3.2 Efficient allocations

Definition 5 Let $\Delta = \{\alpha_1, \dots, \alpha_m\}$ a set of formulae and β a formula. $S \subseteq \Delta$ is a maximal β -consistent subset of Δ iff (a) $S \wedge \beta$ is consistent and (b) there is no S' such that $S \subset S' \subseteq \Delta$ and $S' \wedge \beta$ is consistent. Let $\text{MaxCons}(\Delta, \beta)$ be the set of all maximal β -consistent subsets of Δ .

Proposition 3 Let $\mathcal{P} = \langle \varphi_1, \dots, \varphi_N \rangle$ an allocation problem. Let $\Phi_P = \{\varphi_1^*, \dots, \varphi_N^*\}$. Then π is efficient for $\mathcal{P}$ if and only if $\{\varphi_i^* \mid F(\pi) \models \varphi_i^*\}$ is a maximal Γ_P -consistent subset of Φ_P .

This simple result, whose proof is omitted, suggests that efficient allocations can be computed from the logical expression Φ of the problem, namely, by computing the maximal Γ_P -consistent subsets of Φ ; call them $\{S_1, \dots, S_q\}$. Then for each S_i , let $M_i = \text{Mod}(\bigwedge S_i \wedge \Gamma_P)$ and let $M = \bigcup_{i=1}^q M_i$. Then $F^{-1}(M)$ is the set of all efficient allocations for $\mathcal{P}$.

³Note that there are in general exponentially many maximal Γ_P -consistent subsets of Φ (and therefore exponentially many efficient

Example 2 (cont'd) The maximal Γ_P -consistent subsets of Φ are $S_1 = \{\varphi_1^*, \varphi_2^*\}$, $S_2 = \{\varphi_1^*, \varphi_3^*\}$ and $S_3 = \{\varphi_2^*, \varphi_3^*\}$. $\bigwedge S_1 \wedge \Gamma_P$ has only one model: $\{b_1, c_1, a_2\}$. $\bigwedge S_2 \wedge \Gamma_P$ has two models: $\{a_1, b_3\}$ and $\{b_1, c_1, a_3\}$. $\bigwedge S_3 \wedge \Gamma_P$ has one model: $\{a_2, b_3\}$. Therefore the four efficient allocations for $\mathcal{P}$ are $(bc, a, -)$, $(a, -, b)$, $(bc, -, a)$ and $(-, a, b)$. None of them is envy-free.

3.3 Efficient and envy-free allocations

We are now in position of putting things together. Since envy-free allocations corresponds to the models of Λ_P and efficient allocations to the models of maximal Γ_P -consistent subsets of Φ_P , the existence of an efficient and envy-free (EEF) allocation is equivalent to the following condition: *there exists a maximal Γ_P -consistent subset S of Φ_P such that $\bigwedge S \wedge \Gamma_P \wedge \Lambda_P$ is consistent*. In this case, the models of the latter formula are the EEF allocations. Interestingly, this is an instance of a well-known problem in nonmonotonic reasoning:

Definition 6 A supnormal default theory⁴ is a pair $D = \langle \beta, \Delta \rangle$ with $\Delta = \{\alpha_1, \dots, \alpha_m\}$, where $\alpha_1, \dots, \alpha_m$ and β are propositional formulas. A propositional formula ψ is a skeptical consequence of D , denoted by $D \vdash^s \psi$, if and only if for all $S \in \text{MaxCons}(\Delta, \beta)$ we have $\bigwedge S \wedge \beta \models \psi$.

Proposition 4 Let $\mathcal{P} = \langle \varphi_1, \dots, \varphi_N \rangle$ a fair division problem. Let $D_P = \langle \Gamma_P, \Phi_P \rangle$. Then there exists an efficient and envy-free allocation for $\mathcal{P}$ if and only if $D \not\vdash^s \neg \Lambda_P$.

This somewhat unexpected connection to nonmonotonic reasoning has several implications. First, EEF allocations correspond to the models of $\bigwedge S \wedge \Gamma_P \wedge \Lambda_P$ for $S \in \text{MaxCons}(\Phi_P, \Gamma_P)$; however, $\text{MaxCons}(\Phi_P, \Gamma_P)$ may be exponentially large, which argues for avoiding to start computing efficient allocations and then filtering out those that are not envy-free, but rather compute EEF allocations in a single step, using default reasoning algorithms thus, fair division may benefit from computational work in default logic and connex domains such as belief revision and answer set programming. Moreover, alternative criteria for selecting extensions in default reasoning (such as cardinality, weights or priorities) correspond to alternative efficiency criteria in allocation problems.

4 Allocation problems with dichotomous preferences: complexity

It is known that skeptical inference is Π_2^P -complete [Gottlob, 1992]; now, after Proposition 4, the problem of the existence of an EEF allocation can be reduced to the complement of a skeptical inference problem, which immediately tells that it

allocations). This can be tempered by (a) there are many practical cases where the number of maximal consistent subsets is small; (b) it is generally not asked to look for *all* efficient allocations; if we look for just one, then this can be done by computing one maximal Γ_P -consistent subset of Φ .

⁴“Supnormal” defaults are also called “normal defaults without prerequisites” (e.g., [Reiter, 1980]).

is in Σ_2^p . Less obviously, we now show that it is complete for this class, even if preferences are required to be monotonous.

Let us first note that skeptical inference remains Π_2^p -complete under these two restrictions (to which we refer as RSI, for RESTRICTED SKEPTICAL INFERENCE): (a) $\psi = \varphi_1$; (b) $n \geq 2$. (Here is the justification: $\Delta \sim \psi$ if and only if $\Delta \cup \{\psi, \psi\} \sim \psi$.) Equivalently, RSI is the problem of deciding whether, given $\Delta = \langle \alpha_1, \dots, \alpha_n \rangle$ with $n \geq 2$, all maximal consistent subsets of Δ contain α_1 .

Proposition 5 *The problem EEF EXISTENCE of determining whether there exists an efficient and envy-free allocation for a given problem $\mathcal{P}$ with monotonous, dichotomous preferences under logical form is Σ_2^p -complete.*

We show hardness by the following reduction from $\overline{\text{RSI}}$ (the complement problem of RSI) to EEF EXISTENCE. Given any finite set Δ of propositional formulae, let $V_\Delta = \text{Var}(\Delta)$ the set of propositional symbols appearing in Δ , and let $\mathcal{P} = H(\Delta)$ the following instance of EEF EXISTENCE:

- (1) $I = \{1, 2, \dots, n+2\}$;
- (2) $X = \{v^i | v \in V_\Delta, i \in 1 \dots n\} \cup \{\bar{v}^i | v \in V_\Delta, i \in 1 \dots n\} \cup \{x^1 | i \in 1 \dots n\} \cup \{y\}$;
- (3) for each $i = 1, \dots, n$, let β_i be obtained from α_i by the following sequence of operations: (i) put α_i into NNF form (let α'_i be the result); (b) for every $v \in V_\Delta$, replace, in α'_i , each (positive) occurrence of v by v^i and each occurrence of $\neg v$ by $\bar{v}^i$; let β_i be the formula obtained. Then

- $\varphi_1 = \beta_1 \vee x^1$;
- for $i = 2, \dots, n$, $\varphi_i = \beta_i \wedge x^i$;
- $\varphi_{n+1} = \left(\left(\bigwedge_{v \in \text{Var}(\Delta)} \left(\bigwedge_{i=1}^n v^i \right) \vee \left(\bigwedge_{i=1}^n \bar{v}^i \right) \right) \wedge x^1 \right) \vee y$;
- $\varphi_{n+2} = y$.

Lemma 1 *An allocation π for $\mathcal{P}$ is said to be regular if and only if for all $i \neq n$, $\pi(i) \subseteq \sigma(i)$, where*

- for all $i \neq n$, $\sigma(i) = \bigcup_{v \in V_\Delta} \{v^i, \bar{v}^i\} \cup \{x^i\}$;
- $\sigma(n+1) = \bigcup_{v \in V_\Delta, i=1, \dots, n} \{v^i, \bar{v}^i\} \cup \{x^1, y\}$;
- $\sigma(n+2) = \{y\}$.

Let now π_R defined by $\pi_R(i) = \pi(i) \cap \sigma_i$. Then

1. π_R is regular;
2. π is efficient if and only if π_R is efficient;
3. if π is envy-free then π_R is envy-free.

Proof: (1) is obvious. For all i , the goods outside $\sigma(i)$ do not have any influence on the satisfaction of i (since they do not appear in α_i), therefore $\pi_R(i) \sim_i \pi(i)$, from which (2) follows. The formulas α_i being positive, the preference relations $\succeq_i$ are monotonous, therefore $\pi(j) \succeq_i \pi_R(j)$ holds for all i, j . Now, if π is envy-free then for all i, j we have $\pi(i) \succeq_i \pi(j)$, therefore $\pi_R(i) \sim_i \pi(i) \succeq_i \pi(j) \succeq_i \pi_R(j)$ and π_R is envy-free, from which (3) follows. ■

Lemma 2 *If π is regular then*

1. 1 can only envy $n+1$;

2. $2, \dots, n$ envy noone;
3. $n+1$ can only envy $n+2$;
4. $n+2$ can only envy $n+1$;

Proof: First, note that for any $i, j \neq i$, i envies j if and only if $\pi(i) \models \neg \varphi_i$ and $\pi(j) \models \varphi_i$.

1. Let $i = 1$ and $j \in \{2, \dots, n, n+2\}$. Assume 1 envies j . Then $\pi(j) \models \varphi_1$. π being regular, $x^1 \notin \pi(j)$, therefore $\pi(j) \models \beta_1$. Now, since π is regular, $\pi(j)$ does not contain any v^i nor any $\bar{v}^i$; now, β_1 can only be made true by variables v^i or $\bar{v}^i$ (which cannot be the case here) unless it is a tautology. Now, if β_1 is a tautology, then 1 is satisfied by π and cannot envy j , a contradiction.
2. Let $i \in \{2, \dots, n\}$ and $j \neq i$. If i envies j then $\pi(j) \models \beta_i \wedge x^i$, which is impossible because $x^i \notin \pi(j)$, due to the regularity of π .
3. Let $i = n+1$. Assume $n+1$ envies 1 then $\pi(1) \models \varphi_{n+1}$. Since $\pi(1) \models y$ is impossible (because π is regular), we have $\pi(1) \models \bigwedge_{v \in V_\Delta} \left(\bigwedge_{i=1}^n v^i \right) \vee \left(\bigwedge_{i=1}^n \bar{v}^i \right) \wedge x^1$, which implies that either $\pi(1) \models \bigwedge_{v \in V_\Delta} \left(\bigwedge_{i=1}^n v^i \right)$ or $\pi(1) \models \left(\bigwedge_{i=1}^n \bar{v}^i \right)$. Both are impossible because π is regular and $n \geq 2$. The case $j \in \{2, \dots, n\}$ is similar.
4. Let $i = n+2$ and $j \neq n$. If i envies j then $\pi(j) \models y$, which is impossible because π is regular. ■

Lemma 3 *Let π be a regular allocation satisfying $n+1$ and $n+2$. Let $M(\pi)$ be the interpretation on V_Δ obtained from π by: for all $v \in V_\Delta$, $M(\pi) \models v$ (i.e., $v \in M(\pi)$) if $n+1$ receives $\bar{v}^1, \dots, \bar{v}^n$, and $M(\pi) \models \neg v$ otherwise, i.e., if $n+1$ receives $v^1, \dots, v^n$. Then π is envy-free iff $M(\pi) \models \alpha_1$.*

Proof: Let π be a regular allocation satisfying $n+1$ and $n+2$. Since π satisfies $n+2$, $y \in \pi(n+2)$. Now, π satisfies $n+1$ without giving him y , therefore, for any v , $n+1$ receives either all the v^i 's or all the $\bar{v}^i$'s. This shows that our definition of $M(\pi)$ is well-founded. Now, since π is regular, it is envy-free if and only if (a) 1 does not envy $n+1$, (b) $n+1$ does not envy $n+2$ and (c) $n+2$ does not envy $n+1$. Since $n+1$ and $n+2$ are satisfied by π , we get that π is envy-free if and only if 1 does not envy $n+1$, that is, if and only if either $\pi(1) \models \varphi_1$ or $\pi(n+1) \not\models \varphi_1$. Now, $\pi(n+1)$ contains x^1 , therefore $\pi(n+1) \models \varphi_1$, which entails that π is envy-free if and only if $\pi(1) \models \varphi_1$. This is equivalent to $\pi(1) \models \beta_1$, because 1 does not get x^1 (which is assigned to $n+1$), which in turn is equivalent to $M(\pi) \models \alpha_1$ by construction. ■

Lemma 4 *For each interpretation M over V_Δ , let us define $\pi_M : I \rightarrow 2^X$ by:*

- $\pi_M(1) = \{v^1 | M \models v\} \cup \{\bar{v}^1 | M \models \neg v\}$;
- for each $i \in 2, \dots, n$, $\pi_M(i) = \{v^i | M \models v\} \cup \{\bar{v}^i | M \models \neg v\} \cup \{x^i\}$;
- $\pi_M(n+1) = \{x^1\} \cup \{\bar{v}^i | M \models v, i = 1, \dots, n\} \cup \{v^i | M \models \neg v, i = 1, \dots, n\}$;
- $\pi_M(n+2) = \{y\}$

Then:

1. π_M is a well-defined and regular allocation satisfying $n+1$ and $n+2$;
2. $M_{\pi_M} = M$ (M_{π_M} is obtained from π_M as in Lemma3).
3. for any $i \in 1, \dots, n$, π_M satisfies i iff $M \models \alpha_i$.

4. π_M is efficient iff M satisfies a maximal consistent subset of Δ .

Proof:

1. π_M does not give the same good to more than one individual, therefore it is an allocation. The rest is straightforward.
2. if $M \models v$ then $\pi_M(n+1)$ contains $\{\bar{v}^i \mid i = 1, \dots, n\}$ and therefore $M(\pi_M) \models v$. The case $M \models \neg v$ is similar.
3. let $i \in 2, \dots, n$. Since π_M gives x^i to i , π_M satisfies i if and only if $\pi_M(i) \models \beta^i$, which is equivalent to $M \models \alpha_i$. If $i = 1$ then, since π_M does not give x^1 to 1, π_M satisfies 1 if and only if $\pi_M(1) \models \beta^1$, which is equivalent to $M \models \alpha_1$.
4. from point 3, $\{i, \pi_M \text{ satisfies } i\} = \{i, M \models \alpha_i\}$. Now, since preferences are dichotomous, an allocation π is efficient if and only if the set of individuals it satisfies is maximal with respect to inclusion. Therefore, π_M is efficient if and only if M satisfies a maximal consistent subset of Δ . ■

Lemma 5 *Let π be a regular and efficient allocation satisfying $n+1$ and $n+2$. Then $M(\pi)$ satisfies a maximal consistent subset of Δ .*

Proof: π is regular and satisfies $n+1$ and $n+2$, therefore $\pi(n+2) = \{y\}$, $x^1 \in \pi(n+1)$, and $M(\pi)$ is well-defined. Let π' obtained from π by

- $\pi'(1) = \{v^1 \mid M(\pi) \models v\} \cup \{\bar{v}^1 \mid M(\pi) \models \neg v\}$;
- for each $i = 2, \dots, n$: $\pi'(i) = \{v^i \mid M(\pi) \models v\} \cup \{\bar{v}^i \mid M(\pi) \models \neg v\} \cup \{x^i\}$;
- $\pi'(n+1) = \{x^1\} \cup \{v^i \mid M(\pi) \models \neg v\} \cup \{\bar{v}^i \mid M(\pi) \models v\}$;
- $\pi'(n+2) = \{y\}$.

π being regular and satisfying $n+1$ and $n+2$, we have $\pi(n+1) = \pi'(n+1)$, $\pi(n+2) = \pi'(n+2)$, and then for each i , $\pi(i) \subseteq \pi'(i)$: indeed, let $j \in \{2, \dots, n\}$ (for $n+1$ and $n+2$ this inclusion is obviously satisfied); then (a) π is regular, therefore $\pi(j) \subseteq \sigma(j)$; now, all goods of $\sigma(j)$ are either in $\pi'(j)$ or in $\pi(n+1)$ (namely: x^1 if $j = 1$ and all the v^j such that $M(\pi) \models \neg v$ and all the $\bar{v}^j$ such that $M(\pi) \models v$); therefore, $\pi(j) \subseteq \pi'(j) \cup \pi(n+1)$, which, together with $\pi(1) \cap \pi(n+1) = \emptyset$, implies $\pi(j) \subseteq \pi'(j)$. Since preferences are monotonous, all individuals satisfied by π are satisfied by π' as well; and since π is efficient, π and π' satisfy the same set of individuals. Now, we remark that $\pi' = \pi_{M(\pi)}$. By Lemma 4, π' is efficient iff $M(\pi)$ satisfies a maximal consistent subset of Δ , from which we conclude. ■

Lemma 6 *Any envy-free and efficient allocation for $\mathcal{P}$ satisfies $n+1$ and $n+2$.*

Proof: Suppose π doesn't satisfy $n+1$; then $y \notin \pi(n+1)$; now, if $y \in \pi(n+2)$ then $n+1$ envies $n+2$; if $y \notin \pi(n+2)$ then π is not efficient because giving y to $n+2$ would satisfy $n+2$ and thus lead to a better allocation than π .

Now, suppose π does not satisfy $n+1$, i.e., $y \notin \pi(n+2)$; if $y \in \pi(n+1)$ then $n+2$ envies $n+1$; if $y \notin \pi(n+1)$ then again, π is not efficient because giving y to $n+2$ would satisfy $n+2$ and thus lead to a better allocation than π . ■

Lemma 7 *If there exists an EEF allocation, then there exists a maximal consistent subset of Δ containing α_1 .*

Proof: Let π be an efficient and envy-free allocation. By Lemma 1, π_R is regular, efficient and envy-free. By Lemma 6, π_R satisfies $n+1$ and $n+2$. Then by Lemma 5, $M(\pi_R)$ satisfies a maximal consistent subset of Δ , and by Lemma 3, $M(\pi_R) \models \alpha_1$. Therefore $Sat(M(\pi_R), \Delta)$ is a maximal consistent subset of Δ and contains α_1 . ■

Lemma 8 *If there exists a maximal consistent subset of Δ containing α_1 then there exists an EEF allocation.*

Proof: Assume that there exists a maximal consistent subset S of Δ containing α_1 , and let M be a model of S . By point 4 of Lemma 4, π_M is efficient.

By point 1 of Lemma 4, π_M is regular; then by Lemma 2, π_M is envy-free if and only if (i) 1 does not envy $n+1$, (ii) $n+1$ does not envy $n+2$ and (iii) $n+2$ does not envy $n+1$. By point 1 of Lemma 4, π_M satisfies $n+1$ and $n+2$, therefore (ii) and (iii) hold. Lastly, by point 5 of Lemma 4, $M \models \alpha_1$ implies that π_M satisfies 1, therefore (i) holds as well and π_M is envy-free. ■

Proof of Proposition 5: from Lemmas 7 and 8, the existence of a maximal consistent subset of Δ containing α_1 and the existence of an efficient and envy-free allocation for $\mathcal{P} = H(\Delta)$ are equivalent. Clearly, H is computed in polynomial time. Therefore, H is a polynomial reduction from RSI to EEF EXISTENCE, which shows that the latter problem is Σ_2^P -hard, and therefore Σ_2^P -complete. ■

As a corollary, this Σ_2^P -completeness result holds for general (not necessarily monotonous) dichotomous preferences.

As a consequence of this high complexity, it is worth studying restrictions and variants of the latter problem for which complexity may fall down. We start by considering *identical* dichotomous preference profiles, that is, all agents have the same preference, i.e. the same formula φ .

Proposition 6 *EEF EXISTENCE with N identical dichotomous, monotonous preferences is NP-complete, for any fixed $N \geq 2$.*

Due to space limitations the proofs of this result and the following ones are omitted⁵. Note that we have here a hardness result for any *fixed* number of agents (≥ 2). Things are different with Proposition 5, for which hardness does not hold when N is fixed. Namely, the following holds for $N = 2$:

Proposition 7 *EEF EXISTENCE for two agents with monotonous dichotomous preferences is NP-complete.*

Unlike Proposition 5, these results are sensitive to whether preferences are required to be monotonous or not.

Proposition 8 *EEF EXISTENCE with N identical dichotomous preferences is COBH₂-complete, for any fixed $N \geq 2$.*

Proposition 9 *EEF EXISTENCE for 2 agents with dichotomous preferences is COBH₂-complete.*

Complexity decreases as well if we weaken Pareto-efficiency by only requiring allocations to be *complete*:

⁵They can be found in the long version of the paper, accessible at <http://www.irit.fr/recherches/RPDM/persos/JeromeLang/papers/eeef.pdf>.

Proposition 10 *The existence of a complete envy-free allocation for agents with monotonous, dichotomous preferences is NP-complete, even for 2 agents with identical preferences.*

Lastly, replacing Pareto-efficiency by an utilitarianistic notion of efficiency results in a complexity decrease as well:

Proposition 11 *The existence of an envy-free allocation satisfying a maximal number of agents with monotonous dichotomous preferences is Θ_2^p -complete.*

5 Non-dichotomous preferences

We now consider the case where preferences are no longer dichotomous. Again, since an explicit description of preferences is exponentially large, the need for a compact description thereof is clear. Many languages exist for succinct representation of preference. However, Proposition 5 extends to any language, provided that (a) it extends propositional logic, i.e., it is able to express compactly any dichotomous preference represented by a propositional formula; (b) comparing two sets of goods can be done in polynomial time. Conditions (a) and (b) are met by many languages for succinct representation of preference⁶. Under assumptions (a) and (b):

Corollary 1 *EEF EXISTENCE with monotonous preference under logical form is Σ_2^p -complete.*

For the latter result preferences do not have to be numerical since Pareto efficiency and envy-freeness are purely ordinal notions. Now, if preferences are numerical, which implies the possibility of intercomparing and aggregating preferences of several agents, then, besides Pareto-efficiency, we may consider efficiency based on social welfare functions. We consider here only the two most classical way of aggregating a collection of utility functions $\langle u_1, \dots, u_n \rangle$ into a social welfare function sw : utilitarianism ($sw = \sum_i u_i$) and egalitarianism ($sw = \min_i u_i$).

Proposition 12 *Given a collection of utility functions on 2^R given in compact form:*

- *the problem of the existence of an envy-free allocation maximizing utilitarian social welfare is Δ_2^p -complete, even if $N = 2$.*
- *the problem of the existence of an envy-free allocation maximizing egalitarian social welfare is Δ_2^p -complete, even if $N = 2$.*

A last case that has not been considered is the case of *additive* numerical preferences. In the latter case, the utility function of agent $i \leq N$ is simply expressed by the p numbers $u_i(\{x_j\})$, $j = 1, \dots, p$. While the existence of a *complete envy-free* allocation is easily shown to be NP-complete (see [Lipton et al., 2004]), things become much harder with EEF EXISTENCE: all we know is that this problem is NP-hard and in Σ_2^p , but its precise complexity remains an open problem.

⁶For the sake of illustration, we pick here one of the most simple ones, similar to those used for combinatorial auctions: agents' preferences are numerical (i.e., utility functions) and are represented by a set of propositional formulas, each of which is associated with a weight denoting its importance; the utility of a set of goods is the sum of the weights of the formulas satisfied. Preferences are monotonous if all formulas are positive and all weights are positive. See for instance [Lang, 2004] for a survey of logical languages for compact preference representation.

6 Concluding remarks

We have identified the exact complexity of the key problem of deciding whether there exists an efficient and envy-free allocation when preferences are represented compactly, in several contexts; we have also considered variations of the problem. We have also drawn connections to a well-studied problem in nonmonotonic reasoning. The next step will consist in designing and experimenting algorithms for the search of an EEF allocation (when it exists) and approximation notions for defining optimal allocations when there is no EEF allocation (see [Lipton et al., 2004] for approximate envy-freeness, although not coupled with efficiency).

Acknowledgements: We thank Michel Lemaître for stimulating discussions about fair division and compact representation, and Thibault Gajdos for stimulating discussions about envy-freeness and for pointing to us some relevant papers.

References

- [Bouveret et al., 2005] S. Bouveret, H. Fargier, J. Lang, and M. Lemaître. Allocation of indivisible goods: a general model and some complexity results. In *Proceedings of AAMAS 05*, 2005. Long version available at <http://www.irit.fr/recherches/RPDM/persos/JeromeLang/papers/aig.pdf>.
- [Brams et al., 2000] S. Brams, P. Edelman, and P. Fishburn. Fair division of indivisible items. Technical Report RR 2000-15, C.V. Starr Center for Applied Economics, New York University, 2000.
- [Chevalerey et al., 2004] Y. Chevalerey, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation with k -additive utility functions. In *Proc. DIMACS-LAMSADE Workshop on Computer Science and Decision Theory*, volume 3 of *Annales du LAMSADE*, pages 83–100, 2004.
- [Demko and Hill, 1998] S. Demko and T.P. Hill. Equitable distribution of indivisible items. *Mathematical Social Sciences*, 16:145–158, 1998.
- [Dunne et al., 2005] P. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 2005. To appear.
- [Gottlob, 1992] G. Gottlob. Complexity results for non-monotonic logics. *Journal of Logic and Computation*, 2:397–425, 1992.
- [Herreiner and Puppe, 2002] D. Herreiner and C. Puppe. A simple procedure for finding equitable allocations of indivisible goods. *Social Choice and Welfare*, 19:415–430, 2002.
- [Lang, 2004] J. Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1):37–71, 2004.
- [Lipton et al., 2004] R. Lipton, E. Markakis, E. Mossel, and A. Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of EC'04*, 2004.
- [Papadimitriou, 1994] Ch. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [Reiter, 1980] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

On Maximal Classes of Utility Functions for Efficient one-to-one Negotiation

Yann Chevaleyre
LAMSADE
Université Paris-Dauphine
chevaley@lamsade.dauphine.fr

Ulle Endriss
Department of Computing
Imperial College London
ue@doc.ic.ac.uk

Nicolas Maudet
LAMSADE
Université Paris-Dauphine
maudet@lamsade.dauphine.fr

Abstract

We investigate the properties of an abstract negotiation framework where agents autonomously negotiate over allocations of discrete resources. In this framework, reaching an optimal allocation potentially requires very complex multilateral deals. Therefore, we are interested in identifying classes of utility functions such that any negotiation conducted by means of deals involving only a single resource at a time is bound to converge to an optimal allocation whenever all agents model their preferences using these functions. We show that the class of modular utility functions is not only sufficient but also maximal in this sense.

1 Introduction

The problem of *discrete* resource allocation has recently received much attention from the artificial intelligence community. A large amount of this work is focused on *combinatorial auctions* [Cramton *et al.*, 2005]. In this case, the allocation procedure is centralised, and the so-called *winner determination problem* consists in determining the allocation of resources maximising the auctioneer's revenue.

A different perspective is taken when one assumes that the allocation process is truly distributed, in the sense that agents autonomously negotiate over the bundles of resources they hold. This assumption is justified in many applications where no central authority can be relied on to decide upon the allocation of resources. In this case, the system designer will typically seek to set up the system in such way that it guarantees desirable properties, without directly interfering in the negotiation process itself. In this paper we will make use of such an abstract negotiation framework investigated by a number of authors [Sandholm, 1998; Endriss *et al.*, 2003; Dunne *et al.*, 2005].

To make things more precise, we assume a set of negotiating agents populating the system, and we model their preferences (over different bundles of resources) by means of utility functions. In order to pursue their own interests, agents agree on deals benefitting themselves but without planning ahead (*i.e.* they are both rational and myopic), thereby modifying the allocation of resources. From a global point of view, the quality of an allocation reflects the overall performance of the

system, and the designer will naturally seek to ensure that negotiation converges towards an optimal allocation.

Section 2 introduces the negotiation framework used in this paper. As we shall recall in Section 3, it is known that very complex multilateral deals are potentially required to reach an optimal allocation. When deals are restricted (*e.g.* to a limited number of resources), it is only possible to guarantee an optimal outcome by also restricting the negotiation process to agents whose preferences have certain properties. In this paper, we study the conditions under which negotiation conducted by means of the simplest deals, involving one item at a time (or *1-deal negotiation* for short) still allows us to reach an optimal allocation. Section 4 generalises a result from the literature and shows that modelling preferences with *modular utility functions* is a sufficient condition. However, modularity is not a *necessary* condition. This is demonstrated in Section 5 by means of a counterexample. We also show that there can be *no* condition on utility functions that would be both necessary and sufficient for optimal allocations to be negotiable by means of rational 1-deals. The main contribution of this paper is to show that the class of modular utility functions is *maximal*, in the sense that no class strictly including the modular utility functions would still be sufficient for 1-deal negotiation. The proof detailed in Section 6 shows that, given any non-modular utility function, it is always possible to construct a modular utility function and select a scenario where the optimal allocation cannot be reached by 1-deals. Section 7 concludes.

2 Myopic negotiation over resources

In this section, we introduce the decentralised negotiation framework used throughout this paper and report a number of known technical results. In this framework, a finite set of agents negotiate over a finite set of discrete (*i.e.* non-divisible) resources. A resource *allocation* is a partitioning of the resources amongst the agents (that is, every resource has to be allocated to one and only one agent). As an example, the allocation A defined by $A(i) = \{r_1\}$ and $A(j) = \{r_2, r_3\}$ would allocate resource r_1 to agent i , while resources r_2 and r_3 would be owned by agent j .

We are going to model the preferences of agents by means of *utility functions* mapping bundles of resources to real numbers. Assuming that agents are only concerned with resources they personally own, we will use the abbreviation $u_i(A)$ for

$u_i(A(i))$, representing the utility value assigned by agent i to the bundle it holds for allocation A . The parameters of a negotiation problem are summarised in the following definition:

Definition 1 (Negotiation problems) A negotiation problem is a tuple $\mathcal{P} = \langle \mathcal{R}, \mathcal{A}, \mathcal{U}, A_0 \rangle$, where

- $\mathcal{R}$ is a finite set of indivisible resources;
- $\mathcal{A} = \{1, \dots, n\}$ is a finite set of agents ($n \geq 2$);
- $\mathcal{U} = \langle u_1, \dots, u_n \rangle$ is a vector of utility functions, such that for all $i \in \mathcal{A}$, u_i is a mapping from $2^{\mathcal{R}}$ to $\mathbb{R}$;
- $A_0 : \mathcal{A} \rightarrow 2^{\mathcal{R}}$ is an (initial) allocation.

Agents may agree on a *deal* to exchange some of the resources they possess. It transforms the current allocation of resources A into a new allocation A' ; that is, we can define a deal as a pair $\delta = (A, A')$ of allocations (with $A \neq A'$).

We should stress that this is a *multilateral* negotiation framework. A single deal may involve the displacement of any number of resources between any number of agents. An actual implementation of this abstract framework may, however, not allow for the same level of generality. Sandholm [1998] has proposed a typology of different types of deals, such as *swap deals* involving an exchange of single resources between two agents or *cluster deals* involving the transfer of a set of items from one agent to another. The simplest type of deals are those involving only a single resource (and thereby only two agents).

Definition 2 (1-deals) A 1-deal is a deal $\delta = (A, A')$ resulting in the reallocation of exactly one resource.

The above is a condition on the *structure* of a deal. Other conditions relate to the *acceptability* of a deal to a given agent. We assume that agents are *rational* in the sense of aiming to maximise their individual welfare. Furthermore, agents are assumed to be *myopic*. This means that agents will not accept deals that would reduce their level of welfare, not even temporarily, because they are either not sufficiently able to plan ahead or not willing to take the associated risk (see also [Sandholm, 1998] for a justification of such an agent model in the context of multiagent resource allocation). We will, however, permit agents to enhance deals with *monetary side payments*, in order to compensate other agents for a possible loss in utility. This can be modelled using a *payment function* $p : \mathcal{A} \rightarrow \mathbb{R}$. Such a function has to satisfy the side constraint $\sum_{i \in \mathcal{A}} p(i) = 0$, i.e. the overall amount of money in the system remains constant. If $p(i) > 0$, then agent i pays the amount of $p(i)$, while $p(i) < 0$ means that it receives the amount of $-p(i)$. The following rationality criterion will define the acceptability of deals:

Definition 3 (Individual rationality) A deal $\delta = (A, A')$ is rational iff there exists a payment function p such that $u_i(A') - u_i(A) > p(i)$ for all $i \in \mathcal{A}$, except possibly $p(i) = 0$ for agents i with $A(i) = A'(i)$.

From a system designer's perspective, we are interested in assessing the well-being of the whole society, or *social welfare* [Arrow et al., 2002], which is often defined as the sum of utilities of all the agents.

Definition 4 (Social welfare) The social welfare $sw(A)$ of an allocation A is defined as follows:

$$sw(A) = \sum_{i \in \mathcal{A}} u_i(A)$$

This is the *utilitarian* definition of social welfare. While this is the definition usually adopted in the multiagent systems literature [Wooldridge, 2002], we should stress that also several of the other notions of social welfare developed in the social sciences (e.g. egalitarian social welfare [Arrow et al., 2002]) do have potential applications in the context of multiagent resource allocation.

We conclude this background section by recalling two important results [Sandholm, 1998; Endriss et al., 2003]: the first one makes explicit the connection between the local decisions of agents and the global behaviour of the system, and the second one is the fundamental convergence theorem for this negotiation framework.

Lemma 1 (Individual rationality and social welfare) A deal $\delta = (A, A')$ is rational iff $sw(A) < sw(A')$.

Theorem 1 (Maximising social welfare) Any sequence of rational deals will eventually result in an allocation of resources with maximal social welfare.

The main significance of this latter result, beyond the equivalence of rational deals and social welfare-increasing deals stated in Lemma 1, is that *any* sequence of deals satisfying the rationality criterion will eventually converge to an optimal allocation. There is no need for agents to consider anything but their individual interests. Every single deal is bound to increase social welfare and there are no local minima.

3 Negotiating over one item at a time

While Theorem 1 shows that, in principle, it is always possible to negotiate an allocation of resources that is optimal from a social point of view, deals involving any number of agents and resources may be required to do so [Sandholm, 1998; Endriss et al., 2003]. In particular, the most basic type of deal, which involves moving a single resource from one agent to another and which is the type of deal implemented in most systems realising a kind of *Contract Net* protocol [Smith, 1980], is certainly *not* sufficient for negotiation between agents that are not only rational but also myopic.

This has first been shown by Sandholm [1998] and is best explained by means of an example.¹ Let $\mathcal{A} = \{1, 2, 3\}$ and $\mathcal{R} = \{r_1, r_2, r_3\}$. Suppose the utility functions of these agents are defined as follows (over singleton sets):

$$\begin{array}{lll} u_1(\{r_1\}) = 5 & u_1(\{r_2\}) = 1 & u_1(\{r_3\}) = 0 \\ u_2(\{r_1\}) = 0 & u_2(\{r_2\}) = 5 & u_2(\{r_3\}) = 1 \\ u_3(\{r_1\}) = 1 & u_3(\{r_2\}) = 0 & u_3(\{r_3\}) = 5 \end{array}$$

Furthermore, for any bundle R not listed above, suppose $u_i(R) = 0$ for all $i \in \mathcal{A}$. Let A_0 with $A_0(1) = \{r_2\}$, $A_0(2) = \{r_3\}$ and $A_0(3) = \{r_1\}$ be the initial allocation,

¹A methodology for constructing such examples is easily generated from the proof of the result on the insufficiency of any kind of structurally limited class of deals given by Endriss et al. [2003].

i.e. $sw(A_0) = 3$. The optimal allocation would be A^* with $A^*(1) = \{r_1\}$, $A^*(2) = \{r_2\}$ and $A^*(3) = \{r_3\}$, which yields a social welfare of 15. All other allocations have lower social welfare than A^* . Hence, starting from A_0 , the deal $\delta = (A_0, A^*)$ would be the only deal increasing social welfare. By Lemma 1, δ would also be the only rational deal. This deal, however, involves all three resources and affects all three agents. In particular, δ is not a 1-deal. Hence, if we choose to restrict ourselves to *rational* deals, then 1-deals are not sufficient to negotiate allocations of resources with maximal social welfare.

Of course, for some particular negotiation problems, rational 1-deals *will* be sufficient. The difficulty lies in recognising the problems where this is so. Closely related to this issue, Dunne *et al.* [2005] have shown that, given two allocations A and A' with $sw(A) < sw(A')$, the problem of checking whether it is possible to reach A' from A by means of a sequence of rational 1-deals is NP-hard in the number of resources in the system.

The structural complexity of deals required to be able to guarantee socially optimal outcomes partly stems from the generality of the framework. In particular, so far we have made no assumptions on the structure of utility functions used by the agents to model their preferences. By introducing restrictions on the class of admissible utility functions, it may be possible to ensure convergence to an allocation with maximal social welfare by means of simpler deals. In this paper, we are interested in characterising more precisely those classes of utility functions that permit 1-deal negotiation.

Definition 5 (1-deal negotiation) A class $\mathcal{C}$ of utility functions is said to permit 1-deal negotiation iff any sequence of rational 1-deals will eventually result in an allocation of resources with maximal social welfare whenever all utility functions $\{u_1, \dots, u_n\}$ are drawn from $\mathcal{C}$.

Under this perspective, a relevant result is due to Endriss *et al.* [2003], who show that rational 1-deals are sufficient to guarantee outcomes with maximal social welfare in case all agents use *additive* utility functions.² We are going to prove a slight generalisation of this result in the next section.

4 Modular functions are sufficient

We are now going to define the class of *modular* utility functions. This is an important (see e.g. [Rosenschein and Zlotkin, 1994]), albeit simple, class of functions that can be used in negotiation domains where there are no synergies (neither complementarities nor substitutables) between different resources.

Definition 6 (Modular utility) A utility function u is modular iff the following holds for all bundles $R_1, R_2 \subseteq R$:

$$u(R_1 \cup R_2) = u(R_1) + u(R_2) - u(R_1 \cap R_2) \quad (1)$$

The class of modular functions includes the aforementioned additive functions. This may be seen as follows. Let R be any non-empty bundle of resources and let $r \in R$. Then equation (1) implies $u(R) = u(R \setminus \{r\}) + [u(\{r\}) - u(\{\})]$.

²A utility function is additive iff the utility assigned to a set of resources is always the sum of utilities assigned to its members.

If we apply this step recursively for every resource in R , then we end up with the following equation:

$$u(R) = u(\{\}) + \sum_{r \in R} [u(\{r\}) - u(\{\})] \quad (2)$$

That is, in case $u(\{\}) = 0$, the utility assigned to a set will be the sum of utilities assigned to its members (i.e. u will be additive). Clearly, equation (2) also implies equation (1), i.e. the two characterisations of the class of modular utility functions are equivalent.

It turns out that in domains where all utility functions are modular, it is always possible to reach a socially optimal allocation by means of a sequence of rational deals involving only a single resource each. This is a slight generalisation of a result proved by Endriss *et al.* [2003], and our proof closely follows theirs.

Theorem 2 (Negotiation in modular domains) The class $\mathcal{M}$ of modular utility functions permits 1-deal negotiation.

Proof. By Lemma 1, any rational deal results in a strict increase in social welfare. Together with the fact that the number of distinct allocations is finite, this ensures that there can be no infinite sequence of rational deals (termination). It therefore suffices to show that for any allocation that does not have maximal social welfare there still exists a rational 1-deal that would be applicable.

We are going to use the alternative characterisation of modular utility functions given by equation (2). For any allocation A , let f_A be the function mapping each resource r to the agent i that holds r in situation A . Then, for modular domains, the formula for social welfare (see Definition 4) can be rewritten as follows:

$$sw(A) = \sum_{i \in A} u_i(\{\}) + \sum_{r \in R} u'_{f_A(r)}(\{r\})$$

with $u'_i(R) = u_i(R) - u_i(\{\})$. Now assume we have reached an allocation of resources A that does not have maximal social welfare, i.e. there exists another allocation A' with $sw(A) < sw(A')$. Considering the above definition of social welfare and observing that $\sum_{i \in A} u_i(\{\})$ is a constant that is independent of the current allocation, this implies that at least one resource r must satisfy the inequation $u'_{f_A(r)}(\{r\}) < u'_{f_{A'}(r)}(\{r\})$, i.e. the agent owning r in allocation A values that resource less than the agent owning it in allocation A' . But then the 1-deal consisting of passing r from agent $f_A(r)$ to agent $f_{A'}(r)$ would already increase social welfare and thereby be rational. $\square$

Like Theorem 1, the above establishes an important convergence result towards a global optimum by means of decentralised negotiation between self-interested agents. In addition, provided all utility functions are modular, convergence can be guaranteed by means of a much simpler negotiation protocol, which only needs to cater for agreements on 1-deals (rather than multilateral deals over sets of resources).

5 Modular functions are not necessary

In the previous section we have introduced a class of utility functions (namely modular functions) such that it is possible

to guarantee that sequences of rational 1-deals will converge to an allocation with maximal social welfare under the condition that *all* agents' utilities belong to this class. A natural question to ask would then be whether modularity is also a *necessary* condition in this sense.

It turns out that this is not the case. We demonstrate this by means of the following example. Suppose $\mathcal{R} = \{r_1, r_2\}$ and there are two agents with utility functions u_1 and u_2 :

$$\begin{array}{llll} u_1(\{\}) & = 90 & u_2(\{\}) & = 90 \\ u_1(\{r_1\}) & = 93 & u_2(\{r_1\}) & = 90 \\ u_1(\{r_2\}) & = 95 & u_2(\{r_2\}) & = 90 \\ u_1(\{r_1, r_2\}) & = 98 & u_2(\{r_1, r_2\}) & = 50 \end{array}$$

While u_1 is a modular function, u_2 is not. The optimal allocation is the allocation where agent 1 owns both items. Furthermore, as may easily be checked, any 1-deal that involves moving a single resource from agent 2 to agent 1 is rational. Hence, rational 1-deals are sufficient to move to the optimal allocation for this scenario, despite u_2 not being modular.

In fact, it is possible to show that there can be no class of utility functions that would be both sufficient and necessary in this sense. It suffices to produce two concrete utility functions u_1 and u_2 such that (i) both of them would guarantee convergence if all agents were using them, and (ii) there is a scenario where some agents are using u_1 and others u_2 and convergence is not guaranteed. This is so, because assuming that a necessary and sufficient class exists, (i) would imply that both u_1 and u_2 belong to that class, while (ii) would entail the contrary. We give two such functions for the case of two agents and two resources (the argument is easily augmented to the general case):

$$\begin{array}{llll} u_1(\{\}) & = 0 & u_2(\{\}) & = 0 \\ u_1(\{r_1\}) & = 1 & u_2(\{r_1\}) & = 5 \\ u_1(\{r_2\}) & = 2 & u_2(\{r_2\}) & = 5 \\ u_1(\{r_1, r_2\}) & = 3 & u_2(\{r_1, r_2\}) & = 5 \end{array}$$

The function u_1 is modular, i.e. all agents using that function is a sufficient condition for guaranteed convergence to an optimal allocation by means of rational 1-deals (Theorem 1). Clearly, convergence is also guaranteed if all agents are using u_2 . However, if the first agent uses u_1 and the second u_2 , then the allocation A with $A(1) = \{r_1\}$ and $A(2) = \{r_2\}$ is not socially optimal and the only deal increasing social welfare (and thereby, the only rational deal) would be to swap the two resources simultaneously. Hence, no condition on all agents' utility functions can be both sufficient and necessary to guarantee convergence to an optimal allocation by means of rational 1-deals alone.

Our argument for the inexistence of any such necessary and sufficient condition has directly exploited the fact that we were looking for a *single* condition to be met by the utility functions of *all* agents. The problem could be circumvented by looking for suitable conditions on negotiation problems as a whole, where different utility functions may meet different such conditions. Clearly, such a condition *does* exist. However, the aforementioned result of Dunne *et al.* [2005] on the NP-hardness of checking whether there exists a path of rational 1-deals between two given allocations immediately suggests that verifying whether a given negotiation problem meets any such condition would be intractable.

6 The modular class is maximal

We are now going to prove the main result of this paper, namely the surprising fact that the class of modular utility functions is not only sufficient for 1-deal negotiation but also *maximal* in the sense that no class of utility functions strictly including the modular functions would still be sufficient for 1-deal negotiation. The significance of this result can only be fully appreciated when considered together with the “negative” result on necessary and sufficient conditions discussed in the previous section.

Before stating the main result, we prove the following auxiliary lemma:

Lemma 2 (Alternative characterisation of modularity)

A utility function u is modular iff the following holds for all $R \subseteq \mathcal{R}$ and all $r_1, r_2 \in \mathcal{R}$ with $r_1, r_2 \notin R$ and $r_1 \neq r_2$:

$$u(R \cup \{r_1, r_2\}) = u(R \cup \{r_1\}) + u(R \cup \{r_2\}) - u(R) \quad (3)$$

Proof. To show this, let us recall elementary facts about submodular functions. A function $v : \mathcal{R} \rightarrow \mathbb{R}$ is submodular iff $\forall R_1, R_2 \subseteq \mathcal{R}, v(R_1) + v(R_2) \geq v(R_1 \cup R_2) + v(R_1 \cap R_2)$. It is also known that v is submodular iff $v(R \cup \{r_1\}) + v(R \cup \{r_2\}) \geq v(R \cup \{r_1, r_2\}) + v(R)$ for any $R \subseteq \mathcal{R}, r_1, r_2 \in \mathcal{R} \setminus R$, with $r_1 \neq r_2$ [Nemhauser and Wolsey, 1988, p.662]. Because a function u is modular iff both u and $-u$ are submodular, the lemma holds. $\square$

We are now in a position to prove our theorem on the maximality of the class of modular utility functions with respect to rational negotiation over one resource at a time:

Theorem 3 (Maximality) Let $\mathcal{M}$ be the class of modular utility functions. Then for any class of utility functions $\mathcal{F}$ such that $\mathcal{M} \subset \mathcal{F}$, $\mathcal{F}$ does not permit 1-deal negotiation.

Proof. First observe that for $|\mathcal{R}| \leq 1$, any utility function is modular, i.e. the theorem holds vacuously in these cases. Therefore, without loss of generality, from now on we assume that there are at least two distinct resources in the system.

The proof is constructive. We will show that for any non-modular utility function u_1 on m resources, it is possible to construct a modular utility function u_2 (with $u_i \equiv 0$ for all other agents i) and an initial allocation such that no optimal allocation can be reached by means of 1-deals. This implies that $\mathcal{M} \cup \{u_1\}$ does not permit 1-deals.

Because u_1 is non-modular, Lemma 2 can be applied in the following way: there exist a bundle X and distinct resources $r_1, r_2 \notin X$ such that ϵ , defined as follows, is not equal to 0:

$$\epsilon = u_1(X \cup \{r_1\}) + u_1(X \cup \{r_2\}) - u_1(X) - u_1(X \cup \{r_1, r_2\}) \quad (4)$$

From now on, $A_{12|}$, $A_{1|2}$, $A_{1|2}$ and $A_{2|1}$ will refer to allocations in which r_1 and r_2 belong to one of the first two agents, and in which resources in X are owned by 1, and resources in $Y = \mathcal{R} \setminus (X \cup \{r_1, r_2\})$ by 2, as shown in the following table.

	Agent 1	Agent 2
$A_{12 }$	$\{r_1, r_2\} \cup X$	Y
$A_{1 2}$	X	$\{r_1, r_2\} \cup Y$
$A_{1 2}$	$\{r_1\} \cup X$	$\{r_2\} \cup Y$
$A_{2 1}$	$\{r_2\} \cup X$	$\{r_1\} \cup Y$

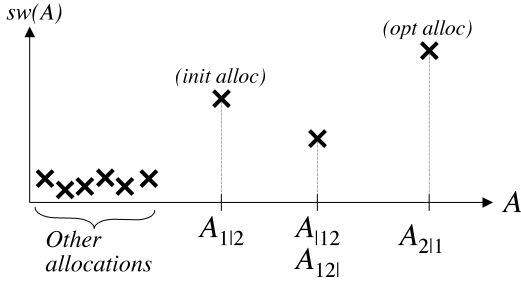


Figure 1: Values of sw for the four allocations (case $\epsilon > 0$).

Let us build a modular utility function u_2 defined as follows:
 $\forall R \in \mathcal{R}$,

$$u_2(R) = \sum_{r \in \{r_1, r_2\} \cap R} \alpha_r + \sum_{r \in R \cap Y} \omega - \sum_{r \in R \cap X} \omega \quad (5)$$

with $\omega = 14 \times \max |u_1| + 1$. Let $\Omega = u_2(Y) = |Y| \times \omega$. As the rest of the proof shall reveal, the value of ω has been chosen such that the social welfare of each of these four allocations is greater than that of any other allocation. Of course, this will imply that the optimal allocation has to be among these four. The values of α_{r_1} and α_{r_2} will be chosen later. The social welfare of each of these four allocations can then be written as follows:

$$\begin{aligned} sw(A_{1|2}) &= \Omega + \alpha_{r_1} + \alpha_{r_2} + u_1(X) \\ sw(A_{12|}) &= \Omega + u_1(X \cup \{r_1, r_2\}) \\ sw(A_{1|2}) &= \Omega + \alpha_{r_2} + u_1(X \cup \{r_1\}) \\ sw(A_{2|1}) &= \Omega + \alpha_{r_1} + u_1(X \cup \{r_2\}) \end{aligned}$$

It remains to be shown that depending on the value of ϵ , we can always choose an initial allocation among these four and values of α_{r_1} and α_{r_2} such that (1) this initial allocation does not have optimal social welfare, (2) there is only one rational deal from this allocation, (3) this deal leads to the optimal allocation but however (4) this rational deal would involve more than one resource. We will have to consider two cases for equation (4): the case of $\epsilon > 0$ and the case of $\epsilon < 0$.

(1st case) Suppose $\epsilon > 0$. Let us choose $\alpha_{r_1} = u_1(X \cup \{r_1\}) - u_1(X) - \frac{\epsilon}{4}$ and $\alpha_{r_2} = u_1(X \cup \{r_1, r_2\}) - u_1(X \cup \{r_1\}) + \frac{\epsilon}{4}$.

Let us first show that the four allocations have a greater social welfare than any other. With the help of equation (4), observe that both $|\alpha_{r_1}|$ and $|\alpha_{r_2}|$ are less than $3 \times \max |u_1|$. Thus, all four allocations have a social welfare of at least $\Omega - |\alpha_{r_1}| - |\alpha_{r_2}| - \max |u_1| \geq \Omega - 7 \times \max |u_1| > \Omega - \frac{\omega}{2}$. All other allocations have a social welfare lower than $\Omega - \omega + |\alpha_{r_1}| + |\alpha_{r_2}| + \max |u_1| \leq \Omega - \omega + 7 \times \max |u_1| < \Omega - \frac{\omega}{2}$. Thus, the social welfare of each of the four allocations is greater than that of any other allocation.

Now let us show that $A_{2|1}$ is the optimal allocation, as illustrated in Figure 1. More precisely, let us show that $sw(A_{1|2}) < sw(A_{12|})$, that $sw(A_{12|}) < sw(A_{1|2})$ and that $sw(A_{1|2}) < sw(A_{2|1})$. By substituting the values of α_{r_1} and

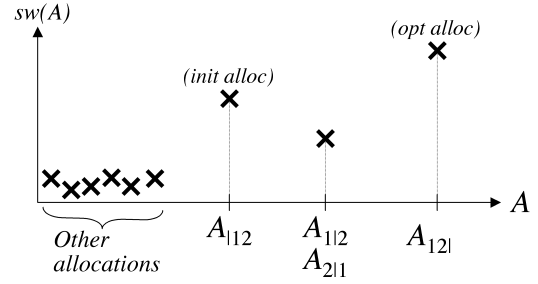


Figure 2: Values of sw for the four allocations (case $\epsilon < 0$).

α_{r_2} and using equation (4), the social welfare of each allocation can be written as follows:

$$\begin{aligned} sw(A_{1|2}) &= \Omega + u_1(X \cup \{r_1, r_2\}) \\ sw(A_{12|}) &= \Omega + u_1(X \cup \{r_1, r_2\}) \\ sw(A_{1|2}) &= \Omega + u_1(X \cup \{r_1, r_2\}) + \frac{\epsilon}{4} \\ sw(A_{2|1}) &= \Omega + u_1(X \cup \{r_1\}) + u_1(X \cup \{r_2\}) \\ &\quad - u_1(X) - \frac{\epsilon}{4} \\ &= \Omega + u_1(X \cup \{r_1, r_2\}) + \frac{3}{4}\epsilon \end{aligned}$$

Here, $A_{2|1}$ is clearly the optimal allocation. If we choose $A_{1|2}$ as the initial allocation, then the only 1-deals involving resources r_1 or r_2 are $\delta(A_{1|2}, A_{12|})$ and $\delta(A_{1|2}, A_{12|})$. These deals decrease social welfare, and thus are not individually rational by Lemma 1. Thus, it is not possible to reach the optimal allocation $A_{2|1}$ starting from $A_{1|2}$ using only 1-deals.

(2nd case) Suppose $\epsilon < 0$. Let us choose $\alpha_1 = u_1(X \cup \{r_1\}) - u_1(X) - \frac{\epsilon}{4}$ and $\alpha_2 = u_1(X \cup \{r_2\}) - u_1(X) - \frac{\epsilon}{4}$.

Note that again, both $|\alpha_{r_1}|$ and $|\alpha_{r_2}|$ are less than $3 \times \max |u_1|$. Thus, by the same argument as in the first case, the four allocations all have greater social welfare than any other allocation.

The optimal allocation is now $A_{12|}$. To see this, let us show that $sw(A_{1|2}) < sw(A_{12|})$, that $sw(A_{2|1}) < sw(A_{12|})$, and that $sw(A_{12|}) < sw(A_{12|})$ as illustrated in Figure 2.

$$\begin{aligned} sw(A_{1|2}) &= \Omega + u_1(X \cup \{r_1\}) + u_1(X \cup \{r_2\}) \\ &\quad - u_1(X) - \frac{\epsilon}{2} \\ sw(A_{12|}) &= \Omega + u_1(X \cup \{r_1, r_2\}) \\ &= \Omega + u_1(X \cup \{r_1\}) + u_1(X \cup \{r_2\}) \\ &\quad - u_1(X) - \epsilon \\ sw(A_{1|2}) &= \Omega + u_1(X \cup \{r_1\}) + u_1(X \cup \{r_2\}) \\ &\quad - u_1(X) - \frac{\epsilon}{4} \\ sw(A_{2|1}) &= \Omega + u_1(X \cup \{r_1\}) + u_1(X \cup \{r_2\}) \\ &\quad - u_1(X) - \frac{\epsilon}{4} \end{aligned}$$

Here, $A_{12|}$ is clearly the optimal allocation. If we choose $A_{1|2}$ as the initial allocation, then the only 1-deals involving

r_1 or r_2 are $\delta(A_{1|2}, A_{1|2})$ and $\delta(A_{1|2}, A_{2|1})$. These deals decrease social welfare, and thus are not individually rational by Lemma 1. Thus, it is not possible to reach the optimal allocation $A_{12|}$ starting from $A_{1|2}$ using only 1-deals. $\square$

Why is this result significant? As argued earlier, while the full abstract negotiation framework introduced at the beginning of this paper would be difficult to implement, designing a system that only allows for pairs of agents to agree on deals over one resource at a time is entirely feasible. As we would like to be able to guarantee socially optimal outcomes in as many cases as possible, also for such a restricted negotiation system, we would like to be able to identify the largest possible class of utility functions for which such a guarantee can be given. However, our discussion in Section 5 has shown that there can be no class of utility functions that *exactly* characterises the class of negotiation problems for which negotiating socially optimal allocations by means of rational 1-deals is always possible. Still, there *are* classes of utility functions that permit 1-deal negotiation. As shown by Theorem 2, the class of modular functions is such a class and it is a very natural class to consider. An obvious question to ask is therefore whether this class can be enlarged in any way without losing the desired convergence property.

Our proof of Theorem 3 settles this question by giving a negative answer: For any agent with a non-modular utility function there exist modular utility functions (for the other agents) and an initial allocation such that rational 1-deals alone do not suffice to negotiate an allocation of resources with maximal social welfare. There may well be further such classes (that are both sufficient and maximal), but the class of modular functions is one that is particularly natural and useful in the context of modelling agent preferences.

7 Conclusion

This paper makes a contribution to the theoretical analysis of a negotiation framework where rational but myopic agents agree on a sequence of deals regarding the reallocation of a number of discrete resources. We have shown that the use of *modular* utility functions to model agent preferences is a *sufficient* condition to guarantee final allocations with maximal social welfare in case agents only negotiate *1-deals* (involving one resource each). We have then seen that this is, however, not a *necessary* condition for optimal outcomes and, indeed, there can be no condition on (individual) utility functions that would be both necessary and sufficient in this sense. We have therefore concentrated on showing that the class of modular functions is *maximal*, *i.e.* no strictly larger class of functions would still permit an optimal allocation to be found by means of rational 1-deals in all cases.

We consider this not only a surprising result, but also a useful characterisation of negotiation domains that can be handled reliably using simple negotiation protocols, catering only for *Contract Net*-like deals over single items between pairs of agents, rather than the full range of multilateral deals foreseen in the abstract framework. Such theoretical results affect both the design of agents and of negotiation mechanisms. For instance, if a given mechanism can only handle 1-deals,

then it may be inappropriate to design myopic agents with very rich preference structures to use such a mechanism.

In a companion paper [Chevalerey *et al.*, 2005], we prove a generalisation of Theorem 1 which shows that rational deals involving at most k resources each are sufficient for convergence to an optimal allocation in case all utility functions are *additively separable* with respect to a common partition of $\mathcal{R}$ (*i.e.* synergies across different parts of the partition are not possible and overall utility is defined as the sum of utilities for the different sets in the partition [Fishburn, 1970]), and each set in this partition has at most k elements. The arguments against the existence of sufficient conditions for negotiation over k items at a time that are also necessary generalise in the expected manner. An important issue that remains to be investigated in the future therefore is to see whether it is possible to derive a similar maximality property as the one proved in this paper for this richer class of utility functions.

Another topic for future work would be to investigate what classes of utility functions are sufficient and maximal for negotiating socially optimal allocations by means of 1-deals without side payments.

References

- [Arrow *et al.*, 2002] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare*, volume 1. North-Holland, 2002.
- [Chevalerey *et al.*, 2005] Y. Chevalerey, U. Endriss, J. Lang, and N. Maudet. Negotiating over small bundles of resources. In *Proc. AAMAS-2005*. ACM Press, 2005.
- [Cramton *et al.*, 2005] P. Cramton, Y. Shoham, and R. Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2005. To appear.
- [Dunne *et al.*, 2005] P. E. Dunne, M. Wooldridge, and M. Laurence. The complexity of contract negotiation. *Artificial Intelligence*, 2005. To appear.
- [Endriss *et al.*, 2003] U. Endriss, N. Maudet, F. Sadri, and F. Toni. On optimal outcomes of negotiations over resources. In *Proc. AAMAS-2003*. ACM Press, 2003.
- [Fishburn, 1970] P. C. Fishburn. *Utility Theory for Decision Making*. John Wiley and Sons, 1970.
- [Nemhauser and Wolsey, 1988] G. L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, 1988.
- [Rosenschein and Zlotkin, 1994] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. MIT Press, 1994.
- [Sandholm, 1998] T. W. Sandholm. Contract types for satisficing task allocation: I Theoretical results. In *Proc. AAAI Spring Symposium: Satisficing Models*, 1998.
- [Smith, 1980] R. G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.
- [Wooldridge, 2002] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.

Two-Sided Bandits and the Dating Market

Sanmay Das

Center for Biological and Computational Learning
and Computer Science and Artificial Intelligence Lab
Massachusetts Institute of Technology
Cambridge, MA 02139
sanmay@mit.edu

Emir Kamenica

Department of Economics
Harvard University
Cambridge, MA 02138
kamenica@fas.harvard.edu

Abstract

We study the decision problems facing agents in repeated matching environments with learning, or *two-sided bandit problems*, and examine the dating market, in which men and women repeatedly go out on dates and learn about each other, as an example. We consider three natural matching mechanisms and empirically examine properties of these mechanisms, focusing on the asymptotic stability of the resulting matchings when the agents use a simple learning rule coupled with an ϵ -greedy exploration policy. Matchings tend to be more stable when agents are patient in two different ways — if they are more likely to explore early or if they are more optimistic. However, the two forms of patience do not interact well in terms of increasing the probability of stable outcomes. We also define a notion of regret for the two-sided problem and study the distribution of regrets under the different matching mechanisms.

1 Introduction

This paper analyzes the learning and decision problems of agents in a model of one-to-one two-sided matching, focusing on the role of the matching mechanism and the exploration-exploitation tradeoff. We consider a repeated game in which agents gain an uncertain payoff from being matched with a particular person on the other side of the market in each time period. A natural example of such a situation is the dating market, in which men and women repeatedly go out on dates and learn about each other. Another example is a spot labor market, in which employers and employees are matched for particular job contracts. A matching mechanism is used to pair the agents. For example, we can consider a mechanism in which all the women decide which man to “ask out,” and then each man selects a woman from his set of offers, with the rejected women left unmatched for that period.

Standard models of matching in economics [Roth and Sotomayor, 1990] almost always assume that each agent knows his or her preferences over the individuals on the other side of the market. This assumption is too restrictive for many markets, including the market for romantic partnerships. Our

model is driven by the need to relax this assumption. The existing literature on two-sided search with nontransferable utility (for example [Burdett and Wright, 1998]) assumes matching is exogenous and random. Our problem is more deeply related to bandit problems [Berry and Fristedt, 1985], in which an agent must choose which arm of an n -armed bandit to pull in order to maximize long-term expected reward, taking into account the tradeoff between *exploring*, that is learning more about the reward distribution for each arm, and *exploiting*, pulling the arm with the maximum expected reward. However, in our model the arms themselves have agency — they can decide whether to be pulled or not, or whom to be pulled by, and they themselves receive rewards based on who the puller is. This motivates our formulation of the problem as a “two-sided” bandit problem.

In principle, we would like to examine the equilibrium behavior of agents in two-sided bandit problems. However, perfect Bayesian equilibria of the game we formulate are prohibitively hard to compute. Because of this difficulty, the approach we use is closely related to the theory of learning in games [Fudenberg and Levine, 1998], which considers more generally how individual learning rules affect outcomes in games and whether agents reach static equilibria.

In this paper we formally define the two-sided bandit problem and describe three important matching mechanisms. We define regret as the difference between the actual reward received and the reward under the stable matching, i.e. a matching such that there is no pair that would rather be with each other than with whom they are matched. We experimentally analyze the asymptotic stability and regret properties when agents use ϵ -greedy learning algorithms adapted to the different matching mechanisms.

The Gale-Shapley mechanism [Gale and Shapley, 1962], which yields stability when information is complete and preferences are truthfully revealed, converges quickly to stable matchings, whereas mechanisms that are more realistic for the dating example, in which women make single offers to the men, do not always converge to stable matchings. Asymptotically stable matches are more likely when agents explore more early on. They are also more likely when agents are optimistic (again, early on) — that is, they assume a higher probability of their offer being accepted or an offer being made to them than is justified by the past empirical frequencies of these events. However, increased optimism does not

interact well with increased exploration in terms of stability; the probability of stability is actually higher for lower exploration probabilities when optimism is greater.

2 The Model

There are M men and W women, who interact for T time periods. v_{ij}^m is the value of woman j to man i , and v_{ij}^w is the value of man j to woman i . These values are constant through time. In each period, men and women are matched to each other through a *matching mechanism*. A matching is a pairing between men and women in which each woman is paired with one or zero men and each man is paired with one or zero women. Formally, a matching mechanism is a mapping from agents' actions to a matching. If man i is matched with woman j in period t , he receives $v_{ij}^m + \epsilon_{ijt}^m$, and she receives $v_{ji}^w + \epsilon_{jit}^w$. If unmatched, individual i receives some constant value K_i .

For our empirical analysis we put some structure on the reward processes and the matching mechanism. First, we make the strong assumption of sex-wide homogeneity of preferences. That is, every man is equally "good" for each woman and vice versa — there are no idiosyncratic preferences and there are no couples who "get along" better than others. Formally, $v_{ij}^m = v_j^m \forall i$ and $v_{ij}^w = v_j^w \forall i$. We also assume that people dislike being single: $\forall i, K_i \ll \min_j v_{ij}^z \forall i, z \in \{m, w\}$, and that the noise terms ϵ are independently and identically distributed.¹ Extensions to more general preferences are straightforward.²

We consider three matching mechanisms. Without loss of generality, we assume that women always ask men out.

Gale-Shapley matching Each agent submits a list of preferences and a centralized matching procedure produces a matching based on these lists. The Gale-Shapley algorithm [Gale and Shapley, 1962] guarantees a matching that is stable under the submitted preferences. The man-optimal variant yields the stable matching that is optimal for men, and the woman-optimal variant the stable matching that is optimal for women. We use the woman-optimal variant.

Simultaneous offers Each woman independently chooses one man to make an offer to. Each man selects one of the offers he receives. Women who are rejected are unmatched for the period, as are the men who receive no offers.

Sequential offers Each woman independently chooses one man to make an offer to. The offers are randomly ordered and men must decide on these "exploding" offers without knowing what other offers are coming. Men see all the offers they receive, including ones that arrive after they accept. If an offer is rejected the woman making the offer is unmatched in that period. A man is unmatched if he rejects all offers he receives.

¹Another plausible and interesting structure to consider is pairwise homogeneity of preferences with $v_{ij}^m = v_{ji}^w \forall i, j$.

²There is always a unique stable matching under the assumed preference structure. With multiple stable matches, we would need to use a different notion of regret, as discussed later.

Intuitively, it is useful to think of the simultaneous choice mechanism as capturing a situation in which women ask men out over e-mail and each man can review all his offers before making a decision, while the sequential choice mechanism captures the situation where women ask men out over the telephone. We are particularly interested in these two matching mechanisms because they are more plausible descriptions of reality than a centralized matchmaker and do not require agents to reveal their preferences to a third party.

3 The Decision and Learning Problems

We first describe the decision problems agents face at each time step if they want to optimize their myopic reward in that time step. After this we discuss the exploration-exploitation issues under the different matching mechanisms and describe specific forms of the ϵ -greedy algorithm.

Let $Q_{ij}^{\{m,w\}}$ denote man (woman) i 's estimate of the value of going out with woman (man) j , p_{ij}^w denote woman i 's estimate of the probability that man j will go out with her if she asks him out and p_{ij}^m denote man i 's estimate of the probability that woman j will ask him out under the sequential choice mechanism.

3.1 Women's Decision Problem

Under Gale-Shapley matching, women's action space is the set of rankings of men. Under the other two mechanisms, a woman chooses which man to make an offer to. She must base her decision on any prior beliefs and the history of rewards she has received in the past. She has to take into account both the expected value of going on a date with each man and (for the non-Gale-Shapley mechanisms) the probability that he will accept her offer.

Under the woman-optimal variant of the Gale-Shapley mechanism, the dominant myopic strategy, and thus the greedy action, is for woman i to rank the men according to the expected value of going out with each of them, Q_{ij}^w . For the other two mechanisms, the greedy action is to ask out man $j = \arg \max_j (p_{ij}^w Q_{ij}^w)$.

3.2 Arms With Agency: Men's Decision Problem

The action space of men, the arms of the bandit, may be constrained by women's actions. The decision problem faced by a man depends on the matching mechanism used. Under the woman-optimal Gale-Shapley mechanism, men may have an incentive to misrepresent their preferences, but since the sex-wide homogeneity of preferences ensures a unique stable matching [Roth and Sotomayor, 1990], this is less likely to be a problem.³ So, the greedy action for man i under Gale-Shapley is to rank women based on their Q_{ij} 's.

With the simultaneous choice mechanism, in each time period a man receives a list of women who have made him an offer. He must decide which one to accept. This is a bandit problem with a different subset of the arms available at each time period. The greedy action is to accept the woman $j = \arg \max_j Q_{ij}^m$.

³Specifically, if the *submitted* rankings satisfy sex-wide homogeneity, man- and woman-optimal algorithms yield the same matching so truthtelling is the dominant myopic strategy for men.

Under the sequential choice mechanism, a man might receive multiple offers within a time period, and each time he receives an offer he has to decide immediately whether to accept or reject it, and he may not renege on an accepted offer. The information set he has is the list of women who have asked him out so far. For each woman who has not asked him out, it could either be that she chose not to make him an offer, or that her turn in the ordering has not arrived yet. We can formulate the man's value function heuristically. Let i be the index of the man, let S be the set of women who have asked him out so far, and let h be the woman currently asking him out ($h \in S$).

$$V(S, h) = \max\{Q_{ih}^m, \sum_{k \notin S} \Pr(k \text{ next woman to ask } i \text{ out}) V(S \cup \{k\}, k)\}$$

The base cases are $V(\mathcal{W}, h) = Q_{ih}^w$ where $\mathcal{W}$ is the set of all women. The greedy action is to accept an offer when

$$Q_{ih}^m > \sum_{k \notin S} \Pr(k \text{ next woman to ask } i \text{ out}) V(S \cup \{k\}, k)$$

The relevant probabilities are:

$$\Pr(k \text{ next woman to ask } i \text{ out}) = \sum_{T \in \text{Perm}(S')} \left[\frac{1}{|S'|} \left(\prod_{j \text{ preceding } k \text{ in } T} (1 - p_{ij}^m) \right) p_{ik}^m \right]$$

where $S' = \mathcal{W} \setminus S$. This is a variant of the classic secretary problem [Gilbert and Mosteller, 1966]. We are not sure at the moment if this particular form can be simplified to yield a closed form solution for the value or decision function.

3.3 The Exploration-Exploitation Tradeoff

Women and men both have to consider the exploration-exploitation tradeoff (summarized in [Sutton and Barto, 1998]). *Exploitation* means maximizing expected reward in the current period (also called the greedy choice), and is solved as above. *Exploration* happens when an agent does not select the greedy action, but instead selects an action that has lower expected value in the current period in order to learn more and increase future rewards.

The one-sided version of the exploration-exploitation problem is central to n -armed bandit problems [Berry and Fristedt, 1985; Gittins and Jones, 1974, *inter alia*]. An n -armed bandit is defined by random variables $X_{i,t}$ where $1 \leq i \leq n$ is the index of the arm of the bandit, and $X_{i,t}$ specifies the payoff received from pulling arm i at time t . The distribution of some or all of the $X_{i,t}$ is unknown so there is value to exploring. The agent pulls the arms sequentially and wishes to maximize the discounted sum of payoffs. In our model, if there is a single woman and n men, the woman faces a standard n -armed bandit problem.

One of the simplest techniques used for bandit problems is the so-called ϵ -greedy algorithm. This algorithm selects the arm with highest expected value with probability

$1 - \epsilon$ and otherwise selects a random arm. Although simple, the algorithm is very successful in most empirical problems, and we therefore use it in our experiments. We have also experimented with alternatives like softmax-action selection with Boltzmann distributions [Sutton and Barto, 1998; Luce, 1959] and the Exp3 algorithm [Auer *et al.*, 2002]. These do not improve upon the empirical performance of ϵ -greedy in our simulations.

Under each matching mechanism the exploratory action is to randomly select an action, other than the greedy one, from the available action space. Since the value of exploration decreases as learning progresses, we let ϵ decay exponentially over time which also ensures that the matchings converge.

At this stage we cannot solve for the perfect Bayesian equilibrium set. We believe, however, that if the agents are sufficiently patient and the horizon is sufficiently long, the matchings will converge to stability on the equilibrium path. Solving for the equilibrium set would enable us to explicitly characterize the differences between equilibrium behavior and behavior under the ϵ -greedy algorithm.

The two-sided nature of the learning problem leads to non-stationarities. Under the sequential and simultaneous mechanisms, women need to consider the reward of *asking out* a particular man, not the reward of going out with him. The reward of asking out a particular man depends on the probability that he will accept the offer. Thus, the reward distribution changes based on what the men are learning, introducing an externality to the search process. The same applies to men under the sequential mechanism since the probability that a particular woman will ask a man out changes over time. This is a problem of coordinated learning that is related to the literature on learning in games [Fudenberg and Levine, 1998] and to reinforcement learning of nonstationary distributions in multiagent environments [Bowling and Veloso, 2002]. Some recent work by Auer *et al.* on "adversarial" bandit problems, which makes no distributional assumptions in deriving regret bounds, is relevant in this context [Auer *et al.*, 2002].

Since the underlying v_{ij} 's are constant we define Q_{ij} as person i 's sample mean of the payoff of *going out* with person j . In order to deal with the nonstationarity of p_{ij} 's, on the other hand, we use a fixed learning rate for updating the probabilities which allows agents to forget the past more quickly:

$$p_{ij}[t] = (1 - \eta)p_{ij}[t - 1] + \eta I[\text{offer made / accepted}]$$

where η is a constant and I is an indicator function indicating whether a man accepted an offer (for the woman's update, applied only if woman i made an offer to man j at time t) or whether a woman made an offer to a man (for the man's update, applied at each time period t).

4 Empirical Results

Our simulations involve a market with 5 women and 5 men. The agents use η of 0.05 for updating their probability estimates and the probability of exploration evolves with time as $\epsilon_t = \epsilon^{t/1000}$. Agents have true values $v_0^m = v_0^w = 10, v_1^m = v_1^w = 9, v_2^m = v_2^w = 8, v_3^m = v_3^w = 7, v_4^m = v_4^w = 6$. We assume $K_i = 0 \forall i$. The noise signals $\epsilon_{ijt}^{\{m,w\}}$ are i.i.d. and

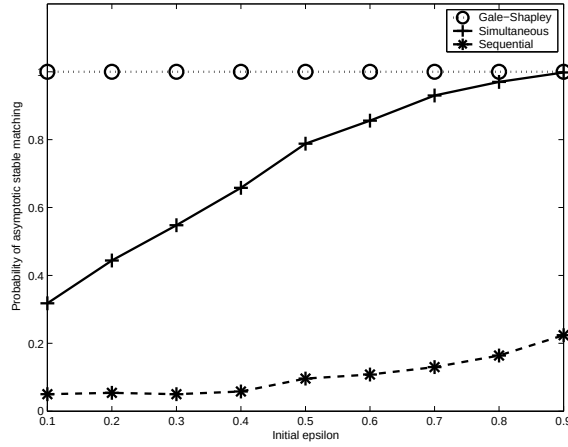


Figure 1: Probability of a stable (asymptotic) matching as a function of initial value of ϵ

drawn from a normal distribution with mean 0. Unless otherwise specified, the standard deviation of the noise distribution is 0.5. Reported results are averages from 500 simulations, each lasting a total of 30,000 time steps. Initial values of Q_{ij} are sampled from a uniform [6, 10] distribution and initial values of p_{ij} are sampled from a uniform [0, 1] distribution.

Our experiments show that settings in which agents are matched using the Gale-Shapley mechanism always result in asymptotically stable matchings, even for very small initial values of ϵ such as 0.1. After a period of exploration, where the agents match up with many different partners and learn their preferences, agents start pairing up regularly with just one partner, and this is always the agent with the same ranking on the other side. This indicates that agents are generally successful at learning their preferences. Interestingly, even if only one side explores (that is, either men or women always pick the greedy action), populations almost always converge to stable matchings, with a slight decline in the probability of stability when only men explore (under the woman-optimal matching algorithm, women’s rankings can have a greater effect on the matching than men’s rankings).

The probabilities of convergence under the simultaneous and sequential choice mechanisms are significantly lower, although they increase with larger initial values of ϵ . We can see this behavior in Figure 1, which also reveals that the probability of convergence to a stable matching is much higher with the simultaneous choice mechanism. Table 1 shows these probabilities as well as the score, which is a measure of how large the deviation from the stable matching is. If men and women are indexed in order of their true value ranking, the score for a matching is defined as $\frac{1}{W} \sum_{i \in \mathcal{W}} |i - \text{Partner}(i)|$ where $\text{Partner}(i)$ is the true value ranking of the man woman i is matched with, and $\mathcal{W}$ is the set of all women.

It is also interesting to look at who benefits from the instabilities. In order to do this, we define a notion of regret for an agent as the (per unit time) difference between the reward under the stable matching and the actual reward received (a negative value of regret indicates that the agent did better than

ID	Simultaneous Woman’s	Regret Man’s	Sequential Woman’s	Regret Man’s
0	0.126	0.126	0.578	0.552
1	0.090	0.278	-0.023	0.009
2	0.236	0.136	-0.153	-0.148
3	0.238	-0.126	-0.005	-0.024
4	-0.690	-0.414	-0.171	-0.187

Table 2: Distribution of regret under simultaneous choice ($\epsilon = 0.1$) and sequential choice ($\epsilon = 0.9$) mechanisms

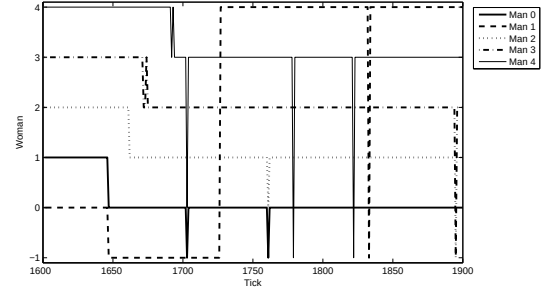


Figure 2: A “phase transition”: men and women are ranked from 0 (highest) to 4 (lowest) with -1 representing the unmatched state. The graph shows the transition to a situation where the second highest ranked man ends up paired with the lowest ranked woman

(s)he would have done under the stable matching). This definition is unambiguous with sex-wide homogeneity of preferences since there is only one stable matching, but could be problematic in other contexts, when there may be multiple stable matchings. In such cases it might make sense to analyze individual agent performance in terms of the difference between average achieved reward and expected reward under one of the stable matchings depending on context.

In the case of sex-wide homogeneity of preferences we of course expect that regret will be greater for more desirable agents since they have more to lose when their value is not known. Table 2 shows the distribution of regrets for simultaneous and sequential choice. The regrets are averaged over the last 10,000 periods of the simulation. Under simultaneous choice, the worst woman benefits at the expense of all other women while the worst two men benefit at the expense of the top three. Under sequential choice, other agents benefit at the expense of the best ones. Further research is needed to better understand this apparent difference in the distribution of regrets under the two mechanisms.

Figure 2 shows interesting dynamic behavior in one particular simulation where the second best man ends up with the worst woman. The graph shows which man is matched with which woman at each time period. The lines represent the men, and the numbers on the Y axis represent the women. The value -1 represents the state of being unmatched in that period for a man. The men and women are ranked from 0 (best) to 4 (worst). Initially, the second best man is paired with the best woman so he keeps rejecting offers from all the

ϵ	Simultaneous Choice		Sequential Choice		Gale-Shapley	
	Pr (stability)	Score	Pr (stability)	Score	Pr (stability)	Score
.1	0.318	0.4296	0.050	0.9688	1.000	0.0000
.2	0.444	0.3832	0.054	0.9280	1.000	0.0000
.3	0.548	0.2920	0.050	0.8560	1.000	0.0000
.4	0.658	0.1880	0.058	0.8080	1.000	0.0000
.5	0.788	0.0992	0.096	0.7448	1.000	0.0000
.6	0.856	0.0672	0.108	0.7064	1.000	0.0000
.7	0.930	0.0296	0.130	0.6640	1.000	0.0000
.8	0.970	0.0120	0.164	0.5848	1.000	0.0000
.9	0.998	0.0008	0.224	0.4912	1.000	0.0000

Table 1: Convergence to stability as a function of ϵ

σ	Pr (stability)	Score
0.5	0.658	0.1880
1.0	0.636	0.1952
1.5	0.624	0.2120
2.0	0.600	0.2328

Table 3: Convergence to stability as a function of σ with simultaneous choice and initial $\epsilon = 0.4$

other women. These women thus learn that he is extremely particular about whom he dates and there is no point in asking him out. When the best woman finally learns that she can get a better man this triggers a chain of events in which all the men sequentially move to the woman ranked one higher than the one they were seeing. However, all the women have such a low opinion of the second best man that he ends up getting matched with the very worst woman. The matching shown at the end of the graph is the final asymptotic matching in this simulation. Note that the gender asymmetry (women ask men out) precludes this from happening to a woman.

Finally, we analyze how the noise distribution affects the probability of stability. We expect that there will be less convergence to stability when the signals are less precise. We ran experiments in which the standard deviation of the noise distribution was changed while holding other factors constant. We used an initial ϵ of 0.4 and the same underlying values as above. Table 3 shows the results using the simultaneous choice mechanism. We vary the standard deviation from one half of the distance between the two adjacent true values (0.5) to twice that distance (2.0), and the probability of stability falls by less than 10%. This suggests that the instabilities arise mostly from the structure of the problem and the non-stationarity of probability estimates rather than from the noise in the signals of value.

5 Optimism and Exploration

The insight that instabilities arise mostly from the structure of the problem suggests an alternative method for engineering asymptotic stability into the system. Suppose agents are initially optimistic and their level of optimism declines over time. This is another form of patience — a willingness to wait for the best — and it should lead to more stable outcomes.

Optimism can be represented by a systematic overestimation of the probability that your offer will be accepted or that

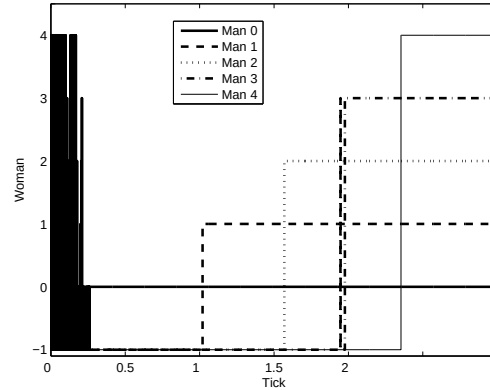


Figure 3: The mechanism of stability with optimism: agents keep trying better ranked agents on the other side until they finally “fall” to their own level

an offer will be made to you. We explore this empirically with the sequential choice mechanism. Instead of using the learned values of p_{ij} as previously defined, agents instead use an optimistic version. At time t , both men and women use the optimistic probability estimate:

$$p'_{ij} = \alpha_t + (1 - \alpha_t)p_{ij}$$

in decision making (the actual p_{ij} is maintained and updated as before). α_t should decline with time. In our simulations $\alpha_0 = 1, \alpha_T = 0$ (where T is the length of the simulation) and α declines linearly with t . There are no other changes to any of the decision-making or learning procedures.

Figure 3 shows the process by which agents converge to asymptotic matchings (in this case a stable one) with the optimistic estimates. The structure of the graph is the same as that in Figure 2. Essentially, each agent keeps trying for the best agent (s)he can match with until the optimism parameter has declined sufficiently so (s)he “falls” to the equivalently ranked agent on the other side of the market. Figure 4 shows that agents are considerably more likely to converge asymptotically to stable matchings using this algorithm for any value of the initial exploration probability. Of course, this convergence comes at the expense of utility achieved in the period before the agents settle down.

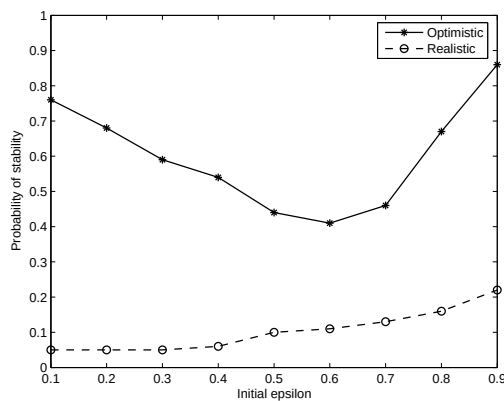


Figure 4: Probability of convergence to stability for different initial values of epsilon with all agents using the optimistic algorithm versus all agents using the realistic algorithm

The surprising feature in Figure 4 is that stable matchings are more likely with smaller initial exploration probabilities. The V-shape of the graph shows that the probability of stability declines with increasing exploration up to an initial ϵ value of 0.6, before starting to increase again, in contrast to the monotonically increasing probability of stability without optimism. This can be explained in terms of the fact that a small level of exploration is sufficient for agents to learn their preferences. Beyond that, additional exploration becomes counterproductive because the probability estimates at the key stages become less reliable.

6 Conclusions and Future Work

We have defined two-sided bandit problems, a new class of problems in multi-agent learning and described the properties of three important matching mechanisms with ϵ -greedy learning rules. Two-sided bandit problems are of great relevance for social science in general and the search for marriage partners in particular. The social norms governing exploration before marriage have been changing rapidly over the last few decades and until now we have had no formal structure within which to study the sources and consequences of these changes. Our model is also more generally applicable to two-sided markets in which agents have to learn about each other.

This paper only scratches the surface of a large and potentially fruitful set of theoretical and empirical questions. We are exploring learning algorithms that would allow agents to perform well⁴ across a broad range of environments without having to make assumptions about the decision-making algorithms or learning processes of other agents. Another direction of research is to explicitly characterize equilibria in simpler settings. We are also interested in more complex versions of the problem that allow for a greater diversity of preferences and a larger number of agents.

⁴In the sense of regret minimization.

Acknowledgements

We would like to thank David Laibson, Sayan Mukherjee, Tommy Poggio, Al Roth and the referees for useful comments. SD acknowledges grants to CBCL from Merrill-Lynch, the National Science Foundation, the Center for e-Business at MIT, the Eastman Kodak Company, Honda R&D Co, and Siemens Corporate Research, Inc. EK acknowledges support from the National Science Foundation Graduate Research Fellowship, the National Institute on Aging (Grant # T32-AG00186), the Institute for Humane Studies, and the Chiles Foundation.

References

- [Auer *et al.*, 2002] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
- [Berry and Fristedt, 1985] D. A. Berry and B. Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Monographs on Statistics and Applied Probability. Chapman and Hall, London, UK, 1985.
- [Bowling and Veloso, 2002] M. Bowling and M. M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.
- [Burdett and Wright, 1998] K. Burdett and R. Wright. Two-sided search with nontransferable utility. *Review of Economic Dynamics*, 1:220–245, 1998.
- [Fudenberg and Levine, 1998] D. Fudenberg and D. Levine. *The Theory of Learning in Games*. MIT Press, Cambridge, MA, 1998.
- [Gale and Shapley, 1962] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.
- [Gilbert and Mosteller, 1966] J. Gilbert and F. Mosteller. Recognizing the maximum of a sequence. *Journal of the American Statistical Association*, 61:35–73, 1966.
- [Gittins and Jones, 1974] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. In J. Gani, K. Sakadi, and I. Vinczo, editors, *Progress in Statistics*, pages 241–266. North Holland, Amsterdam, 1974.
- [Luce, 1959] D. Luce. *Individual Choice Behavior*. Wiley, New York, 1959.
- [Roth and Sotomayor, 1990] A. E. Roth and M. Sotomayor. *Two-Sided Matching: A Study in Game-Theoretic Modeling and Analysis*. Econometric Society Monograph Series. Cambridge University Press, Cambridge, UK, 1990.
- [Sutton and Barto, 1998] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Multi-agent Coordination using Local Search

Boi Faltings and Quang-Huy Nguyen

Ecole Polytechnique Fédérale de Lausanne (EPFL)

Artificial Intelligence Laboratory (LIA)

CH-1015 Lausanne, Switzerland

{boi.faltings, quanghuy.nguyen}@epfl.ch

Abstract

We consider the problem of coordinating the behavior of multiple self-interested agents. It involves constraint optimization problems that often can only be solved by local search algorithms.

Using local search poses problems of incentive-compatibility and individual rationality. We thus define a weaker notion of bounded-rational incentive-compatibility where manipulation is made impossible with high probability through computational complexity. We observe that in real life, manipulation of complex situations is often impossible because the effect of the manipulation cannot be predicted with sufficient accuracy. We show how randomization schemes in local search can make predicting its outcome hard and thus form a bounded-rational incentive-compatible coordination algorithm.

1 Introduction

There are many practical settings where multiple self-interested agents have to coordinate their actions. This coordination often involves joint decisions about resource allocation, scheduling and planning that can be formulated as constraint optimization problems. We thus extend the standard definition of constraint optimization to the multi-agent setting as follows:

Definition 1.1 A discrete multi-agent constraint optimization problem (MCOP) is a tuple $\langle A, X, D, C, R \rangle$ where:

- $A = \{A_1, \dots, A_m\}$ is a set of agents.
- $X = \{x_1, \dots, x_n\}$ is a set of variables.
- $D = \{d_1, \dots, d_n\}$ is a set of domains of the variables, each given as a finite set of possible values.
- $C = \{c_1, \dots, c_p\}$ is a set of constraints, where a constraint c_i is a function $d_{i_1} \times \dots \times d_{i_l} \rightarrow \{0, 1\}$ that returns 1 if the value combination is allowed and 0 if it is not.
- $R = \{r_1, \dots, r_o\}$ is a set of relations, where a relation r_i is a function $d_{i_1} \times \dots \times d_{i_l} \rightarrow \mathbb{R}$ giving the utility of choosing each combination of values.
- R_i is the subset of R that gives the relations associated with agent A_i .

The solution of an MCOP is an assignment of values to all variables that satisfies all constraints and maximizes the sum of agent utilities as expressed by their relations. Note that variables, domains and constraints are common and agreed upon knowledge among the agents. On the other hand, relations are specified by the individual agents, and they do not necessarily have to report them correctly.

An example of an MCOP problem is allocating capacity in a public network, for example a train or pipeline network. The network is a graph of connections, and only one agent can use any one connection at a given time. This can be represented by having one variable per link and time interval whose domain ranges over the set of agents. Constraints would enforce for example that successive links and times are assigned to the same agent.

Agents serve customers' transportation demands with different efficiency by using different combinations of links. Thus, each agent has utilities for being able to use certain combinations of links, and reports these as relations. Agents want to find a combined assignment that maximizes the sum of their utilities. Such combinatorial optimization is NP-complete and thus can be solved exactly only for small problems. For large instances, in many cases only *local search* methods can be implemented. They can provide no optimality guarantees, but with high probability will find a solution that is very close to optimal.

Two additional considerations need to be addressed due to the multi-agent setting: individual rationality and incentive-compatibility. We say that a mechanism is *individually rational* if and only if it is in the best interest of each agent to participate in the mechanism, i.e if the expected utility that each agent gets when it participates in the mechanism is at least as high as if it did not. This is important because otherwise, agents may choose not to participate in the mechanism.

We say that an optimization mechanism is *incentive-compatible* if and only if each agent maximizes its expected utility when the protocol finds the truly optimal solution. Depending on the protocol, incentive-compatibility often means that each agent is reporting its relations truthfully, thus one also speaks of *truthful* mechanisms. Clearly, incentive-compatibility is important to obtain a meaningful solution to the MCOP. It is often achieved through tax or auction mechanisms such as the VCG mechanism ([Vickrey, 1961; Clarke, 1971; Groves, 1973]).

The seminal work of Ephrati and Rosenschein [1991] was the first to propose applying VCG mechanisms to agent coordination. For constraint optimization, game theory has shown that the only practical mechanism for incentive-compatibility in MCOP is of the form of a VCG mechanism ([Green and Laffont, 1979]). However, it has also been shown that VCG mechanisms require finding the provably optimal solution ([Nisan and Ronen, 2000]). Many practical settings of optimization problems are too large for complete optimization algorithms. We thus introduce a weaker concept of *bounded-rational incentive-compatibility* where manipulation is hard through computational complexity. The uncertainty created by randomized local search makes it computationally intractable to evaluate the outcome of an untruthful behavior, thus rendering it uninteresting to agents.

This paper is structured as follows: we first present the local search framework for solving MCOP and define bounded-rational incentive-compatibility. We show how incentive-compatibility can be achieved in each local search step, and then address the key issue of avoiding speculation through the succession of local search steps. We report on experimental results on randomly generated network resource allocation problems that show their performance.

2 Assumptions and Definitions

2.1 Local search framework

Since finding the optimal solution for MCOP is computationally infeasible (NP-complete), in this paper we use local search algorithms (also called *neighborhood search*) to find good, but not necessarily optimal solutions at a reasonable computational cost. Local search is widely used for solving large optimization problems. It has been particularly well studied for satisfiability problems, based on the GSAT procedure ([Selman *et al.*, 1992]). The local search framework we assume in this paper is given in Algorithm 1.

Algorithm 1 Local search algorithm for MCOP

```

procedure LocalSearch( $A, X, D, C$ )
   $\underline{v} \leftarrow \text{SelectInitialSolution}(X, D, C)$ 
  for  $i \leftarrow 1 : m$  do
     $R_i \leftarrow \text{AskRelations}(a_i, \underline{v})$ 
  end for
  repeat
     $\underline{v}^{old} \leftarrow \underline{v}$ 
     $N \leftarrow \text{ChooseNeighbours}(\underline{v}^{old}, X, D, C)$ 
     $(\underline{v}, \text{pay}) \leftarrow \text{LocalChoice}(N, R)$ 
    agents make/receive payments according to  $\text{pay}$ 
  until termination condition met
  return  $\underline{v}$ 
end procedure

```

The algorithm manipulates a complete assignment of values to all variables, represented as a vector $\underline{v}$. It is initially set by function *SelectInitialSolution* to an assignment that satisfies all constraints and could be random. The algorithm then asks each agent to state its set of relations R_i , where the utilities should be relative to the initial assignment $\underline{v}$ such that the

utility of all relations for the assignment $\underline{v}$ is zero. This fixes the otherwise open utility of the initial assignment which has no influence on the optimization result.

Search then proceeds iteratively by local improvements. Function *ChooseNeighbours* provides a set of candidate assignments that are close to the current one and could possibly improve it. In our implementation, they are generated by randomly selecting a variable $x_i \in X$ and generating all assignments that are equal to $\underline{v}^{old}$ but assign to x_i different values in the domain of x_i that are consistent with the rest of $\underline{v}^{old}$ and the constraints in C .

In the second step of the iteration, the assignment $\underline{v}$ is updated using the function *LocalChoice*. It chooses a new assignment to optimize the combined utility according to the relations in R . It also computes a vector of payments pay that agents must make or receive in the third step of the iteration. The payments sum up to zero and the way they are derived is described in detail in Section 3.

The iteration continues until a termination condition is met, for example when there is no further improvement in the utility of all agents for some number of steps. To avoid getting stuck in local optima, the performance of a local search procedure is significantly improved by randomization ([Kirkpatrick *et al.*, 1983; Selman *et al.*, 1994]). This means that occasionally *LocalChoice* chooses a value that may even decrease agent utility.

2.2 Bounded-rational incentive-compatibility

The local search procedure can only work correctly if agents accurately report their utilities R . Using side payments, we can create an *incentive-compatible* mechanism where agents are motivated to truthfully report these valuations. Well-known results in game theory ([Green and Laffont, 1979]) have shown that all mechanisms for MCOP that are incentive-compatible, individually rational and select the optimal solution must be a kind of VCG mechanism. Furthermore, Nisan and Ronen [2000] have shown that a VCG mechanism requires a provably optimal solution. Thus, there is no mechanism that makes local search incentive-compatible while maintaining individual rationality.

We thus replace incentive-compatibility with a weaker notion, called *bounded-rational incentive-compatibility* that uses computational complexity to rule out manipulation:

Definition 2.1 (*Bounded-rational agent*): an agent is called *bounded rational* if it can examine at most C states of the local search before declaring its utilities R_i .

Definition 2.2 Let p_t be a bound on the probability that a *bounded rational agent* can predict whether it has an expected utility gain from an untruthful utility declaration. A mechanism is *bounded rational incentive compatible* if by varying a parameter, p_t can be made arbitrarily close to 0.

Earlier work, such as ([Conitzer and Sandholm, 2003]), has proposed using NP-hardness as a protection against manipulation. However, our definition goes further as it requires *in almost all cases*, manipulation requires an amount of computation that is beyond the means of a bounded-rational agent. Any real computational agent is bounded rational for a sufficiently high C .

3 Local choice step

We now consider incentive-compatibility and randomization of a single step in the local search, carried out by the function *LocalChoice*. We consider that local changes are made to an individual variable x only, but it is straightforward to generalize the mechanism to other neighbourhoods.

We assume that the number of possible alternatives is sufficiently small so that *LocalChoice* can apply a systematic and complete optimization procedure. We call v_c the current assignment to x , and let v_S^* be the value that would be optimal for the set of agents S , i.e. would most improve the sum of their utilities.

3.1 Incentive-compatibility

As the local choice depends on the relations declared by the individual agents, agents would normally have an incentive to report excessive utility gains for their preferred changes so that they impose them over those preferred by others. Thus, the optimal solution according to the declarations of a set of agents A , which we call $\overline{v_A}$, could be different from v_A^* . We counter this tendency by imposing side payments depending on the utility declarations and the change chosen by the mechanism.

As already mentioned earlier, the only side payments that achieve incentive-compatibility, individual rationality and choose the optimal solution are VCG payments. For an agent a_i , the VCG payment is the “damage” it does the others, i.e. the decrease in utility gain its presence causes to the remaining agents:

$$\begin{aligned} VCGtax(a_i) &= \sum_{r \in R \setminus r_i} [(r(\overline{v_{A \setminus a_i}}) - r(v_c)) - (r(\overline{v_A}) - r(v_c))] \\ &= \sum_{r \in R \setminus r_i} [r(\overline{v_{A \setminus a_i}}) - r(\overline{v_A})] \end{aligned}$$

Note that since $\overline{v_{A \setminus a_i}}$ is optimized for $A \setminus a_i$, the sum of its utilities for these agents will always be at least as large as that for $\overline{v_A}$ and thus the *VCGtax* is never negative. Thus, the payments of all agents together leave a positive budget surplus.

3.2 Randomization

Randomization has first been proposed as simulated annealing ([Kirkpatrick *et al.*, 1983]), a technique inspired from the cooling of spin glasses. More recently, it has been studied more systematically, particularly in the context of satisfiability problems. For example, the GSAT algorithm ([Selman *et al.*, 1992]) has been turned into the GWSAT algorithm ([Selman *et al.*, 1994]) by adding a random walk strategy: with some probability, the strategy forces an improvement in a clause by ignoring the other clauses that would also be affected. It has been shown that this randomization has the effect that the algorithm eventually finds the optimal solution with high probability.

The random walk steps in GWSAT leave certain randomly chosen constraints out of the local choice steps. We adopt a similar scheme where we randomly select a set of relations to

be left out of the optimization at a local choice step. It turns out that a good way to select these relations is to take all the relations belonging to a randomly selected agent a_e . As we show below, this way of randomizing allows us to simultaneously guarantee budget-balance of the VCG tax mechanism.

3.3 Payment budget balance and individual rationality

One problem with the VCG mechanism is that agents generate a surplus of taxes that cannot be returned to them without violating the incentive-compatibility properties. This not only reduces their net utility gain, but also creates incentives for whatever third party receives this gain.

The randomization allows us to make the VCG tax scheme budget balanced by simply paying the payment surplus to the agent a_e that was excluded from the optimization step. Each agent a_i other than a_e pays to a_e the following tax:

$$VCGtax_{-e}(a_i) = \sum_{r \in R \setminus (r_i \cup r_e)} [r(\overline{v_{A \setminus (a_i \cup a_e)}}) - r(\overline{v_{A \setminus a_e}})]$$

and the mechanism chooses $\overline{v_{A \setminus a_e}}$ to implement. This can be seen as compensating the agent for the loss of utility it is likely to incur as a consequence of having been left out of the optimization, and does not affect the incentive-compatibility properties:

- for agents other than a_e , it is still best to report their utilities truthfully since they follow a VCG mechanism in a world where a_e does not exist.
- for a_e , its declarations have no effect on the outcome or payments so any declaration is equally good. However, it does not know in advance that it will be excluded, so it still has an interest to make a truthful declaration.

This mechanism is similar to the proposal in [Ephrati and Rosenschein, 1991], who proposed giving the surplus to agents that have no interest in the variable being considered. We call such agents *uninterested* agents. The mechanism proposed here applies even when no uninterested agent exists. When there are uninterested agents, optimization can be improved by selecting these to be chosen as excluded agents. More details on the mechanism can be found in [Faltings, 2004].

In certain cases the sum of the taxes could be less than the utility loss of the excluded agent, and thus it would not be individually rational for the agent to participate. In fact, no matter what payment scheme is used, whenever the local search step leads to a reduction in total agent utility, there must be at least one agent for which individual rationality is violated. Any randomized local search algorithm will occasionally make such moves, for otherwise it would be susceptible to getting stuck in local optima. Thus, no scheme can guarantee individual rationality at every randomized local choice step.

As the algorithm on the whole improves utility for the community of agents, this does not mean that the local search process as a whole is not individually rational. No agent is systematically disadvantaged by the randomization, and so in expectation the scheme is individually rational for all agents.

This is confirmed in our simulations, where individual rationality was always satisfied for all agents in all runs.

4 Sequences of local choices

A local search algorithm is in general incomplete and not guaranteed to find a particular optimal solution. Thus, as pointed out by [Nisan and Ronen, 2000], non-truthful declarations can drive the local search algorithm to a solution that gives a manipulating agent (MA) a better utility than the truthful declaration. However, effectively using such manipulation requires that the MA is capable of correctly predicting the effect of a non-truthful utility declaration on the outcome, and compare it against the utility loss it incurs by carrying out the manipulation in one or several local choice steps. We now show that in a randomized local search algorithm and a sufficiently large problem, with high probability (arbitrarily close to 1), such prediction would require an amount of computation that is beyond the capabilities of a bounded-rational agent.

To obtain a worst-case result, we assume that a manipulating agent (MA) has complete and accurate knowledge of the relations declared by all other agents. Furthermore, we assume that it has access to an oracle that provides it with the most promising manipulation.

The remaining task of the MA is then to show that the manipulation actually produces a better utility than truthful behavior. As the local search algorithm is randomized, the MA can only predict an *expected* utility, obtained by considering the probability of certain states and the utilities that the agent would obtain in each of them. The key idea of our argument is that with high probability, the number of states that need to be considered in this calculation will grow exponentially with the size of the problem. Thus, for a certain problem size it will exceed the computational capacity of a bounded rational agent.

To show this result, we first argue that the MA has to examine a significant fraction of the probability mass of the states to ensure success of the manipulation. This fraction depends on two factors:

- the utility distribution of the problem: if only few states give a significant utility, or if there are strong symmetries so that the state space can be factored, it could be sufficient to sample only a small subset of the states, and
- the desired confidence of the prediction: since a manipulation will mean a certain utility loss to the agent in the search step where it is applied, the manipulation needs to succeed with a certain minimal probability in order to give an increase in expected utility. Depending on the utility distribution, this translates to a certain fraction of the state probability mass that will need to be examined.

Note that both parameters are independent of the size of the search space. Thus, we can assume that the MA will have to examine a minimal number of states that corresponds to some fraction α of the probability mass of the entire search space.

Next, we show that with high probability, this probability mass is distributed over a number of states that grows exponentially with problem size.

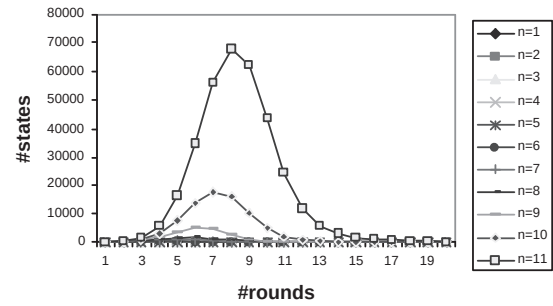


Figure 1: New states discovered in successive cycles of a simulation of local search, for several problem sizes.

The local search algorithm contains several possibilities for randomization that can make its outcome hard to predict:

- random choice of neighbourhood,
- random choice of excluded agent a_e ,
- random choice of equivalent local choices.

To make the effect of randomization easy to analyze, we consider only randomizations whose outcomes can be regarded as independent. This is not the case of the random choice of neighbourhood, as it will often be the case that choosing n_1 and then n_2 will lead to the same states as choosing n_2 and then n_1 , thus cancelling the randomization effect. Also, local search has to ensure that all neighbourhoods are considered, placing limits on the amount of randomization that can be allowed. Furthermore, a manipulating agent could make declarations to create symmetries that makes the randomization ineffective.

Fortunately, for the choice of excluded agent as well as the choice among equivalent solutions, it does appear reasonable to assume independence of subsequent random choices. Furthermore, since the manipulating agent's relations may be excluded, it cannot render the randomization invalid through its own declarations.

To find the global optimum, a local search algorithm has to be able to reach the entire search space. However, eventually it will come close to the global optimum and then remain within a much smaller subspace of nearly optimal states. While it is possible to give a theoretical analysis that shows that with arbitrarily high probability, the probability of reaching any given state is bounded by an exponentially decreasing value, such an analysis requires many independence assumptions that may not hold in practice. Here, we present the following experimental measurements, obtained on the experimental scenario given in the next section.

Figure 1 shows the number of new states discovered in successive cycles of a simulated randomized local search. It initially grows exponentially, but eventually search stabilizes on certain optimal outcomes and thus fails to discover new states. Importantly, however, the total number of states discovered, shown in Figure 2, still grows exponentially with problem size: in this example, it multiplies with a factor of about 3 whenever the size increases by 1. Thus, the total

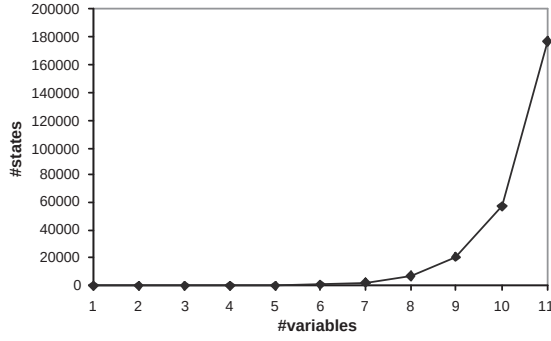


Figure 2: Growth of the total number of states involved in a local search simulation as a function of the problem size.

number of states has exponential growth with problem size, even though it does not reach the total state space because of the convergence of the algorithm.

Another aspect that needs to be shown is that the MA cannot limit its consideration to only certain states in this space, i.e. that the probability mass is distributed over a large subset of the states. We show this by considering the probabilities of the resulting states at each randomized step. Let p_m denote the probability that at a random branch, each of the branches is taken with probability at most $1/m$. We have measured p_m experimentally (see later section) and have obtained for example for $p_4 \simeq 0.908$, showing that the search process has a significant branching factor. Thus, with high probability the probability mass is distributed among a large number of states.

A theoretical analysis with independence assumptions on this basis gives for example that in a local search with 1000 variables, a stopping probability of 0.0001 (expected number of cycles = 10^4) and $\alpha = 0.1$, the probability that an agent would have to examine less than 10^{54} states is bounded by $p_t < 10^{-9}$. This is certainly well beyond the capability of any computational agent today.

While we have so far only analyzed relatively simple models, it seems clear that in general the probability mass is very likely to be spread among a large set of states as the size of the problem increases, and thus the method will be bounded-rational incentive-compatible with the parameter being the problem size.

5 Experimental results

We have implemented the mechanism we described for a network resource allocation problem. It consists of allocating tracks in the train network shown in Figure 3 to different operators. To avoid collisions, each arc in the graph can only be allocated to one operator who can then run trains on it. At the same time, there are certain demands for transporting goods. For each demand, 3 feasible routes are computed and it can take any of these routes. This is modelled as an MCOP having a variable for each task whose domain is the agent and route assigned to it. For example, if task 3 (*London, Manchester*) is assigned to agent a_1 on the path

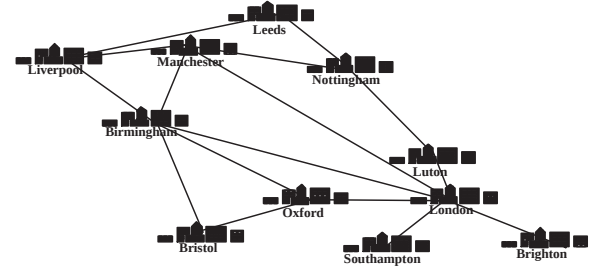


Figure 3: The transportation network used in the experiments

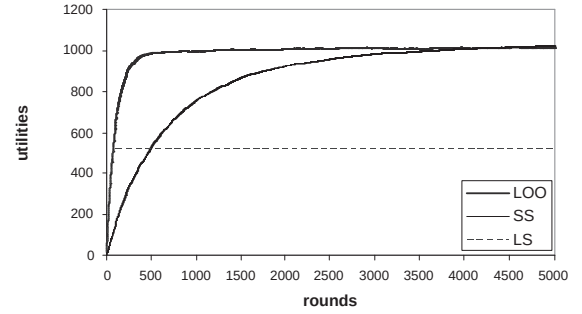


Figure 4: Average utility gain of different local search algorithms as a function of the number of steps.

(*London → Birmingham → Manchester*), the corresponding variable x_3 is assigned the value ($a_1, \text{London} \rightarrow \text{Birmingham} \rightarrow \text{Manchester}$).

The network capacity is modelled by binary constraints between any pair of tasks whose routes share at least one arc that rule out assigning it to such overlapping routes but different agents. Each operator has a different and randomly generated profit margin for being assigned a demand/route combination, and declares these through its relations. We randomly generated tasks and routes and simulated the problem starting from a situation where no task is assigned to any agent.

We used the experiments to observe three properties: efficiency, branching probabilities and individual rationality. First, we want to show the efficiency of the randomized protocol with respect to straight hill-climbing to show that it indeed escapes from local minima. Figure 4 compares the performance of local search with randomization (LOO, shown by the thick line) with stochastic search (SS, thin line) and strict hill-climbing (LS, dashed line) on 100 randomly generated problem instances. We see that local search gets stuck in a local optimum and only reaches about half the total utility that the randomized search gets. Thus, the scheme seems to be effective at avoiding local minima.

Second, we are interested in the average probability p_m

m	1	2	3	4	5	6	7	≥ 8
$r(m)$	2	21	69	101	194	120	160	333

Table 1: Computational results for p_m

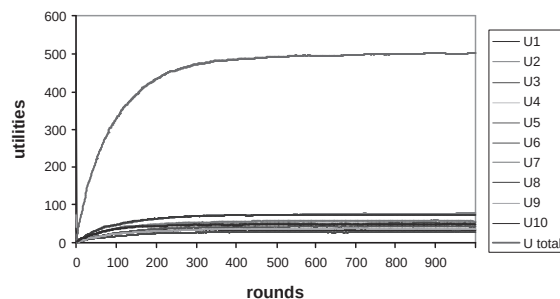


Figure 5: Agents' utilities during search

that a Localchoice generates no branch with probability mass larger than $1/m$. We thus took a histogram over 1000 iterations of the number $r(m)$ that this condition is satisfied for m . Table 1 shows the result for $m \leq 10$. From the table, we can estimate for example $p_4 \simeq 0.908$.

Third, we are interested in the actual utilities for each agent, and in particular whether we can guarantee individual rationality. Figure 5 shows the utilities of agents during the local search process. In this experiment, we run the local search on random problems with 10 agents and 100 tasks for 1000 rounds. It can be seen that the agents' net utilities are positive and stable when the number of rounds increases.

6 Conclusions

Finding an optimal coordination between multiple self-interested agents is a problem that occurs frequently in practice. Incentive-compatibility is an essential property to ensure meaningful results of such an optimization. Previous work ([Ephrati and Rosenschein, 1991]) has shown the applicability of VCG mechanisms to such problems. However, it requires provably optimal solutions to the NP-hard optimization problem and thus cannot be applied to large problems.

Our work is based on the observation that in real life, the potential for manipulation is limited by uncertainty and risk. This uncertainty makes it difficult for a manipulator to predict the consequences of his manipulation and thus makes attempts at manipulating it uninteresting. Similar uncertainty exists in local search algorithms where randomization is necessary to escape local optima. We have analyzed a scheme for randomization and shown that in sufficiently large problems, it creates a large amount of uncertainty so that simulating a sufficient part of the possible outcomes quickly surpasses the computational capacity of any real computational agent. Problems that are too small for this result to apply can likely be addressed by VCG mechanisms with complete optimization.

Note that the techniques in this paper all rely on local choices and payments and are thus very suitable for asynchronous, distributed implementation.

An important limitation of the MCOP formulation is that agents cannot claim private constraints. Thus, it is impossible to model trading scenarios where an agent has control over certain variables, for example the ownership of a good.

This limitation is important because it ensures that the VCG payments never leave a deficit that would have to be covered by the excluded agent.

While no randomized local search algorithm can guarantee individual rationality, we found that it seems to be satisfied with high probability. It would be interesting to analyze the individual rationality properties of local search schemes to obtain probabilistic guarantees similar to those for non-manipulability.

The most important weakness of the current scheme is that the parameter that needs to be varied to guaranteed bounded-rational incentive-compatibility is the size of the problem. It would be much better if we had a mechanism that could guarantee high manipulation complexity even for small problems through suitable randomization of this choice, similarly to certain cryptographic hash functions.

Acknowledgements

This work has been funded by the Swiss National Science Foundation under contract No. 200020-103421/1.

References

- [Clarke, 1971] E.H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.
- [Conitzer and Sandholm, 2003] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. *Proceedings of IJCAI-03*, pp. 781–788, 2003.
- [Ephrati and Rosenschein, 1991] E. Ephrati and J. S. Rosenschein. The clark tax as a consensus mechanism among automated agents. In *Proceedings of AAAI-91*, pp. 173–178, San Jose, California, July 1991.
- [Faltings, 2004] Boi Faltings. A Budget-balanced, Incentive-compatible Scheme for Social Choice, *Agent-Mediated Electronic Commerce VI*, Springer LNAI 3435, 2004.
- [Green and Laffont, 1979] J. Green and J.J. Laffont. Incentives in public decision making. *Studies in Public Economics*, 1, 1979.
- [Groves, 1973] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–31, 1973.
- [Kirkpatrick et al., 1983] S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [Nisan and Ronen, 2000] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *Proceedings of AMEC-2000*, pp. 242–252, 2000.
- [Selman et al., 1992] B. Selman, H. Levesque, and D. Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of AAAI-92*, pp. 440–446, 1992.
- [Selman et al., 1994] B. Selman, H. Kautz, and B. Cohen. Noise strategies for improving local search. In *Proceedings of AAAI-94*, pp. 337–343, 1994.
- [Vickrey, 1961] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16(2):8–37, 1961.

More on the Power of Demand Queries in Combinatorial Auctions: Learning Atomic Languages and Handling Incentives

Sébastien Lahaie Florin Constantin David C. Parkes

Division of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138
{slahaie, florin, parkes}@eecs.harvard.edu

Abstract

Query learning models from computational learning theory (CLT) can be adopted to perform elicitation in combinatorial auctions. Indeed, a recent elicitation framework demonstrated that the *equivalence* queries of CLT can be usefully simulated with price-based demand queries. In this paper, we validate the flexibility of this framework by defining a learning algorithm for atomic bidding languages, a class that includes XOR and OR. We also handle incentives, characterizing the communication requirements of the Vickrey-Clarke-Groves outcome rule. This motivates an extension to the earlier learning framework that brings truthful responses to queries into an equilibrium.

1 Introduction

There has been a burst of recent attention in the problem of preference elicitation in combinatorial auctions (CAs), which are auctions in which agents can express values on bundles of items [Blum *et al.*, 2004; Lahaie and Parkes, 2004; Conen and Sandholm, 2001; Santi *et al.*, 2004; Parkes, 2002]. Indeed, elicitation is today recognized to be just as important a computational problem (and perhaps more so) as that of winner-determination [Nisan and Segal, 2004]. CAs can potentially require agents (e.g. people, firms, automated bidding agents) to value an exponentially large number of different bundles of indivisible goods, when determining the value on even a single bundle can be hard [Sandholm, 1993].

In *query-based elicitation* [Parkes, 2002; Conen and Sandholm, 2001], agents must be able to respond to simple queries, such as identifying a preferred bundle at given prices, or providing the value of a specified bundle. Many of these methods are closely related to the “exact learning with queries” model from computational learning theory (CLT) [Blum *et al.*, 2004]. In fact, these are typically learning methods in their own right because they elicit valuations entirely and exactly. One exception is Lahaie and Parkes [2004], who give a preference elicitation scheme that avoids complete learning when possible. The main contribution in that paper is to explain how to simulate any learning algorithm with *membership* and *equivalence* queries as an elicitation algorithm with *value* and *demand* queries.

We extend this earlier work by providing a preference elicitation scheme for a broad class of languages, called *atomic languages*, which includes XOR and OR as special cases (see Nisan [2000] for a formal study of bidding languages). In addition to validating the flexibility of the Lahaie and Parkes [2004] framework, our new algorithm demonstrates the power of demand queries. For instance, we know of no learning algorithm for OR using just value and equivalence queries.

Our main contribution is to characterize the communication requirements of the Vickrey-Clarke-Groves (VCG) outcome rule [Jackson, 2000]. Already known to be sufficient for determining the VCG outcome [Mishra and Parkes, 2004], we prove here that any elicitation protocol for the VCG mechanism *necessarily* also determines a set of “universal competitive equilibrium” (UCE) prices. This result broadens our understanding of demand queries, demonstrating that if the goal is to verify a VCG outcome it is necessary to verify an efficient allocation and set of UCE prices, which can be done through a simple extension of demand queries to *universal* demand queries.¹ We demonstrate how to extend the existing learning framework to terminate with UCE prices and an efficient allocation.

With n agents, a naïve way to handle incentives is to simply run the preference elicitation algorithm, and then run it again n more times with each agent removed. This yields enough information to derive VCG payments. However, the UCE-based characterization motivates a design for an extension of the learning-based framework that we call *Learner Extend and Adjust* (LeEA), that obtains VCG payments in a single run of the algorithm.

Related Work. Nisan and Segal [2004] characterize the minimal communication requirements of implementing an efficient allocation, but do not consider incentives. The large literature on ascending-price Vickrey auctions (see Parkes [2004] for a survey) is largely motivated by issues of costly elicitation, and recent auctions are designed to terminate with UCE prices [Mishra and Parkes, 2004]. Similarly,

¹This is not to say that elicitation algorithms for VCG based on demand queries are necessarily efficient. Indeed, in some cases demand-query based algorithms are known to be exponentially-inefficient [Nisan and Segal, 2004]. However, it is at least suggestive that demand queries are powerful in general.

Conen and Sandholm [2002] had previously considered elicitation methods that terminate with the VCG outcome. What is new in our work is the proof that UCE prices are *necessary* (as well as sufficient) for the VCG outcome, together with the careful integration of this methodology into elicitation methods with polynomial query complexity.

2 Preliminaries

The purpose of a CA is to allocate a set G of m distinct and indivisible goods among a set N of n agents, each with a valuation $v_i : 2^G \rightarrow \mathbb{R}$. Let Γ be the set of possible allocations, in which no good is given to more than one agent. We aim for an efficient allocation $S^* = (S_1^*, \dots, S_n^*)$, namely an allocation that maximizes total value $\sum_{i=1}^n v_i(S_i^*)$. By definition, agent valuations satisfy the property of *no externalities*, meaning that an agent only cares about its own bundle, and not those allocated to other agents. Valuations also satisfy *free-disposal*, meaning that $v_i(S) \leq v_i(T)$ if $S \subseteq T$, are *normalized*: $v_i(\emptyset) = 0$, and *bounded*, so that there is a constant $K > 0$ (known to the center) such that $v_i(S) < K$ for all $S \subseteq G$ and all $i \in N$. Agents have *quasi-linear* utility functions, so that agent i 's utility for a bundle S at prices p is $u_i(S, p) = v_i(S) - p(S)$.

2.1 Bidding Languages

A bidding language is used to encode and communicate real-valued functions over bundles; this may for instance be an agent's entire valuation function, or an underestimate of its true valuation function (as in bids in ascending-price CAs, hence the name "bidding language"). In this work, the center also uses bidding languages to quote prices to the agents.

Formally, a bidding language consists of syntax and semantics that allow one to encode value information and interpret these encodings. We consider the class of *atomic languages*. An atomic representation is a pair $(\mathcal{B}, w)$ where $\mathcal{B} \subseteq 2^G$ is a set of bundles (the atomic bundles), and $w : \mathcal{B} \rightarrow \mathbb{R}_{\geq 0}$ is a real-valued function over these bundles (the values of the atomic bundles). An alternate useful syntax is a list of *atomic bids*, where an atomic bid is a bundle-value pair (S, x) , and x is to be interpreted as the value of bundle $S \subseteq G$. The semantics of an atomic language L are defined through the evaluation function $\phi_L(\cdot; \mathcal{B}, w)$ that extends w to the set of all bundles, so that the value of a bundle S is $v(S) = \phi_L(S; \mathcal{B}, w)$. A well-formed atomic language evaluation function satisfies the following conditions:

- L1. $\phi_L(B; \mathcal{B}, w) = w(B)$ for all $B \in \mathcal{B}$.
- L2. $\phi_L(S; \mathcal{B}, w) = \phi_L(S; \mathcal{B}|_S, w)$ for all $S \subseteq G$, where $\mathcal{B}|_S = \{T \in \mathcal{B} \mid T \subseteq S\}$.
- L3. $\phi_L(S; \mathcal{B} \cup \mathcal{B}', w) \geq \phi_L(S; \mathcal{B}, w)$, for all $\mathcal{B}, \mathcal{B}' \subseteq 2^G$ and all $S \subseteq G$.

As an important example, consider the family of languages $\{L_k\}_{k=1}^m$. The evaluation function for language L_k is $\phi_k(S; \mathcal{B}, w) = \max_{S \in \mathcal{D}_k(S; \mathcal{B})} \sum_{B \in S} w(B)$, where $\mathcal{D}_k(S; \mathcal{B})$ defines the set of all $S \subseteq 2^{\mathcal{B}}$ that satisfy (i) $|S| \leq k$, (ii) $\cup_{B \in S} B \subseteq S$, (iii) $B_1 \cap B_2 = \emptyset$ for $B_1, B_2 \in S$ s.t. $B_1 \neq B_2$. We call elements of $\mathcal{D}_k(S; \mathcal{B})$ *decompositions* of S into atomic bundles from $\mathcal{B}$. It is easy to see that ϕ_k

satisfies conditions L1-L3 above. As special cases we have $L_1 = \text{XOR}$ and $L_m = \text{OR}$.

A bidding language is *expressive* for CAs if it can encode all valuation functions $v : 2^G \rightarrow \mathbb{R}_{\geq 0}$ that satisfy free disposal. It is not hard to see that L_1 (XOR) is expressive, but that L_k for $k > 1$ are not (consider for example the valuation that places a value of 1 on all bundles). Representation $(\mathcal{B}, w)$ in a language L of valuation v is minimal if $|\mathcal{B}|$ is minimized over all valid L representations of v . In what follows, $\text{size}_L(v)$ denotes the size of a minimal L representation of function v , when such a representation exists.

2.2 Queries

Agents are modeled as "black-boxes" that can respond to queries. A response to a query provides partial information about an agent's valuation function. We adopt queries that are more or less natural in economic settings:

Value query. A value query $\text{VALUE}(S)$ to agent i on bundle S asks the agent to report its *exact* value $v_i(S)$.

Demand query. A demand query $\text{DEMAND}(S, p)$ presents an agent i with a bundle S and prices p over all bundles (in some bidding language). The agent replies YES if bundle S is a best-response at prices p , meaning that $v_i(S) - p(S) = \max_{T \subseteq G} [v_i(T) - p(T)]$. Otherwise the agent replies with a bundle S' that makes it strictly better off than S at prices p , i.e. $v_i(S') - p(S') > v_i(S) - p(S)$.

Universal demand query. A universal demand query $\text{UNIDEMAND}(S_1, \dots, S_n, p)$ presents an agent i with prices p over all bundles together with a set of n bundles $S_1, \dots, S_n$ (not necessarily distinct). The agent replies YES if *every* bundle presented is a best-response to prices p . Otherwise, the agent responds with an index j and a bundle S' such that $v_i(S') - p_i(S') > v_i(S_j) - p_i(S_j)$.

Note that $\emptyset$ may be a valid response to a demand query. Also note that prices are general functions $p : 2^G \rightarrow \mathbb{R}_{\geq 0}$, and may be nonlinear (bundles are priced, not just items alone) and non-anonymous (different agents may face a different price for the same bundle).

2.3 CE and UCE Prices

CAs, and the preference elicitation scheme we consider here, generally operate by converging to a *competitive equilibrium*. This ensures that the final allocation is indeed efficient.

Definition 1 A competitive equilibrium *among agents* N is an allocation $S^* = (S_1^*, \dots, S_n^*)$ together with prices p that satisfy: (1) $S_i^* \in \arg\max_{S \subseteq G} [v_i(S) - p_i(S)]$ for $i = 1, \dots, n$ and (2) $S^* \in \arg\max_{S \in \Gamma} \sum_{i=1}^n p_i(S_i)$.

If (S^*, p) constitutes a CE, we call p the *CE prices* and say that prices p *support* allocation S^* . For our results later on incentives, we also need the following concept:

Definition 2 A universal competitive equilibrium is an allocation-price pair (S^*, p) that constitutes a competitive equilibrium among agents N , and such that (S_{-i}^*, p) constitutes a competitive equilibrium among agents $N \setminus \{i\}$, for some efficient allocation S_{-i}^* of items G among agents $N \setminus \{i\}$, for all $i \in N$.

In fact, it is not hard to show that prices p that support some efficient allocation S_{-i}^* among agents $N \setminus \{i\}$ for all $i \in N$ also support *all* such efficient allocations.

3 A Learning-Based Architecture for Elicitation

The learning framework of Lahaie and Parkes [2004] (that we call ‘Learner’) converts individual exact learning algorithms into preference elicitation algorithms. We will demonstrate how to instantiate it here for the class of atomic languages. The goal of a learning algorithm for our purposes is to exactly determine an unknown *target* valuation v represented in a given bidding language L in a number of queries that scales polynomially with $size_L(v)$ and m . Let $\mathcal{V}$ be the space of possible valuations, in our case all bounded and normalized valuations that satisfy free-disposal and have no externalities.

Exact learning in the computational learning theory literature typically uses *membership* and *equivalence* queries. A membership query in our domain is just a value query. Learning algorithms maintain a *manifest* valuation $\tilde{v}$, which is their current estimate of the target function. On an equivalence query we present $\tilde{v}$ to the agent; it replies YES if $\tilde{v} = v$, and otherwise replies with some *counterexample* S such that $\tilde{v}(S) \neq v(S)$.

Lahaie and Parkes [2004] note that for a multi-agent scenario, learning algorithms can be run in parallel for each agent while they perform value queries. Demand queries play a key role in coordinating learning across agents. When all agents require the response to an equivalence query, one can instead compute an efficient allocation $\tilde{S}^*$ and supporting CE prices $\tilde{p}$ with respect to the current manifests, and present these to the agents as demand queries. Call these the *manifest allocation* and the *manifest prices*. If all agents reply YES, we have reached a CE and $\tilde{S}^*$ is already efficient; we need not learn any more information. Otherwise, the responses to the demand queries are in fact counterexamples that can be returned to the learning algorithms as responses to their equivalence queries.

In the case of a specific bidding language, such as an atomic language, we need to provide means to: (1) learn a target valuation in the given bidding language; (2) compute an efficient allocation with respect to manifests (i.e. solve winner-determination); (3) compute supporting CE prices.

3.1 Learning Atomic Languages

We now describe a learning algorithm with value and demand queries for atomic languages.² The manifest valuation $\tilde{v}$ is stored as pair $(\tilde{\mathcal{B}}, \tilde{w})$. Let $(\mathcal{B}, w)$ be the minimal L representation of target valuation v . The manifest representation will always satisfy (1) $\tilde{\mathcal{B}} \subseteq \mathcal{B}$, and (2) $\tilde{w}(B) = w(B)$ for $B \in \tilde{\mathcal{B}}$. This is true initially because we set $\tilde{\mathcal{B}} = \emptyset$. Throughout the learning process we will ensure that only elements $B \in \mathcal{B}$

²Note that it is perfectly alright for individual learning algorithms to use demand queries in the Learner framework. In simulating these algorithms, we just need to be clear which demand queries can be performed asynchronously and in parallel, and which queries need to be performed on all agents simultaneously.

are ever added to $\tilde{\mathcal{B}}$, and we make value queries on these new additions to appropriately set $\tilde{w}(B)$. From these conditions and L3 we get that $\tilde{v}(S) \leq v(S)$ for all $S \subseteq G$ at all times, and see that $\tilde{v}$ *underestimates* v .

The outer-loop of the learning algorithm issues $\text{DEMAND}(\emptyset, \tilde{v})$, i.e. with prices quoted equal to the current manifest and in the same bidding language. If the response is YES we are done since $0 \leq v(S) - \tilde{v}(S) \leq 0$ for all $S \subseteq G$. Otherwise we obtain a more-preferred bundle S that is a counterexample (this follows from a simple adaptation of Lemma 1 in Lahaie and Parkes [2004]). We then have $0 < v(S) - \tilde{v}(S) = \phi_L(S; \mathcal{B}, w) - \phi_L(S; \tilde{\mathcal{B}}, \tilde{w})$. By property L2 this means there is at least one undiscovered atomic bundle $B \in \mathcal{B}_S \setminus \tilde{\mathcal{B}}_S$. To derive the atomic bundle B from S , we use the subroutine presented as Algorithm 1. Recall that K is an upper-bound on agent values.

Algorithm 1 findNewAtomic(S)

Construct prices p_S where

$$p_S(T) := \begin{cases} K & \text{if } T \not\subseteq S \\ \tilde{v}(T) & \text{otherwise} \end{cases}$$

Issue $\text{DEMAND}(\emptyset, p_S)$

if the agent replies YES **then**

return S

else if the agent replies with bundle R **then**

return findNewAtomic(R)

end if

The prices p_S in this subroutine can be constructed in the atomic language as $(\mathcal{B}_p, \tilde{w}_p)$, as follows. First, initialize $\mathcal{B}_p = \tilde{\mathcal{B}}$ with $\tilde{w}_p = \tilde{w}$. Then, add atomic bids (S, K) and $(\{j\}, K)$ for all $j \notin S$ to complete the representation. The size of this construction is clearly polynomial in $size_L(v)$ and m . Finally, we then update our manifest by adding $(B, \tilde{w}(B))$, where B is the atomic bundle discovered in Algorithm 1 and $\tilde{w}(B)$ is obtained via a value query, and continue. This is correct by property L1. The correctness of the algorithm follows directly from this lemma:

Lemma 1 *On performing query $\text{DEMAND}(\emptyset, p)$ in Algorithm 1, if the reply to the demand query is YES then S is an atomic bid, and otherwise the bundle returned R is a subset of S and also a counterexample to $\tilde{v}$.*

Proof. If the agent replies YES, then it must be that $\tilde{v}(R) = v(R)$ for all $R \subset S$, because $\tilde{v}$ always underestimates v . In particular, we have $\tilde{v}(B) = v(B)$ for each $B \in \mathcal{B}_S$, and every atomic bundle $B \subset S$ has been discovered by the minimality of $(\mathcal{B}, w)$ and condition L2. Then, it must be that $S \in \mathcal{B}$ is a new atomic bid since $\tilde{v}(S) \neq v(S)$. If instead bundle R is returned then $R \subset S$ by the structure of the prices (all other bundles are priced at K). Moreover, as $v(R) > p(R) = \tilde{v}(R)$, we have a new counterexample. $\square$

Note that this subroutine is called at most $size_L(v)$ times since a new atomic bundle is returned each time. Also, observe that Algorithm 1 makes at most m demand queries, because we always recurse on strict subsets of the original argument S and $|S| \leq m$.

Theorem 1 An unknown target valuation v in any atomic language L can be learned with at most $\text{size}_L(v) \cdot (m+1) + 1$ demand queries and at most $\text{size}_L(v)$ value queries.

This algorithm is efficient for atomic languages and provides the first polynomial query learning algorithm for the OR language. OR can be more concise than XOR although it is not always expressive [Nisan, 2000]. We do not know of a learning algorithm for OR with membership and equivalence queries alone, so this result suggests the power of demand queries. In this context, demand queries are used to provide a kind of “focused” equivalence query in which an agent’s new counterexample is restricted to being a subset of the current counterexample.

3.2 Computing CE Prices for L_k

In this section we explain how to compute the manifest allocation and prices for language L_k . It is useful to work with the explicit structure of L_k , although we intend this to be suggestive of an approach that is workable for any atomic language. We first describe a generalized mixed-integer program (MIP) for winner-determination (WD), which will also be relevant for computing CE prices. All mathematical programs defined in this section can be readily solved with MIP solvers such as CPLEX.

Assume agent i ’s manifest is represented in language L_{k_i} . The program description takes five arguments: (1) a set of agents $N' \subseteq N$; (2) a vector of bundles S , with one bundle S_i for each agent $i \in N'$; (3) a vector of sets of atomic bundles $\mathcal{B}$, with one set $\mathcal{B}_i$ for each agent $i \in N'$; (4) values $w_i(B)$ for each agent over all atomic bundles $B \in \mathcal{B}_i$; (5) prices $p_i(B)$ for each agent over all atomic bundles $B \in \mathcal{B}_i$. Let $r_i = |\mathcal{B}_i|$. Let $\Gamma_i \subseteq 2^{\mathcal{B}_i}$ be the possible allocations to agent i (sets of pairwise disjoint atomic bundles with cardinality at most k_i). An allocation here is a vector $\gamma = (\gamma_1, \dots, \gamma_n)$ where $\gamma_i \in \Gamma_i$ for each agent $i \in N$. The generalized formulation $\text{WD}(N', S, \mathcal{B}, w, p)$ is as follows:

$$\begin{aligned} \max_{a_{ij}, e_{\beta_i}, z_i} \quad & \sum_{i \in N'} \left[\sum_{j=1}^{r_i} w_i(B_{ij}) a_{ij} - z_i \right] \\ \text{s.t.} \quad & \sum_{j=1}^{r_i} a_{ij} \leq k_i \quad \forall i \in N' \quad (1) \end{aligned}$$

$$\sum_{i \in N'} \sum_{\{j: B_{ij} \ni g\}} a_{ij} \leq 1 \quad \forall g \in G \quad (2)$$

$$e_{\beta_i} \geq 1 - |\{j : B_{ij} \in \beta_i\}| + \sum_{\{j: B_{ij} \in \beta_i\}} a_{ij} \quad \begin{matrix} \forall \beta_i \in \Gamma_i \\ \forall i \in N' \end{matrix} \quad (3)$$

$$z_i \geq \sum_{\{j: B_{ij} \in \beta_i\}} p_i(B_{ij}) e_{\beta_i} \quad \begin{matrix} \forall \beta_i \in \Gamma_i \\ \forall i \in N' \end{matrix} \quad (4)$$

$$\sum_{\{j: B_{ij} \ni g\}} a_{ij} = 0 \quad \begin{matrix} \forall g \notin S_i \\ \forall i \in N' \end{matrix} \quad (5)$$

$$a_{ij} \in \{0, 1\}, e_{\beta_i} \in \{0, 1\}, z_i \geq 0$$

Let $\vec{\mathcal{B}} = (\vec{\mathcal{B}}_1, \dots, \vec{\mathcal{B}}_n)$, $\vec{G} = (G, \dots, G)$ (with n entries), $\vec{w} = (w_1, \dots, w_n)$ (the values in agent manifests), and

$\vec{0} = (0, \dots, 0)$ (with n entries).³ The usual program to solve winner-determination is $\text{WD}(N, \vec{G}, \vec{\mathcal{B}}, \vec{w}, \vec{0})$.⁴ Let v^* be the value of the efficient allocation obtained through WD. The linear program (LP) to obtain CE prices is as follows.

$$\begin{aligned} \min_{\pi^s, \pi_i, p_i(B_{ij})} \quad & \pi^s \\ \text{s.t.} \quad & \pi^s \geq \sum_{i=1}^n \sum_{\{j: B_{ij} \in \gamma_i\}} p_i(B_{ij}) \quad \forall \gamma \in \Gamma \quad (6) \end{aligned}$$

$$\pi_i \geq \sum_{\{j: B_{ij} \in \beta_i\}} \tilde{w}_i(B_{ij}) - \sum_{\{j: B_{ij} \in \beta'_i\}} p_i(B_{ij}) \quad \begin{matrix} \forall i \in N \\ \forall S \subseteq G \\ \forall \beta_i, \beta'_i \in \mathcal{D}_k(S; \mathcal{B}_i) \end{matrix} \quad (7)$$

$$\pi^s + \sum_{i=1}^n \pi_i = v^* \quad (8)$$

$$\pi^s \geq 0, \pi_i \geq 0, p_i(B_{ij}) \geq 0$$

Variables π_i can be interpreted as the utility to each agent from their allocations at the prices computed, and variable π^s as the revenue to the seller. This formulation has an exponential number of utility-maximization constraints (7) and revenue-maximization constraints (6). To address this we can use *delayed constraint generation* [Bertsimas and Tsitsiklis, 1997]. We only keep a subset of these constraints (initially empty), and obtain specific values for each of the variables in the LP. To check whether any implicit constraints of type (6) are violated, we solve $\text{WD}(N, \vec{G}, \vec{\mathcal{B}}, \vec{p}, \vec{0})$, where $\vec{p}$ is the vector of prices obtained from the LP. If the solution to this is greater than π^s , we add the constraint of type (6) that corresponds to the allocation obtained by this auxiliary integer program.

Similarly, we can run $\text{WD}(\{i\}, \vec{G}, \vec{\mathcal{B}}, \vec{w}, \vec{p})$ for each agent i to see if we need to generate any constraints of type (7), and check whether the result is greater than π_i . Note that each such integer program has an exponential number of constraints of type (3) and (4). To use delayed constraint generation for these, we can run $\text{WD}(\{i\}, (\dots, S_i, \dots), \vec{\mathcal{B}}, \vec{p}, \vec{0})$ as an auxiliary program, where S_i is the solution to the main integer program. If the solution to this program has value greater than z_i , we must generate corresponding constraints of type (3) and (4).⁵

4 Communication Requirements of Implementing the VCG Outcome

In the above discussion, we set aside the issue of incentives in preference elicitation. A reliable elicitation scheme must also induce the agents to truthfully reveal their preferences.

Suppose that instead of implementing the final CE prices

³The function $\vec{0}$ is identically 0 over all bundles.

⁴In this case, constraints (3) are irrelevant because they only serve to activate e_{β_i} if appropriate atomic bundles are selected, to indirectly ensure that z_i is set to the correct price through constraint (4), and the price here is always 0. Similarly constraints (5) are irrelevant because any agent can be allocated any bundle.

⁵Note that this last program is the usual one the proxy would use to determine the value or price of a bundle, given the L_k representation of its agent’s manifest valuation.

obtained by Learner, we provide a payment of $\sum_{i \neq j} v_i(S_i^*)$ to agent j for all $j \in N$, where $(S_1^*, \dots, S_n^*)$ is the efficient allocation (agent values are available via value queries). As Nisan and Segal [2004] point out, this aligns each agent's utility with the overall objective, and truthful revelation becomes an *ex post* Nash equilibrium of the elicitation protocol. However, this scheme is clearly very costly for the center. The center pays the agents! The cheapest payment scheme that aligns the agents' individual incentives with economic efficiency is the VCG payment [Krishna and Perry, 2000].

In this section, we may drop the assumption of *no externalities* so that our results hold with greater generality. Valuations and prices are therefore defined over entire allocations. Let $\mathcal{V}_i$ be the set of possible valuations for agent i . A state is a valuation profile $v \in \mathcal{V} = \mathcal{V}_1 \times \dots \times \mathcal{V}_n$. An outcome in our scenario is an element of $\mathcal{O} = \Gamma \times \mathbb{R}^n$, namely a specification of the allocation and the agents' payments.

The objective is to implement the *Vickrey outcome rule*, which is a correspondence $F = (F_1, F_2) : \mathcal{V} \rightarrow \mathcal{O}$ mapping states $v \in \mathcal{V}$ to pairs $(\hat{\gamma}, q)$ such that $\hat{\gamma}$ is an efficient allocation for profile v and $(q_1, \dots, q_n)$ is the associated vector of VCG payments. Letting $\hat{\gamma}(v) \in \arg \max_{\gamma \in \Gamma} \sum_{i \in N} v_i(\gamma)$ and $\hat{\gamma}_{-j}(v) \in \arg \max_{\gamma \in \Gamma} \sum_{i \neq j} v_i(\gamma)$, the VCG payments (from the agents to the center) are defined as:

$$F_{2j}(v) = \sum_{i \neq j} v_i(\hat{\gamma}_{-j}(v)) - \sum_{i \neq j} v_i(\hat{\gamma}(v)) \quad \text{for } j \in N$$

Mishra and Parkes [2004] have shown that to compute VCG payments it is sufficient to obtain UCE prices. In fact, we show here that it is not only sufficient but necessary that a communication protocol for VCG discover UCE prices.⁶

We consider *nondeterministic* communication protocols. This is the setting in which a center claims the VCG outcome is (γ, q) , and must send messages to each agent to convince them of this outcome. Each agent checks that the message is valid given the semantics of the protocol and its private type, and if all respond YES the protocol has verified the outcome.

Definition 3 [Nisan and Segal, 2004] A nondeterministic communication protocol is a triple $\Pi = \langle W, \mu, g \rangle$, where W is the message set, $\mu : \mathcal{V} \rightarrow W$ is the message correspondence, and $g : W \rightarrow \mathcal{O}$ is the outcome function, and the message correspondence μ has the following properties:

- Existence: $\mu(v) \neq \emptyset$ for all $v \in \mathcal{V}$.
- Privacy preservation: $\mu(v) = \bigcap_i \mu_i(v_i)$ for all $v \in \mathcal{V}$, where $\mu_i : \mathcal{V}_i \rightarrow W$ for all $i \in N$.

Protocol Π realizes choice correspondence $F : \mathcal{V} \rightarrow \mathcal{O}$ if $g(\mu(v)) \subseteq F(v)$ for all $v \in \mathcal{V}$.

Let $r = |\Gamma|$ be the size of the (finite) set of allocations. Denote the set of universal price equilibria in state v by $E(v)$, where $E : \mathcal{V} \rightarrow \Gamma \times \mathbb{R}^{nr}$ denotes the universal price equilib-

rium correspondence.⁷ The proof of the following is straightforward and omitted in the interest of space.

Lemma 2 Let $\langle W, \mu, \langle h, q \rangle \rangle$ be a nondeterministic communication protocol that realizes the Vickrey outcome rule F . Let $w \in W$ and let $\hat{\gamma} = h(w)$. If $v, v^* \in \mu^{-1}(w)$, then $F_2(v) = F_2(v^*)$, where the VCG payments are with respect to efficient allocation $\hat{\gamma}$.

We note in particular that $F_{2j}(w)$ is entirely independent of component v_j in the original profile v . Thus if $v_{-j}, v_{-j}^* \in \mu_{-j}^{-1}(w)$, then $F_{2j}(v) = F_{2j}(v^*)$.

Theorem 2 Communication protocol $\Pi = \langle W, \mu, \langle h, q \rangle \rangle$ realizes the Vickrey outcome rule F if and only if there exists an assignment $p : W \rightarrow \mathbb{R}^{nr}$ of prices to messages such that protocol $\Pi' = \langle W, \mu, \langle h, p \rangle \rangle$ realizes the universal price equilibrium correspondence E .

Proof. As mentioned, Mishra and Parkes [2004] provide the proof of sufficiency. Suppose protocol $\langle W, \mu, \langle h, q \rangle \rangle$ realizes Vickrey outcome rule F . For each $w \in W$, let $\hat{\gamma} = h(w)$, and let $p_i(\gamma) = \sup_{v_i \in \mu_i^{-1}(w)} [v_i(\gamma) - v_i(\hat{\gamma})]$ for all $i \in N$ and $\gamma \in \Gamma$. Note that $\hat{\gamma}$ is directly obtained from the information generated by Π , but state v and efficient allocations in the marginal-economies $\hat{\gamma}_{-j}(v)$ for $j \in N$ are not. However, we do not need to explicitly compute this information to implement a universal price equilibrium according to Definition 2. Let $q(w)$ be the VCG payments associated with efficient allocation $\hat{\gamma} = h(w)$. The first part of the proof of necessity, which shows that prices p are CE prices, is proved by Nisan and Segal [2004]. We will show that the prices p just constructed are in fact UCE prices corresponding to $\hat{\gamma}$, that are valid for all $v \in \mu^{-1}(w)$. Fix agents $i \neq j$. By Lemma 2, any two $v_{-j}, v_{-j}^* \in \mu_{-j}^{-1}(w)$ lead to the same VCG payment. Hence we obtain $\sum_{i \neq j} [v_i(\hat{\gamma}_{-j}(v)) - v_i(\hat{\gamma})] = \sup_{v_{-j} \in \mu_{-j}^{-1}(w)} \sum_{i \neq j} [v_i(\hat{\gamma}_{-j}(v)) - v_i(\hat{\gamma})]$, which holds for all $v_{-j} \in \mu_{-j}^{-1}(w)$ (call this equation (I)). By privacy preservation we can write the right-hand side as $\sup_{v_{-i,j} \in \mu_{-i,j}^{-1}(w)} \sum_{h \neq i,j} [v_h(\hat{\gamma}_{-j}(v)) - v_h(\hat{\gamma})] + p_i(\hat{\gamma}_{-j}(v))$.

By definition $p_i(\hat{\gamma}) = 0$, so Equation (I) gives: $[v_i(\hat{\gamma}) - p_i(\hat{\gamma})] - [v_i(\hat{\gamma}_{-j}(v)) - p_i(\hat{\gamma}_{-j}(v))] = \sum_{h \neq i,j} [v_h(\hat{\gamma}_{-j}(v)) - v_h(\hat{\gamma})] - \sup_{v_{-i,j} \in \mu_{-i,j}^{-1}(w)} \sum_{h \neq i,j} [v_h(\hat{\gamma}_{-j}(v)) - v_h(\hat{\gamma})]$. The right-hand side is at most 0, so $\hat{\gamma}_{-j}(v)$ is utility-maximizing for all agents $i \in N$ at prices p_i . The final step is to establish that $\hat{\gamma}_{-j}(v)$ is a revenue-maximizing allocation among agents $N \setminus \{j\}$. By privacy preservation, the supremum on the right-hand side of equation (I) can be brought within the summation. Using our price construction and rearranging, this yields: $\sum_{i \neq j} p_i(\hat{\gamma}_{-j}(v)) = \sum_{i \neq j} [v_i(\hat{\gamma}_{-j}(v)) - v_i(\hat{\gamma})] \geq \sum_{i \neq j} [v_i(\gamma) - v_i(\hat{\gamma})]$, for all $\gamma \in \Gamma$. This holds for all $v_{-j} \in \mu_{-j}^{-1}(w)$, so $\sum_{i \neq j} p_i(\hat{\gamma}_{-j}(v)) \geq \sup_{v_{-j} \in \mu_{-j}^{-1}(w)} \sum_{i \neq j} [v_i(\gamma) - v_i(\hat{\gamma})] = \sum_{i \neq j} \sup_{v_i \in \mu_i^{-1}(w)} [v_i(\gamma) - v_i(\hat{\gamma})] = \sum_{i \neq j} p_i(\gamma)$, for all

⁶The proof of this result, together with the associated definitions, closely follows the developments by [Nisan and Segal, 2004] who show that CE prices are necessarily revealed when a (nondeterministic) communication protocol computes an efficient allocation.

⁷This correspondence only maps states to personalized prices and efficient allocations, and not to efficient allocations in the marginal-economies with each agent removed. It is implied that the prices also support efficient allocations in the marginal-economies.

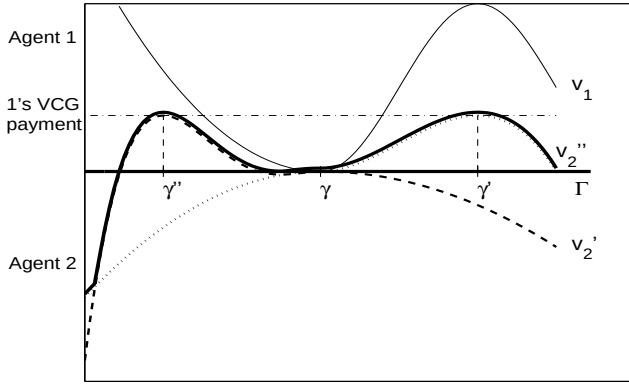


Figure 1: Constructing UCE prices.

$\gamma \in \Gamma$, where the first equality follows by privacy preservation. This completes the proof. $\square$

Figure 1 gives the intuition behind this result for the case with two agents. Given a message w , the figure shows all valuations that are consistent with w for agent 1 (here only v_1) and for agent 2 (v_2' and v_2''). The valuations are normalized so that the efficient allocation γ has value 0. The conditions for a CE require that all of agent 1's valuations consistent with w be above all of agent 2's valuations consistent with w . To construct valid CE prices, we take the envelopes of agent 1 and 2's valuations (lower and upper, respectively). Since agent 2's valuations are all consistent with the same VCG payments, they all peak at the same level. This ensures that the construction also satisfies the constraints of a UCE.

4.1 Using Universal Demand Queries: LeEA

A naïve way to compute VCG payments is simply to run the protocol once with all agents, then once with each agent removed for a total of $n + 1$ runs. We would then determine the value of the efficient allocation in each run. This gives us sufficient information to compute VCG payments. But given this new characterization result, we can instead modify the general elicitation framework to converge to UCE prices. We call this new framework *Learner Extend and Adjust* (LeEA). In the first stage we run the standard Learner, until CE prices and an efficient allocation are determined. The second stage uses universal demand queries. Whenever all the individual learning algorithms are stalled waiting to perform an equivalence query, we determine manifest allocations in the main economy (with agents N) and also in each of the marginal economies (with agents $N \setminus \{j\}$, for all j). In addition, we determine (manifest) universal CE prices, p . We can then issue query $\text{UNI-DEMAND}(B_i, p)$ to each agent i , where B_i is the vector of bundles currently allocated to the agent in the main economy and in each of the marginal economies, for $j \neq i$. If all agents reply YES, we have a UCE and can derive and implement VCG payments. Otherwise, we obtain counterexamples to push forward the individual learning algorithms.

5 Conclusions and Future Work

We presented novel applications of demand queries and demonstrated how they can be used in elicitation other than as parallels to equivalence queries. Specifically, we provided a

learning algorithm that uses demand queries with specialized prices for the class of atomic languages, a generalization of the OR and XOR languages. We also showed how demand queries can be used in a CA setting to make truthful bidding an *ex post* Nash equilibrium: we modify an elicitation protocol to converge to UCE prices by using universal demand queries, which are a simple generalization of demand queries. Indeed, we proved that UCE prices are *necessarily* implicitly discovered in any VCG protocol.

In future work, it would be useful to find a learning algorithm for the OR^* bidding language [Nisan, 2000]. OR^* representations are similar to OR representations except that they also allow *dummy* (phantom) items, that impose restrictions on which atomic bundles can be combined together. This bidding language can represent many natural valuations concisely but lies outside our class of atomic languages. As opposed to the L_k languages, knowing the atomics in an OR^* representation (without the dummy items) does not fully characterize all the valid atomic decompositions of a bundle. It is the structure generated by dummy items that must also be learned. In addition, we would like to completely characterize the relative power of demand and equivalence queries. We conjecture that demand queries are strictly more powerful than equivalence queries.

Acknowledgements

This work is supported in part by NSF grant IIS-0238147.

References

- [Bertsimas and Tsitsiklis, 1997] D Bertsimas and J Tsitsiklis. Introduction to Linear Optimization. Athena Scientific, 1997.
- [Blum *et al.*, 2004] A Blum, J Jackson, T Sandholm, and M Zinkevich. Preference elicitation and query learning. *Journal of Machine Learning Research (JMLR)*, 5:649–667, 2004.
- [Conen and Sandholm, 2001] W Conen and T Sandholm. Preference elicitation in combinatorial auctions. In *Proc. 3rd ACM Conf. on Electronic Commerce (EC-01)*, pages 256–259. ACM Press, New York, 2001.
- [Conen and Sandholm, 2002] W Conen and T Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. In *Proc. 18th Nat. Conf. on Artificial Intelligence (AAAI'02)*, 2002.
- [Jackson, 2000] M Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.
- [Krishna and Perry, 2000] V Krishna and M Perry. Efficient mechanism design. Technical report, Pennsylvania State University, 2000. Available at: <http://econ.la.psu.edu/~vkrishna/vcg18.ps>.
- [Lahaie and Parkes, 2004] S Lahaie and D C Parkes. Applying learning algorithms to preference elicitation. In *Proc. ACM Conf. on Electronic Commerce*, pages 180–188, 2004.
- [Mishra and Parkes, 2004] D Mishra and D C Parkes. Ascending price Vickrey auctions using primal-dual algorithms. Technical report, Harvard University, 2004.
- [Nisan and Segal, 2004] N Nisan and I Segal. The communication requirements of efficient allocations and supporting prices. To appear in the *Journal of Economic Theory*, 2004.
- [Nisan and Segal, 2004] N Nisan and I Segal. Exponential communication inefficiency of demand queries. Technical report, Hebrew University, 2004.
- [Nisan, 2000] N Nisan. Bidding and allocation in combinatorial auctions. In *Proc. 2nd ACM Conf. on Electronic Commerce (EC-00)*, pages 1–12, 2000.
- [Parkes, 2002] D C Parkes. Price-based information certificates for minimal-revelation combinatorial auctions. In *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, volume 2531 of *Lecture Notes in Artificial Intelligence*, pages 103–122, 2002.
- [Parkes, 2004] D C Parkes. Iterative combinatorial auctions. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, chapter 2. MIT Press, 2004.
- [Sandholm, 1993] T Sandholm. An implementation of the Contract Net Protocol based on marginal-cost calculations. In *Proc. 11th National Conference on Artificial Intelligence (AAAI-93)*, pages 256–262, July 1993.
- [Santi *et al.*, 2004] P Santi, V Conitzer, and T Sandholm. Towards a characterization of polynomial preference elicitation with value queries in combinatorial auctions. In *Proc. Conf. on Computational Learning Theory (COLT)*, 2004.

The Role of Clustering on the Emergence of Efficient Social Conventions

Josep M. Pujol^{*}, Jordi Delgado, Ramon Sangüesa[†] and Andreas Flache

Polytechnic University of Catalonia
Barcelona, Spain
{jmpujol,delgado,sanguesa}@lsi.upc.edu

University of Groningen
Groningen, The Netherlands
a.flache@ppsw.rug.nl

Abstract

Multiagent models of the emergence of social conventions have demonstrated that global conventions can arise from local coordination processes without a central authority. We further develop and extend previous work to address how and under what conditions emerging conventions are also socially efficient, i.e. better for all agents than potential alternative conventions. We show with computational experiments that the clustering coefficient of the networks within which agents interact is an important condition for efficiency. We also develop an analytical approximation of the simulation model that sheds some light to the original model behavior. Finally, we combine two decision mechanisms, local optimization and imitation, to study the competition between *efficient* and *attractive* actions. Our main result is that in *clustered* networks a society converges to an efficient convention and is stable against invasion of sub-optimal conventions under a much larger range of conditions than in a *non-clustered* network. On the contrary, in *non-clustered* networks the convention finally established heavily depends on its initial support.

1 Introduction

Social conventions, according to [Ullmann-Margalit, 1977], are a special type of norms related to coordination problems, that is, those regularities of behavior which are a result of being a solution of a recurrent coordination problem, which with time, turn normative. The emergence and stabilization of norms, including social conventions, is one of the fundamental problems of social sciences [Bendor and Swistak, 2001]. Intuitively, a social convention might be regarded as any rule of behavior or a behavioral constraint [Walker and Wooldridge, 1995]. A convention simplifies people's decision making problem by dictating how to act in certain situations. Therefore, social conventions help to reduce complexity and uncertainty, particularly when the environment is

open and dynamic. From this perspective it is obvious that mechanisms underlying the emergence of social conventions are of great interest for Distributed Artificial Intelligence.

[Shoham and Tennenholtz, 1992] addressed the question whether and how the *emergence* of conventions is possible from decentralized, local interaction, as opposed to conventions designed and enforced a priori by a *central* authority. Shoham demonstrated that self-coordination based on a simple local optimization rule can emerge from initial disorder. Following Shoham's, many researchers have studied and improved his seminal work, proposing other strategies rather than *HCR* [Walker and Wooldridge, 1995], studying the time needed before the convention is established [Kittock, 1993], adding complex interaction patterns among agents [Delgado, 2002], and many others.

However, an important question remains open, when two potential different conventions exist which one will be established at the end? This question is not applicable to the original work of [Shoham and Tennenholtz, 1992], since the game was a *pure* coordination game, where no action *pareto*-dominates the other. Nevertheless, the question applies to all coordination games where one action is more efficient than the other. The discussion in social sciences gives no clear answer to why and under what conditions efficient conventions may prevail. As [Bendor and Swistak, 2001] point out, there are at least two conflicting positions in sociology, the *strong functionalism* thesis and the *rational choice* view. *Strong functionalism* claims that norms and conventions exist because they are functional for the group, that is, they yield optimal collective outcomes. According to this perspective, the system should always end up in an efficient convention. However, the *functionalist* approach has been criticized for its lack of a microfoundation. Adherents of the *rational choice* view on norms argue that individuals adopt norms only when it serves their self-interest to do so. This includes the possibility that mechanisms such as social control may stabilize conventions that are an individually efficient response to the given constraints, but are socially inefficient.

In section 2 we describe our model. In section 3 we describe the experimental results and analyze the necessary conditions to converge to the optimal convention. In section 4 we analyze the time efficiency of the emergence of conventions. In section 5 we integrate and analyze experimentally optimization and imitation partially competing decision mecha-

^{*}This work was supported in part by EuropeAid @LIS TechNET project ALA2002/049-055/2209

[†]This work was supported in part by the Catalan Autonomous Government Project e-Catalunya

nisms in the model. Finally, main results are summed up in section 6.

2 Formal Model

We follow the conceptual framework introduced by [Shoham and Tennenholtz, 1992], and the extension proposed by [Delgado, 2002] that introduced complex networks as the underlying topology.

2.1 The Coordination Game

A set of N agents must choose to play one of two possible actions: either A or B . Accordingly to its current action, or state, an agent interact with its neighbors receiving an outcome defined in payoff matrix G .

		Agent j	
		A	B
Agent i	A	(x,x)	(u,v)
	B	(v,u)	(y,y)

Figure 1: Payoff Matrix of the Game G

The payoff matrix G defines a 2-person 2-choice symmetric coordination game provided that $x > v$ and $y > u$. The condition on the entries of G makes clear that to play the same action is the best choice. It is trivial to demonstrate that the game G has two nash-equilibrium, both agents playing either A or B . Most previous work focused in the study of *pure coordination games*: where $x = y = +1, v = u = -1$ [Walker and Wooldridge, 1995; Kittock, 1993; Shoham and Tennenholtz, 1997; Delgado, 2002]. In our approach we move beyond this work and allow for coordination games with equilibrium differing in social efficiency (i.e. $x > y$).

For simplicity we assume that coordination on action A is at least as profitable as coordination on action B : $x \geq y$. Thus, game G is defined as follows: $v = u = -1, y = +1$, and $x = \alpha$ provided that $\alpha \geq 1$. When $\alpha > 1$, coordination in B is a sub-optimal solution since there exists a pareto-efficient solution that pareto-dominates B , which is, coordination in action A .

2.2 Action Selection Rule and Dynamics

Our MAS is composed of N agents that interact only with its neighbors in the social network, playing the game G once per interaction. Every agent, say the k th, has memory M_k that records the M last interactions of agent k . The value of the position i of the memory M_k is the tuple $\langle a_k^i, p_k^i, t^i \rangle$, where a_k^i stands for the action played by k , p_k^i stands for the payoff received after playing action a_k^i , and t^i denotes the time the interaction took place. The initial action of the agents is set randomly with a probability r_B , which is the density of agents playing action B in the beginning.

Following [Shoham and Tennenholtz, 1992] we will use the *Highest Cumulative Reward (HCR)* action selection rule. Intuitively *HCR* says: if the accumulated payoff obtained from playing A is bigger than that from playing B then keep on A , otherwise change to action B . The *HCR* rule is very

appropriate since it provides: 1) Locality: the selection function only depends on the agent's personal history. No global knowledge of the system is required, not even the payoff matrix of the game. 2) Adaptability: the agent learns from its experience without assuming cognitive capabilities. These characteristics are very important in MAS.

The dynamics of the system are as follows. At each time step t , an agent k is randomly activated. Once the agent is activated, it plays the game G with an agent randomly chosen from k 's neighborhood, say agent l . The result of the interaction is stored into agent k 's memory M_k , removing the oldest entry if necessary. Finally, agent k must decide whether to change its action or not. To do so it uses the *Highest Cumulative Reward* rule. Agent k will compute the payoff received for using action S on the last M activations: $P_S^k = \sum_{i: a_k^i = S} p_k^i$, where $S = \{A|B\}$. Agent k will switch to action $\bar{S}$ if $P_{\bar{S}}^k > P_S^k$. Agent l also carries out the memory storage and the action updating while the rest of agents remain still. The system ends once all agents play either action A or B , which means, that a convention on either A or B have been established.

2.3 Underlying Topology

To model qualitatively different interaction structures, we use several graph models that recently have been shown to have profoundly different effects on cooperation and diffusion dynamics in MAS [Watts and Strogatz, 1998; Albert and Barabási, 2002; Delgado, 2002]. The chosen graphs are: a) Random graphs: $R_N^{(k)}$, where N is the number of nodes, and $\langle k \rangle$ is the average connectivity, that is, the average size of node's neighborhood. Random graphs have a clustering coefficient that tends to zero. The average path length grows logarithmically in function of N , the number of nodes. b) Regular graphs: C_N^k , regular graphs display an extremely high clustering coefficient, while its average path length and diameter grows linearly. Which means, for big graphs the average path length is very long, which does not agree with empirically studied networks. However, regular graphs display the close-knit property due to its high clustering coefficient, which does agree with empirical studies. c) Small-world graphs: $W_N^{(k),p}$, these are *highly-clustered* graphs (like regular graphs) with small average path length (like random graphs). This is the small-world property. We chose the [Watts and Strogatz, 1998] model as model of small-world graphs, where p is the rewiring probability. d) Scale-free graphs: $S_N^{(k),-\gamma}$, these are graphs with a connectivity distribution $P(k)$ of the form $P(k) \propto k^{-\gamma}$. The connectivity degree, the number of neighbors of a node, decays as a potential law. This favors the so-called *fat-tail* phenomena; few nodes with an extreme high connectivity. We chose the [Barabási and Albert, 1999] model as a model of scale-free graph.

Recent studies on empirical networks show that neither regular nor random graphs appear in nature. Noticing this, [Delgado, 2002] studied the effect of *complex* networks (small-world and scale-free networks) on the emergence of coordination and found that this class of networks were as efficient as the complete graph in terms of time to reach a convention, $O(N \log N)$ compared to $O(N^3)$ from regular

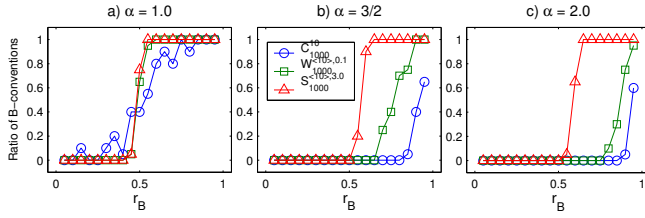


Figure 2: Ratio of conventions to action B . The x -axis is the initial density of agents playing action B : r_B , ranging from .05 to .95. The y -axis is the ratio of B -conventions, that is, the number of replications that ended up in all agents playing B over the total number of replications (20). In sub-figure a) there is no efficient convention, $\alpha = 1$. In sub-figure b) and c) all agents playing action A is the efficient convention, $\alpha = \frac{3}{2}$ and $\alpha = 2$ respectively.

graphs, already shown by [Kittock, 1993]. It is important to stress that many empirical social networks such as the collaboration network among actors, the co-authorship network of researchers on mathematics, the researchers on neuroscience and many others are classified as scale-free networks. However, for these empirical social networks the clustering coefficient found is very high: 0.79, 0.59, 0.76 respectively. Let us show the clustering coefficient of graphs we will use in the experiments: $C_{1000}^{10}=0.666$, $W_{1000}^{<10>,0.1}=0.492$, $S_{1000}^{<10>,-3}=0.0433$ and $R_{1000}^{<10>}=0.0081$. Notice that empirical social networks are much more clustered than the scale-free networks yield by the [Barabási and Albert, 1999] model. But we still use this model since scale-free models are focused in reproducing the connectivity degree rather than clustering. This fact must be kept in mind when analyzing the results and conclusions. For a comprehensive survey in complex networks see [Albert and Barabási, 2002].

3 HCR-Model Experimental Results

We use a graph classification based on the clustering coefficient, regular and small-world graphs will be called *highly-clustered* graphs, and scale-free and random graphs will be called *low-clustered* graphs. For the sake of clarity we did not include results on random graphs in the figures, since they behave like scale-free graphs.

Let us stress which are the key manipulations in the parameters of our model. We systematically variate the proportion of agents who initially adopt the convention B (r_B), the efficiency gap α between the two conventions, and finally, the type of network, in particular its clustering coefficient, that defines the agent's neighborhood.

Another thing to mention before getting into the details is that there are two clear phases: 1) all the replications converging to A , and 2) all the replications converging to B . There is as well a space between these two phases where the system converges, with some replications to A and some to B , which we call *transitional* space, that is wider or narrower depending on the underlying topology and α . There is a critical point r_B^* that sets the boundaries between the two phases; and the *transitional* space may be defined with an ε , such that the result is: $r_B^* \pm \varepsilon$.

Now, let us comment the results of the experiments on the HCR model. It is helpful to first consider the case where both conventions are equally efficient ($\alpha = 1$), which is the case of a *pure coordination game*. The results for this case are displayed in sub-figure 2.a. When the initial number of agents playing B is less than half of the population, that is, $r_B < \frac{1}{2}$, the system ends up establishing the convention on B . And, when $r_B > \frac{1}{2}$ the convention on A is established. The subfigure shows in particular that this result is widely independent of the network topology. It also shows that the transitional space is very narrow except for the regular graphs.

This pattern changes when convention differ in efficiency. Sub-figures 2.b and 2.c show the case where the payoff for action A exceeds the payoff for B ($\alpha > 1$). To understand the results, consider an example. Let $\frac{3}{4}$ of the population follow the action B , and $\frac{1}{4}$ follow the action A . In this case, although B is the initially chosen action for most of the agents, coordination in A is more efficient since it yields a better payoff. Thus, which will be the final convention agreed by the whole population? The answer depends on 1) how much better off, more efficient, is the action A over B , denoted by α . And 2) the underlying topology. If $\alpha = \frac{3}{2}$ (sub-figure 2.b) the final convention will be B when the underlying topology corresponds to a random or scale-free graph, A when having a regular graph, and can be both when having a small-world graph. By increasing the efficiency of coordination on A to $\alpha = 2$ (sub-figure 2.c) both small-world and regular graph converge to convention A , whereas random and scale-free graph still converge to convention B . The explanation behind the result is striking, *low-clustered* graphs seems very sensitive to the initial population density, whereas *highly-clustered* graphs behave in the opposite way, they are more sensitive to the efficiency of a particular action.

The conclusion derived from these results can be summed up as follows: *low-clustered* agent communities where a convention already exists will not be infected by a set of agents who play a new action, even though it is more efficient. Conversely, *highly-clustered* agent communities can be infected by a new action if the new action is more efficient, replacing the current convention for the convention on the efficient action.

Therefore, *highly-clustered* agent communities are more innovative, or adaptive, since a new action can be spread and finally established as a convention. The drawback would be that this community would be unstable due to its receptiveness to new action, and in the transient time needed to reach a stable new convention no coordination will exist. In contrast, *low-clustered* agent communities are very stable since the infection with a new convention is unlikely to happen, but on the other hand they are reluctant to adopt new actions even though they are more efficient, thus, becoming very conservative and static communities. We must remark that studies on empirical social networks have shown that these networks are very clustered [Albert and Barabási, 2002].

We do not provide a proof of the system's convergence. Nevertheless, throughout all the simulation runs, with their corresponding replications, the system has always converged to a convention with an upper bound of $O(N^3)$.

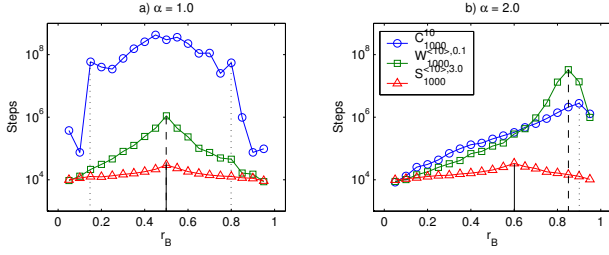


Figure 3: Time to converge to a convention, the value is the mean over 20 replications. The x -axis is the initial density of agents playing action B : r_B , ranging from .05 to .95. The y -axis is the number of interactions (steps) before reaching a convention. Note the logarithmic scale. In the left sub-figure both conventions, A or B . In the right sub-figure A is the optimal convention since $\alpha = 2$.

On figure 3 (left), α is set to 1, therefore both conventions are equally efficient. It can be observed that the regular graph takes much longer to converge than the rest of graphs, which is completely consistent with the findings of [Kittock, 1993] and [Delgado, 2002]. Notice the existence of peaks in $r_B = \frac{1}{2}$ in all the graphs except regular. These peaks are close to the critical point $r_B^* = \frac{1}{2}$ where the transition takes place. This is a typical behavior, when the system is close to a transition boundary it takes longer to converge [Yeomans, 1992]. The regular graph, however, is a particular case since it shows a *plateau* rather than a peak. This is due to its wide *transitional* state, in which different replications of the same setting can converge to different conventions.

On figure 3 (right) the convention A is more efficient than convention B , since playing action A yields payoff $\alpha = 2$, when playing action B yields a payoff 1. In this case the *plateau* of the regular graph does not appear because its *transitional* space narrowed as shown in figure 2.c. Thus, we see only the peaks in r_B where the transition takes place $r_B^* = \{.6, .85, .90\}$ for scale-free, small-world, and regular graphs respectively, which correspond to the transitions observed in figure 2.c. Again, we observe that the convergence is more inefficient in time when the initial r_B is close to r_B^* . It is important to remark that the regular graph has improved its efficiency in reaching the convention, that is, it is more efficient than the small-world graph when $\alpha = 2$. As was shown by [Young, 1999], convergence to a risk-dominant equilibrium, which in our case corresponds to the pareto-efficient one, is surprisingly rapid provided a close-knit (clustered) graph. The scale-free graph is always the most efficient graph in terms of convergence no matter α , however, scale-free graphs do not allow convergence to the efficient convention unless the initial number of agents playing A is very big, 40% compared to the 15% needed with a small-world graph, with α set to 2).

4 Analytical Model

First of all, we replace the role of past interactions stored in agent i 's memory (M_i) by the current state of agent i 's neighbors; notice that in our model we set the memory size to the average connectivity, $M = \langle k \rangle$. Instead of apply-

ing the *HCR*-rule over M_i we will transform it as follows: Let us take k_S as the number of neighbors in the same state as agent i , and $k_{\bar{S}}$ the number of neighbors in the opposite state. Therefore, the update is performed if $k_{\bar{S}} > \phi k_S$, where ϕ is the payoff yield by the payoff matrix G when playing (S, S) . Therefore, the switch from playing A to play B is done when $k_B > \alpha k_A$, since $G(A, A) = \alpha$. Similarly, the switch from playing B to play A is done when $k_A > k_B$, since $G(B, B) = 1$. The probability of updating the state is defined by equation 1, we decided to introduce some stochasticity to compensate the fact that now the update is calculated by the current state of the neighbors instead of by the interactions, as the *HCR*-rule does.

$$f_{\beta, \phi}(k_{\bar{S}}) = \frac{1}{1 + e^{\beta(\phi - (\phi + 1) \frac{k_{\bar{S}}}{k_S + k_{\bar{S}}})}} \quad (1)$$

We will use what in physics is called a *mean-field* argument [Pastor-Satorras and Vespignani, 2001]. Let $N_B(t)$ be the number of agents playing action B at time t , and $\rho(t) = \frac{N_B(t)}{N}$ be the density of agents playing B . A first approach is to assume the following *homogeneity* condition: for every agent with k neighbors, the number of neighbors in state B is $k_B(t) \simeq k\rho(t)$. This condition is completely fulfilled for random graphs, and approximately fulfilled for scale-free and small-world graphs (when $p \rightarrow 1$). Nevertheless, this *homogeneity* condition is not fulfilled for regular and for small-world graphs for low values of p . What breaks the *homogeneity* is the clustering coefficient, for *low-clustered* graphs this condition holds since the global density of agents in state S corresponds to the proportion of neighbors in state S . Intuitively, the clustering coefficient can be defined as the probability that a node i and a node j have a link provided that node l has a link to both i and j . Thus, when clustering tends to 0 the node's neighborhood is a good sample of the graph. However, when clustering coefficient is high the node's neighborhood is not a sample of the graph, since its neighbors form a *clique*, a *close-knit* group. Therefore, we propose a new *homogeneity* condition that takes clustering (cc) into account. Let us define cc as the clustering coefficient, provided that cc is the probability of agent l 's neighbors being also neighbors, $(1 - cc)k$ is the number of neighbors which are not in the l 's *clique*, and to whom the previous *homogeneity* condition holds. Therefore, for an agent playing A with k neighbors, the number of neighbors in the opposite state (B) is $k_A \simeq (1 - cc)k\rho(t)$, which is the *cc-biased-homogeneity* condition.

Now we can write an equation for the evolution of $\rho(t)$. First, notice that the variation of $\rho(t)$ after a small time interval Δt is proportional to Δt , that is, $\rho(t + \Delta t) = \rho(t) + \frac{\partial \rho(t)}{\partial t} \Delta t + O(\Delta t^2)$. Then, we can neglect the $O(\Delta t^2)$ term (since we perform a continuum approximation $\Delta t \rightarrow 0$) and compute the variation of $\rho(t)$ as the balance between the agents switching from action A to B and the agents switching from action B to A . On one hand, the fraction of agents in A (that is, $1 - \rho(t)$) that change to state B in a time interval Δt is the product $(1 - \rho(t))f(\rho(t))\Delta t$, provided Δt is small enough; on the other hand, the fraction of agents that switch from action B to A in Δt is $\rho(t)f(1 - \rho(t))\Delta t$, also for small

Table 1: Fixed points ρ^* of the Analytical Model: those are the critical point such that $\frac{\partial \rho}{\partial t} = 0$. In brackets the critical points observed experimentally with the *HCR* model (r_B^*) (figure 2). The parameters that model the graph are the clustering coefficient cc , and the average connectivity $\langle k \rangle$ set to 10.

$\rho^*(r_B^*)$	$\alpha = 1$	$\alpha = \frac{3}{2}$	$\alpha = 2$
$cc=0.666$	0.5 (0.5)	0.831 (0.90)	$\bar{A}$ (0.95)
$cc=0.492$	0.5 (0.5)	0.68 (0.7)	0.827 (0.85)
$cc=0.0433$	0.5 (0.5)	0.566 (0.55)	0.614 (0.60)
$cc=0.0081$	0.5 (0.5)	0.560 (0.55)	0.607 (0.60)

Δt . Thus after $\Delta t \rightarrow 0$, the mean-field equation for $\rho(t)$ can be written as

$$\frac{\partial \rho(t)}{\partial t} = (1 - \rho(t))f(\rho(t)) - \rho(t)f(1 - \rho(t)) \quad (2)$$

After substitution of $f_{\beta, \phi}$ to which the *cc-biased-homogeneity* condition has been applied. And setting $\beta = \langle k \rangle$ and $\phi = (S, S)$ the equation reads

$$\frac{\partial \rho}{\partial t} = \frac{1 - \rho}{1 + e^{\langle k \rangle (\alpha - (\alpha + 1)(1 - cc)\rho)}} - \frac{\rho}{1 + e^{\langle k \rangle (1 - 2(1 - cc)(1 - \rho))}} \quad (3)$$

We want to study the stable fixed points of 3 since these will give us information on the final state of the system. Thus we must find the solutions of $\frac{\partial \rho}{\partial t} = 0$. As we can see in 4, stable fixed points are $\rho^* \sim 0$, and $\rho^* \sim 1$ (these have been computed numerically) and the unstable fixed-point lies in $(0, 1)$. In table 1 we display the unstable fixed points of the density equation, which are possible critical points of the *HCR* model provided our assumptions (see above) are correct. Furthermore, in table 1 we find a comparison between the analytical unstable fixed points and those critical points coming from the simulation of the *HCR* model. Notice that the plausibility of the simplifying assumptions behind our analytical model is supported by the agreement between analytical and experimental results (remind that r_B was sampled with a resolution of .05).

Figure 4 shows the variation in ρ , $\Delta \frac{\partial \rho}{\partial t}$, for different α and cc . We can see the effect of α enlarging the basin of attraction of convention *A*. On the other hand cc has the effect of reducing the amount of variation due to the effects of the *cc-biased-homogeneity* condition. Consequently, the time elapsed to reach a convention will be longer, and fluctuations in initial conditions will have a bigger impact. As a matter of fact, the experimental results on the *HCR* model show us these two consequences apply for highly-clustered graphs such as the regular graph. The convergence time is much higher compared to non-clustered graphs. And the transitional space, where the system can converge either in *B* or in *A* for the same initial parameters, is wider.

5 The Role of Imitation

In this section we will modify the model based of the *HCR*-rule introducing an imitation propensity i_S , which is the probability that, after a dyadic interaction where at least one agent

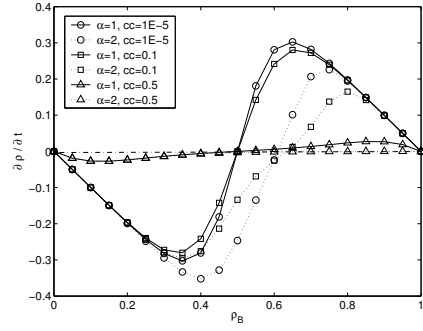


Figure 4: Study of the fixed points of the density equation 3. ρ is the density of agents playing *B*. Clustering coefficient is set to $cc = \{1E - 05, 0.1, 0.5\}$, α is set to $\{1, 2\}$

was playing action *S*, both agents end up playing action *B* regardless of the *HCR* action updating rule. By introducing an imitation propensity we model the effect of having an *attractive* action which more likely to be chosen. Imitation is considered as the key factor of the adoption of norms, and by extension, conventions. How does the existence of an *attractive* action affect the final convention reached by the agents? And what if the *attractive* action happens to be sub-optimal? In our model coordination on action *A* is most efficient solution (if $\alpha > 1$), however, action *B* (if $i_B > 0$) might become a better replicator since it can be adopted by imitation as well as adopted by the learning process (*HCR*-rule).

The effect of i_B heavily depends on the clustering of the underlying topology. For instance, when $\alpha = 2$ and $i_B = .4$ the clustered communities adopts the *attractive* convention (*B*) over the efficient one (*A*) regardless of the initial density (r_B), while *non-clustered* communities will still adopt *A* provided $r_B < .25$. Therefore, it might be derived that *non-clustered* communities are more resilient against *attractive* conventions in favor of efficient conventions. This result would seem to contradict the previous claim, that clustered communities are better off in converging to the efficient convention. However, when i_B is small enough, for example when $i_B = .1$ the opposite effect is observed, clustered communities keep on converging to the *efficient* convention regardless of the initial density of agents. This two-fold behavior is perfectly clear in the case of the regular graph (left column of figure 5). There is a threshold i_B^* under which the system ends up in the efficient convention, and over which the system ends up in the *attractive* convention. For example $i_B^* = .2$ when $\alpha = 2$ and the underlying topology is the regular graph. Notice, that this threshold is not found in the case of the small-world graph whose clustering coefficient is high, although not so high as the case of the regular graph. However, even without the threshold we observe a similar behavior than in the regular graph although more progressive. If we compare it against the behavior of *low-clustered* graphs, we find again that the system is more resilient to the *attractive* action invasion for low values of i_B . For high values, on the contrary, the system is very receptive to an invasion of agents playing the *attractive* action. We must state that the model is interesting for small values of i_B , for high values of i_B the dichotomy of the agent to choose between the efficient or the *attractive* action disappear and becomes an epidemic spread model [Pastor-Satorras and Vespignani, 2001].

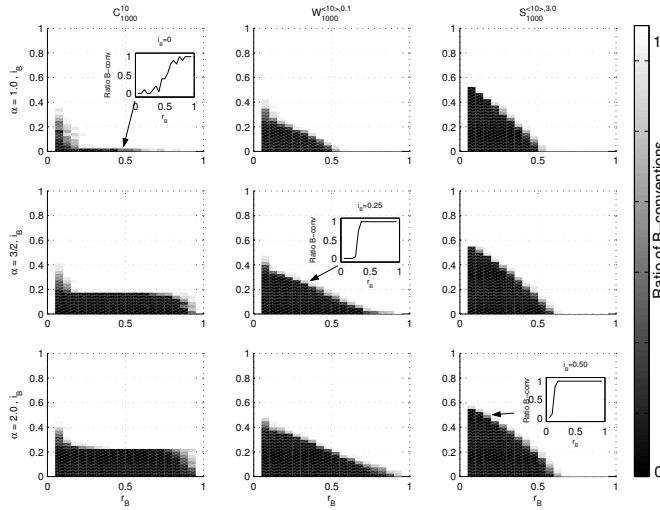


Figure 5: Effect of imitation propensity (i_B) on the ratio of conventions on action B . i_B ranges from $[.0.975]$ in a $.0275$ resolution, r_B ranges from $[.05.95]$ in a $.05$ resolution. The sub-figures within the figures are for the sake of clarity: setting the parameter i_B we observe the evolution of B -conventions depending on r_B , which correspond to the way the previous results were shown (when $i_B = 0$).

6 Concluding Remarks

Our research aimed to identify conditions under which local interactions in a multi agent community can give rise to an efficient convention. We have shown that the key factor is the clustering of the underlying agent's social network. When communities are *highly-clustered* the system converges to the pareto-efficient action even though the initial population choosing that action was clearly a minority. This suggests that the efficient convention is a stable convention because it cannot be invaded by a set of agents playing another sub-optimal action. However, a sub-optimal convention can be replaced by a set of agent playing an action that yields a better payoff. Accordingly, when the clustering coefficient is high the system always converges to the most efficient convention and this convention is stable. On the other hand when clustering tends to zero the adopted convention depends solely on the density of agents following an action. If the majority of agents play the sub-optimal action the inefficient convention will be established, and it will be stable. To back up our findings, we provided an analytical approximation that reproduces the results observed in our model based in the *HCR*-rule. To do so, we had to introduce a new *homogeneity* condition which let us work in clustered graphs, where the classical mean-field *homogeneity* condition is not met.

In accordance with the *strong functionalism* thesis from classical sociology, we found that in certain graphs the agent system was capable to find and maintain the optimum in the stable state. However, this only applies to *highly-clustered* communities, which resemble many empirical social networks. At the same time, our model also corresponds in two respects to the view that rational choice theorists in sociology take on social norms. First, we have shown that global effi-

ciency arises from individual goal oriented actions. Second, we found that under certain conditions optimizing individual actions fail to generate socially efficient outcomes, a problem that is central to the contemporary discussion about the emergence of conventions and norms.

To conclude, our results seem to correspond more with a rational choice on norms than with the strong functionalism thesis. We have shown that socially optimal conventions can arise from individual optimization, but there is no guarantee that this happens. In this sense, our model matches well the ample evidence of examples of suboptimal conventions, for example in market processes. We believe that a part of the explanation for this may lie in the competition between optimizing and imitation that we have addressed with our model. We have shown that imitation processes make it possible for a sub-optimal yet *attractive* action to overthrow the efficient action, and become stable, provided that its *attractiveness* is high enough to be worthy of imitation.

References

- [Albert and Barabási, 2002] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Review of Modern Physics*, 74:47–97, 2002.
- [Barabási and Albert, 1999] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
- [Bendor and Swistak, 2001] J. Bendor and P. Swistak. The evolution of norms. *American Journal of Sociology*, 106(6):1493–1545, 2001.
- [Delgado, 2002] J. Delgado. Emergence of social conventions in complex networks. *Artificial Intelligence*, 141:171–185, 2002.
- [Kittock, 1993] J.E. Kittock. Emergent conventions and the structure of multi-agent systems. In *In Proc. of the 1993 Santa Fe Inst. Complex Systems Summer School*, pages 384–389, 1993.
- [Pastor-Satorras and Vespignani, 2001] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Phys. rev. Lett.*, 86:3200–3203, 2001.
- [Shoham and Tennenholtz, 1992] Y. Shoham and M. Tennenholtz. Emergent conventions in multi-agent systems. In *In Proceedings of Knowledge Representation and Reasoning*, pages 225–231, 1992.
- [Shoham and Tennenholtz, 1997] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: Modeling, analysis and simulations. *Artificial Intelligence*, 94:139–166, 1997.
- [Ullmann-Margalit, 1977] E. Ullmann-Margalit. *The Emergence of Norms*. Oxford Univ. Press., 1977.
- [Walker and Wooldridge, 1995] A. Walker and M. Wooldridge. Understanding the emergence of conventions in multi-agent systems. In *In Proceedings of 1st International Conference on Multi-Agent Systems*, pages 384–389, 1995.
- [Watts and Strogatz, 1998] D.J. Watts and S.H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.
- [Yeomans, 1992] J.M. Yeomans. *Statistical Mechanics of Phase Transitions*. Oxford University Press, 1992.
- [Young, 1999] H. Peyton Young. Diffusion in social networks. Technical report, Center on Social and Economics Dynamics, Brookings Institution, 1999.

Distributive and Collective Readings in Group Protocols

Silvia Rossi

ITC-irst – Povo, Trento, Italy
Istituto di Cibernetica, CNR Napoli – Italy
silvia.rossi@dit.unitn.it

Sanjeev Kumar and Philip R. Cohen

Oregon Health and Science University
Beaverton, OR – USA
{skumar, pcohen}@cse.ogi.edu

Abstract

Collaborative applications require protocols that specify how distributed entities interact with one another in order to achieve a specified behavior. Many different kinds of relationships can be established between these entities as a result of such interactions. Distributive and Collective readings are two important ways to characterize group interaction. Starting from an attempt-based semantics of group communicative acts, we distinguish between these two concepts and evaluate group protocols with respect to formation of different types of teams during the interaction.

1 Motivations

Group communication is common in collaborative applications of all kinds such as clustering, grid computing, transactions and database replication, distance learning, application sharing, distributed interactive simulations, on-line games and financial markets. In all these applications the emphasis shifts to distributed global collaboration rather than a centralized view of the problem. A protocol definition specifies how distributed elements interact with one another in order to achieve a specified behavior and therefore, protocols are critical for the interoperability of the various components. Most of the protocols presented in multi-agent systems propose bilateral interaction among agents, with few cases of protocols that try to capture cases involving groups of agents. These representations of bilateral interactions are very useful in describing isolated conversation between two agents, but fail when the structure of the conversation becomes complex or the number of agents involved within the conversation grows.

One of the most challenging examples in distributed interactions and group communication is represented by the third generation of grid computing systems [De Roure *et al.*, 2003]. Grid computing enables geographically distributed systems to form dynamic *virtual organizations* [Foster *et al.*, 2001] whose members use shared and private resources to collaborate in order to find a solution to a common problem. Given this situation, according to Jennings [Jennings, 2001], agent-oriented approaches are well-suited to engineering complex, distributed software systems. In the case of distributed grid computing (i.e., when there is no single big computer for

collecting results), it should be possible for each computer to send messages and results to groups of other components without knowing in advance who are the recipients and who actually need such results. In this sense, an agent communication language should support communication addressed to individuals as well as to groups, where a group may have a stable, known membership (i.e. it can be addressed by a referring name) or its membership may be unknown. For example, in a publish/subscribe system, senders label each message with the name of a topic ("publish"), rather than addressing it to specific recipients and the messaging system then sends the message to all eligible systems that have asked to receive messages on that topic ("subscribe").

But complex systems like grids not only require that collections of components be treated as a single conceptual unit when viewed from a different level of abstraction, they also involve the changing of relationships between their various components. Group communicative acts by themselves are not sufficient to model such situations. The group communication semantics of speech acts is sufficient to model one-shot one-to-many communicative interactions - the intended recipients can be multiple agents and/or groups of agents. However, we need group protocols in order to enable computational mechanisms for flexibly forming, maintaining, and disbanding organizations or teams. This is not just a matter of allowing new groups to appear and then disband, but also allowing different kinds of groups that have different types of commitments among the group members themselves. The use of communicative acts in different contexts and in different orders leads to the creation of different types of commitment among the agents. Different commitments imply different methods to form and disband teams of agents.

In this paper, starting from an attempt-based semantics of communicative acts [Cohen and Levesque, 1990] and the semantics of group communication [Kumar *et al.*, 2000], we analyze the concept of *distributive reading* vs. *collective reading* [Clark and Carlson, 1982] in group protocols. These types of group communication will be evaluated with respect to the formation of different types of teams during interaction. In the case of a collective reading, we claim that joint commitments and mutual beliefs have to be established among all the members of the group. We will see that by means of group communication, one can obtain a joint commitment among the members of a group that has different properties from the

one achieved by means of single utterances.

2 Formalism

Addressing an utterance to more than one recipient is more than a convenient way to refer to many people simultaneously. For example, it may be that we want to make other people aware of what is going on say, by using the *Carbon Copy* field in e-mails. Even if everybody in the "TO" and in the "CC" fields receives the e-mail, it is clear that the intended recipients of our utterance are just the people in the "TO" field, but, we also want the people in the "CC" field to be aware of it. So, the reader of an e-mail has to be able to make certain inferences about the mental state of others who the e-mail is addressed to or who will be reading the e-mail. Moreover, consider the utterance: "*Can both of you help me in lifting this table?*". The main effect of this utterance is not only that both the recipients are aware of the request, but also that both of them know that the request was made to *all the recipients as a group* and the requested action should be a joint action. An agent communication language should be able to properly convey these nuances of a requester's intentions about the performers of an action.

In a previous paper [Kumar *et al.*, 2000], we introduced a framework for semantics of group communicative acts and provided a semantics of the REQUEST communicative act within this framework. It was demonstrated that this group communicative act can be performed even when the requester does not know about the intended actor and when the requester does not know everyone who will get the message. Even though that paper emphasized the importance of distinguishing between a group doing an action as an entity (collective reading) and everybody in a list of individuals performing the action (distributive reading), it provided a clear semantics only for the latter. Furthermore, plurals and utterances addressed to groups can introduce semantic ambiguities. For example, there can be a potential ambiguity as to whether the aggregated entities are to act individually (distributive) or they are to act together as one (collective). This paper makes a clear distinction in the semantics of communicative acts to enable understanding whether an utterance requires a collective or a distributive reading. In particular, we want the agents that receive a communicative act to interpret its correct reading (distributive or collective) so that they can act accordingly. Moreover, group communicative acts have to be inserted and analyzed in the wider context of protocols, where distributive and collective reading of an utterance is important in order to generate the next communicative act of the protocol. However, in this paper we will not be concerned with how the action is done and how the agents coordinate in order to perform the action.

We regard a group as being defined by a characteristic function such as the membership property. This can be captured by a predicate consisting of a free variable that ranges over individuals. With Greek letters we will represent groups' names, and we use the same symbol in a functional notation to denote the associated membership predicate. For example, γ is a group having the membership predicate $\gamma(x)$ where x is a free variable. We will underline the entities that

represent groups ($\underline{\gamma}$, for example) to indicate the *distributive readings* of that group. An entity without an underline can be either an individual or a group and indicates a *collective reading*. If ψ is a formula with term $\underline{\gamma}$, and x does not appear in ψ , and $\gamma(x)$ is the property predicate that corresponds to $\underline{\gamma}$, and $\psi(x)$ is a formula formed by replacing $\underline{\gamma}$ with x in ψ , then $\psi(\underline{\gamma}) = \forall x \gamma(x) \supset \psi(x)$. For example: $(BEL \underline{\gamma} p) = \forall x \gamma(x) \supset (BEL x p)$. In other words, everyone in group γ believes p . In case of ambiguity in the determination of the scope, we will mark the group term that it applies to and the starting bracket with the free variable in the superscript. $(^y BEL \underline{\gamma}^y (BEL x (^z BEL \underline{\gamma}^z p))) = \forall y \gamma(y) \supset (BEL y (BEL x (\forall z \gamma(z) \supset (BEL z p))))$.

This paper will concentrate on group protocols in which the intended recipient of a message is a group of (possibly unknown) agents. The difference between a distributive reading and collective reading will be shown to depend upon whether the intended actors are separate individuals or groups. Here, we summarize the results from literature that this paper builds upon and redefine semantics for the communicative acts that we need for our example. The terms α , β , and γ in the following definitions can represent either groups or individuals. Here, α is the entity performing the communicative act, β is the recipient (including the "Over-hearers") of the message, and γ is the intended recipient (or intended actor). The semantics of the communicative acts as described below has the implicit requirement that the intended recipients of an utterance are a subset of all the recipients. Moreover, the sender of the utterance is always a single agent that may be an individual that acts on behalf of the group. The analysis of authorized representative of groups is left for future work.

A persistent weak achievement goal or PWAG represents the one-way commitment of one agent directed towards another and it is used to define the semantics of various communicative acts. A PWAG made public via performance of a communicative act represents a social commitment using mentalist notions. The definition of $(PWAG \alpha \gamma p q)$ states that an entity α has a PWAG towards another entity γ to achieve p with respect to q when the following holds: if α believes that p is not currently true then it will have a persistent goal to achieve p , and if it believes p to be either true, or to be impossible, or if it believes the relativizing condition q to be false, then it will have a persistent goal to bring about the corresponding mutual belief with γ . In the philosophy of language, it is argued that the illocutionary effect of a speech act consists of the hearers recognition of the speakers communicative intention. A communicative act succeeds when the hearer successfully recognizes the speakers intention and it is satisfied when the hearer successfully acts on the speakers intention. Communicative acts must be characterized as attempts because there is a possibility that the act may not succeed. Accordingly, attempt is defined as having a goal and an intention. An ATTEMPT to achieve p via q represented as $(ATTEMPT \gamma e p q t)$ is a complex action expression in which the entity γ is the actor of the event e and just prior to e , the actor chooses that p should eventually become true, and intends that e should produce at least q relative to that choice. Mutual belief for groups is defined in terms of unilateral mutual belief, or BMB [Cohen and Levesque, 1990], that

is treated as a semantic primitive [Kumar *et al.*, 2000].

Definition 2.1 Mutual Belief

$$(MB \gamma p) \equiv \forall x \gamma(x) \supset (BMB x \gamma p)$$

MB between two groups ($MB \beta \gamma$) is defined similarly. In this article, we assume that mutual belief is established by default. Establishing mutual belief by default means that agents make certain defeasible assumptions if they have no information to the contrary. If at a later time, the agent having a defeasible belief discovers that its belief was incorrect than it needs to revise that belief¹. But until that happens, an agent can assume the defeasible belief and make inferences based on that belief. The symbol $\vdash_d$ represents the deduction relation and $\vdash_d (\varphi \Rightarrow \psi)$ denotes that ψ can always be derived from φ using defeasible rules. In particular, we assume that the sender and the receivers of a communicative act use the same defeasible rules, like, for example, that i) the speaker does not change its mental state about the propositional content of the communicative act immediately after performing it; ii) the sender of a communicative act is sincere toward the recipients about its (i.e. sender's) mental attitude; iii) if a communicative act has been done successfully then the sender and the hearer mutually believe that it has been done [Kumar *et al.*, 2002]. This enables us to define the other communicative acts that we require for our example.

Definition 2.2 Request

$$(REQUEST \alpha \beta \gamma e a q t) \equiv (ATTEMPT \alpha e \phi \psi t) \\ \text{where } \phi = ({}^z DONE \gamma^z a) \wedge \\ [PWAG \gamma^z \alpha (DONE \gamma^z a) (PWAG \alpha \gamma ({}^w DONE \gamma^w a) q)] \\ \text{and } \psi = [BMB \beta \alpha [GOAL \alpha (BEL \gamma [PWAG \alpha \gamma \phi q])]]$$

The goal of the REQUEST is that the intended actor eventually does the action a and also has a PWAG with respect to the requester to do a . The intention of the REQUEST is that all the recipient β believe there is a mutual belief between the recipient and the requester that the requester α had a goal that he (the requester) will have a PWAG with respect to the intended actor γ about the goal ϕ of the REQUEST². It was shown in [Kumar *et al.*, 2000] that the recipient β and the intended actor γ never quantify into the beliefs of the requester α - meaning thereby that the requester α does not need to know who β and γ are. It was also shown how to construct a model in which a request can be performed even when the requester does not know the recipients as well as the intended actors of the request. Therefore, unlike the previous definitions in agent communication languages literature, this definition of REQUEST allows for the broadcasting of a request.

Theorem 2.1 Successful performance of the REQUEST communicative act establishes a mutual belief by default between the requester α and the intended recipient γ , about the requester's PWAG toward the intended recipient γ for doing the requested action. Formally,

$$\vdash_d (DONE(REQUEST \alpha \beta \gamma e a q t)) \Rightarrow \\ ({}^x MB \gamma^x \alpha ({}^y PWAG \alpha \gamma^y \phi q)) \\ \text{where, } \phi = (DONE \gamma a) \wedge \\ [PWAG \gamma \alpha (DONE \gamma a) (PWAG \alpha \gamma (DONE \gamma a) q)]$$

¹We do not discuss belief revision in this paper.

²In accordance with the first defeasible rule, we have simplified the temporal aspects of communicative act definitions.

Proof Sketch. γ is a subset of β , so the intention part of the definition of the REQUEST speech act is also valid by substituting β with γ . After making these substitutions, the proof is the same as the one provided in [Kumar *et al.*, 2002].

Definition 2.3 Inform

$$(INFORM \alpha \beta \gamma e p t) \equiv (ATTEMPT \alpha e \phi \psi t) \\ \text{where } \phi = [BMB \gamma \alpha p] \\ \text{and } \psi = [BMB \beta \alpha [GOAL \alpha (BEL \gamma [BEL \alpha p])]]$$

The goal of an INFORM is that the intended recipients γ come to believe that there is mutual belief between γ and the senders α that the proposition p is true. The intention of the INFORM is that the recipients (including the over-hearers) come to believe that there is mutual belief between the recipients and the senders that the senders had the goal that the intended recipients will believe that the senders belief p .

Theorem 2.2 Successful performance of the INFORM communicative act establishes a mutual belief by default between the sender α and the intended recipient γ , that the sender believes the informed proposition. Formally,
 $\vdash_d (DONE(INFORM \alpha \beta \gamma e p t)) \Rightarrow (MB \alpha \gamma (BEL \alpha p))$

Proof Sketch. Again, by substituting β with γ the proof of the theorem is the same as in [Kumar *et al.*, 2002].

Starting from the definition of INFORM and REQUEST, we can define some other communicative acts, modifying the definitions in [Kumar *et al.*, 2002]. One has to replace the sender x and the recipients y appropriately by the sender α , the recipients β , and the intended actors γ .

Definition 2.4 Agree

$$(AGREE \alpha \beta \gamma e a q t) \equiv (INFORM \alpha \beta \gamma e \phi t) \\ \text{where } \phi = (PWAG \alpha \gamma (DONE \alpha a) \\ (PWAG \gamma \alpha (DONE \alpha a) q))$$

The agreeing entities α inform the intended recipients γ that they have a PWAG with respect to γ to perform action a with respect to γ 's PWAG that α do a relative to q .

Definition 2.5 Refuse

$$(REFUSE \alpha \beta \gamma e a q t) \equiv (INFORM \alpha \beta \gamma e \phi t) \\ \text{where } \phi = (\neg \Box (PWAG \alpha \gamma (DONE \alpha a) \\ (PWAG \gamma \alpha (DONE \alpha a) q)))$$

The refusing entities α inform the intended recipients γ that they will never have the PWAG to perform action a with respect to γ 's PWAG that α do a relative to q . As in the individual case, the effect of a REFUSE is opposite to that of the AGREE.

In the next section, we discuss how these group communicative acts fit together in a group protocol that is described using the Joint Intention Theory [Kumar *et al.*, 2002]. In particular, we will see how to obtain the collective and distributive reading cases just by addressing the utterances to groups instead of single agents.

3 An Example: The Request for Action Protocol

For simplicity of exposition, let us consider the Request for Action protocol that establishes joint commitment(s) between the requester(s) and the requestee(s) to get an action done. One of the objectives of this protocol is to create a joint commitment (JPG) among the requester α and the group γ for

a	Distributive Request + Distributive Agree	<i>Distributive Reading</i>
b	Distributive Group Request + Distributive Agree	<i>Distributive Reading</i>
c	Collective Group Request + Collective Agree	<i>Collective Reading</i>
d	Collective Group Request + Distributive Agree to Group	<i>Collective Reading</i>

Figure 1: Distributive and Collective behaviors in the Request for Action protocol.

doing a specific action a . The discussion that follows applies equally well to other protocols (such as Contract-Net) that use communicative acts to create and discharge joint commitments. The simplest (and the most well-studied) case involves an individual REQUEST from one agent to another who then responds with either an AGREE or a REFUSE. It has been established in multi-agent literature that this sequence of communicative actions is sufficient to establish a joint commitment between the two agents for doing the requested action. The resulting joint commitment constrains the mental state and future behavior of both agents such that they work cooperatively as a team that is robust to a wide range of failures. However, this simple one-to-one request/accept protocol is insufficient for many real life scenarios that involve joint commitment not only between two individuals but between and among individuals and groups that may possibly be unknown at the start of the protocol. Furthermore, there may be over-hearers in any group communication that may affect the team creation process.

Even ignoring the effects of over-hearers, there are at least four different cases of team formation when the request for action protocol is generalized to account for group communication as listed in the above table (see Fig. 1). In the first case (a), individual requests are sent to multiple entities (that may be groups or individuals) for doing the same task. The requester has to know all the agents to whom it wants to request. Some of the recipients of that request may agree to the request, some may reject the request, and others may not respond at all. As a result of this communication, the requester forms pair-wise joint commitment with every entity that responded with an AGREE. The second case (b) is similar in that it also results in pair-wise joint commitments between the requester and each entity that agreed. However, it achieves the same effect by using a single group REQUEST that does not require a requester to be aware of the intended recipients of the request. This is very common in broadcast communications, for example, consider a request for helping with a programming task that has been posted on a notice board. Both the above two cases can be used to create competing teams, and this is not restricted to the request protocol. For example, consider a call for proposal from a funding agency for a competitive project where multiple contracts are awarded to different universities - each university has a pair-wise joint commitment with the funding agency for doing the research but at the same time they compete with each other to produce the best results for ongoing funding.

In the third case (c), the requester sends a collective REQUEST to a group which then replies collectively to that request. It is essentially communication from one entity (individual or group) to a group treated and addressed as an entity (such as an organization). An interesting aspect of this communication is to what extent the members of the requested organization are jointly committed towards the requester for doing the requested task. We speculate that this has to do with how that group was created in the first place. For example, if the bylaws in the articles of incorporation of that group require that the group makes its decisions by a majority vote but once a decision has been made then all group members are committed to that decision then it may very well be the case that each member of that organization is individually committed to seeing the success of the joint commitment.

In this case, we have collective commitment among the members of the group. This kind of commitment is the same that we can achieve in the last case (d), where a requester sends a REQUEST to a number of agents treated as a group and they reply individually with an AGREE addressed to the group. The main difference between the cases c and d is that while in the first case the message is addressed to an existing group (i.e., it has rules for team creation decision even if we are not aware of and we do not now who are the members of the group), in the last case the collective REQUEST followed by individual replies addressed to the group has the effect that the members of the group are bound together by their commitment and so a team is formed (we have a collaboration).

In the next sections, we will analyze two of the above cases. In particular, we will discuss the case b for distributive commitment (Sec. 3.1) and the case d for collaborative commitment (Sec. 3.2). Notice that protocols may also require that teams be disbanded (i.e., they have to discharge their commitment) after the achievement of the goal. The complete study of the protocol is left for future work.

3.1 Distributive Reading of Request Protocol

The first step of the Request protocol consists of a REQUEST sent from α to the group γ that establishes a MB about α 's PWAG towards each member of γ in a distributive way: ($REQUEST\ \alpha\ \beta\ \gamma\ e\ a\ q\ t$). In this case the sender α may not be aware of who are the members of γ , as in the case of posting a message on a advice board or, in the case of Grids, the agents may not be aware of who are the other agents connected to the grid. Notice that in this case, the intended actor of this communicative act is a single (possibly unknown) agent x .

A subset of the participants that we represent by the group γ_{agree} may agree to do the requested action and another subset of the participants represented by the group γ_{refuse} may refuse to the requested action. $\forall x\ \gamma_{agree}(x) \supset (DONE(AGREE\ x\ \beta\ \alpha\ e\ a\ \psi\ q\ t))$, and $\forall x\ \gamma_{refuse}(x) \supset (DONE(REFUSE\ x\ \beta\ \alpha\ e\ a\ \psi\ q\ t))$. In this case the communication is overheard by all the members of β . If we want private agreements, then we have: $\forall x\ \gamma_{agree}(x) \supset (DONE(AGREE\ x\ \alpha\ \alpha\ e\ a\ \psi\ q\ t))$. For example, if we want to design an auction protocol, the first method (with public communication) describes in a better way what happens during the protocol. Note that with the semantics of the FIPA

speech acts, it is not possible to model the same effects. In fact during an auction, all members of the group have to overhear all other bids in order to bid again.

The group property for the group γ_{agree} specifies the agents who chose to agree from a rational choice between agreeing and refusal. The group γ_{agree} is specified by noting that it consists of those agents who performed an AGREE within the specified timeout period after the original request was performed. In either case, the group predicate is evaluated retrospectively i.e. by looking backwards from a future point in time to determine which agents 'agreed'. We will specify γ_{agree} using the predicate $\gamma_{agree}(x)$ which is true if there exist events e' and e_1 such that the event sequence $e; e'; e_1$ was done earlier where e is the original REQUEST event and e_1 is an AGREE event performed by x .

$$\gamma_{agree}(x) \equiv \exists e', e_1 (EARLIER (DONE e; e'; (AGREE x \beta \alpha e_1 a q t_1)))$$

Similarly, the group property for γ_{refuse} specifies the agents who refused within the specified timeout period after the original REQUEST was performed.

According to these definitions of group predicates, an agent executing this protocol may not know all the agents who agree and all the agents who refuse until the deadline is reached, that is, when this protocol ends. In fact, the groups γ_{agree} and γ_{refuse} are dynamic groups because evaluating the respective group predicates at any time before the deadline gives only those agents that have agreed or refused up to that time. Until the deadline is reached, the membership of these groups increases whenever an agent performs an AGREE or a REFUSE communicative act.

The goal associated with the protocol is to form teams for doing an action a relative to some condition q . In other words, the goal is to establish pair-wise joint commitments between the initiator α and whoever agreed (i.e. every agent x who belongs to the group γ_{agree}) that x does the action a relative to the initiator's original PWAG. The existence of a JPG between two agents is a sufficient condition for the formation of a team with respect to that JPG [Levesque *et al.*, 1990]. Mutual beliefs in each other's PWAG toward the other to achieve a goal p is sufficient to establish a joint commitment to achieve p provided that there is a mutual belief that p has not already been achieved, and the PWAGs are interlocking (i.e., one PWAG is relative to the other) [Kumar *et al.*, 2002].

Theorem 3.1 $\vdash_d (DONE (REQUEST \alpha \beta \gamma e a q t); (AGREE \gamma_{agree} \beta \alpha e a q t')) \Rightarrow (JPG \alpha \gamma_{agree} (DONE \gamma_{agree} a))$

Proof Sketch. Using the definitions of the group communicative acts REQUEST and AGREE (definitions 2.2 and 2.4) and applying Theorems 2.2 and 2.1 under the assumptions of mutual belief between about each other's sincerity and the reliability of the communication channel, we establish that there is a mutual belief in each other's PWAG: $(^x MB \gamma^x \alpha (^y PWAG \alpha \gamma^y (DONE \gamma a))) \wedge \forall z \gamma_{agree}(z) \supset (MB z \alpha (PWAG z \alpha (DONE z a)))$.

In distributive reading, the effect of multiple single utterances is the same as that of making group utterances. Notice that here we are just considering the effects of group communication on the intended actors and not on the overhearers.

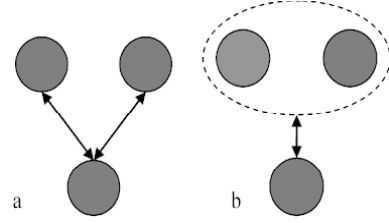


Figure 2: a. Case of Distributive Commitment; b. Case of Collective Commitment.

We will see in the next section that when we are dealing with collective reading, the effects of group communication are different from the one obtained with the above utterances.

3.2 Collective Reading of Request Protocol

In the case of collective reading, we need JPG and MB to be established among all the members of the group. It may be possible to achieve this with the exchange of pair-wise communicative acts resulting in $\mathcal{O}(n^2)$ messages. On the other hand, with messages addressed to groups via broadcast communication we would have just $\mathcal{O}(n)$ messages. Using the definition of group speech acts, it can be shown that the MB among the group is achieved. Notice that a collective request is more than sending individual requests to all the members of the group, i.e. a collective request implies mutual belief in the group that the intended actor of the request is the whole group, and so the action should be a joint action for the group. We claim that one cannot obtain a similar affect by just sending individual requests to the members.

In the case of collective reading, the team that is formed using group communication has different properties from the one discussed earlier. These different properties result from how the team is created. Note that one can also talk about groups in the case of distributive reading but in that case the requester and the agents who agree have pair-wise joint commitment (see Fig. 2 case a). However, in the case of a collective reading of the Request protocol, the following property holds in order to establish $(JPG \gamma (DONE \gamma a))$ for the group γ : $\forall x \gamma(x) \supset (MB \gamma (PWAG x \gamma p)) \wedge (MB \gamma (PWAG \gamma x p))$ where there is explicitly stated that the member of the team γ has beliefs about the others agents or about the team as an entity. This demonstrates that in the case of a collective reading, the effects of multiple single utterances are different from group utterances.

So in the case of collective REQUEST, the REQUEST is addressed to a group and not to a single agent: $(REQUEST \alpha \beta \gamma e a q t)$. Then, each member of the group will send an AGREE (or a REFUSE) to α plus the group itself (i.e., $\gamma_{agree+\alpha}$): $\forall x \gamma_{agree}(x) \supset (AGREE x \beta \gamma_{agree+\alpha} e a \psi q t)$. Also, in this case each member of the group γ_{agree} may be not aware of who are the other members.

Theorem 3.2 *Successful performance of a REQUEST addressed to the group for collective reading, followed by AGREES addressed to the requester and the group establishes a JPG by default among the members of the agree group and*

the requester. Formally,

$$\begin{aligned} &\vdash_d (DONE (REQUEST \alpha \beta \gamma e a q t); \\ &\quad ({}^x AGREE \gamma_{agree}^x \beta \gamma_{agree+\alpha} e a q t')) \\ &\Rightarrow (JPG \gamma_{agree+\alpha} (DONE \gamma_{agree} a)) \end{aligned}$$

Proof Sketch. First, consider agent α who sends a REQUEST to γ as a group. Successful performance of the REQUEST communicative act establishes a mutual belief by default between α and γ about α 's PWAG towards γ for doing the requested action: $({}^x MB \alpha \gamma_{agree}^x ({}^y PWAG \alpha \gamma_{agree}^y ({}^y DONE \gamma_{agree}^y a) q))$. Then, the agents in γ_{agree} reply with a AGREE to the group $\gamma_{agree+\alpha}$. From the definition of the AGREE communicative act and Theorem 2.2 we have that successful performance of the AGREE communicative act establishes a mutual belief by default between α and γ_{agree} about γ_{agree} 's PWAG toward α (i.e., $\forall x \gamma_{agree}(x) \supset (MB x \gamma_{agree+\alpha} (PWAG x \gamma_{agree+\alpha} (DONE \gamma_{agree} a) q)) \equiv \forall x \gamma_{agree}(x) \supset ({}^y MB \gamma_{agree+\alpha}^y (PWAG x \gamma_{agree+\alpha} (DONE \gamma_{agree} a) q)) \equiv ({}^y MB \gamma_{agree+\alpha}^y ({}^x PWAG \gamma_{agree}^x \alpha (DONE \gamma_{agree} a) q))$). Using the definitions of the group communicative acts and the assumptions of mutual belief about each other's sincerity and the reliability of the communication channel, we establish that there is a mutual belief in each other's PWAG. Thereafter, the property for JPG for a collaborative reading is satisfied. We note that the key difference between a collective reading and distributive reading lies in the AGREES sent to both the requester and the other agents as a group. What we argue is that one cannot achieve the same results sending individual utterances.

Theorem 3.3 *Successful performance of a REQUEST addressed to each individual in the group, followed by an AGREE to each individual in the group establishes a JPG by default among the members of the AGREE group but does not constitute a collective reading. Formally,*

$$\begin{aligned} &\forall_d (DONE (REQUEST \alpha \beta \gamma e a q t); \\ &\quad ({}^{xy} AGREE \gamma_{agree}^x \beta \gamma_{agree+\alpha}^y e a q t')) \\ &\Rightarrow (JPG \gamma_{agree+\alpha} (DONE \gamma_{agree} a)) \end{aligned}$$

Proof Sketch. In order to show this is not a theorem, we provide a counterexample. Consider three agents (namely the requester a , and $b, c \in \gamma$) in group γ . After the REQUEST communicative act has been performed, we have that $\forall x \gamma(x) \supset (MB \alpha x (PWAG \alpha x p q))$. So, for example, in this case we don't have that $(MB a b (PWAG a c p q))$ holds, because the REQUEST is addressed to individuals and b cannot see the message addressed to c .

4 Discussion

Group communication is common in collaborative applications of all kind. There are a variety of distributed applications that provides group communication. An infrastructure supporting group communication can be based on many different approaches and technologies, whose features vary significantly. Some of them rely on message passing systems and are implemented as webs of point-to-point links or through multicast IP. Another common approach for dealing with group communication is based on distributed

event systems or message queue-based infrastructure for publish/subscribe systems. Publish/subscribe is a pattern often used in systems for human-to-human interaction, for example, bulletin boards, newsgroups and their more recent Internet incarnations such as blogs, group-chat rooms, and so on. While their technical characteristic vary among each other, they share the same objective of allowing direct conversations among any partners while simultaneously making exchanged message visible to a large number of destinations, normally unknown to the conversing parties. In whatever way group communication systems are implemented, what it is missing is a clear semantics for the communication. In this paper we have shown that by using group communicative acts in the context of group protocols we can obtain two different types of commitments among group members. In particular, just the fact that an agent addresses an utterance to the whole group instead of to each single member of the group may imply different properties for the group itself. In the future we would like to expand our work by reasoning about the effects that group communicative acts may have on the other recipients, for example overhearers, and their role in the behavior of the group.

References

- [Clark and Carlson, 1982] H. H. Clark and T. B. Carlson. Speech acts and hearers' beliefs. In *Mutual Knowledge*, pages 1–36. Academic Press, London, 1982.
- [Cohen and Levesque, 1990] Philip R. Cohen and Hector J. Levesque. Performatives in a rationally based speech act theory. In *Proceedings of the 28th ACL conference*, pages 79–88, 1990.
- [De Roure et al., 2003] D. De Roure, M.A. Baker, N.R. Jennings, and N.R. Shadbolt. The evolution of the grid. In *Grid computing - making the global infrastructure a reality*, pages 65–100. John Wiley and Sons Ltd, 2003.
- [Foster et al., 2001] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid: Enabling scalable virtual organizations. *International Journal Supercomputer Applications*, 15(3), 2001.
- [Jennings, 2001] Nicholas R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.
- [Kumar et al., 2000] S. Kumar, M. J. Huber, D. McGee, P. R. Cohen, and H. J. Levesque. Semantics of agent communication languages for group interaction. In *Proceedings of the 17th International Conference on Artificial Intelligence*, pages 42–47, 2000.
- [Kumar et al., 2002] Sanjeev Kumar, Marcus J. Huber, Philip R. Cohen, and David R. McGee. Toward a formalism for conversation protocols using joint intention theory. *Computational Intelligence*, 18(2):174–174, 2002.
- [Levesque et al., 1990] Hector J. Levesque, Philip R. Cohen, and José H. T. Nunes. On acting together. In *Proceedings of the 8th National Conference on Artificial Intelligence*, AAAI, pages 94–99, 1990.

Learning Payoff Functions in Infinite Games

Yevgeniy Vorobeychik, Michael P. Wellman, and Satinder Singh

University of Michigan
Artificial Intelligence Laboratory
1101 Beal Avenue
Ann Arbor, MI 48109-2110 USA
{ yvorobey, wellman, baveja }@umich.edu

Abstract

We consider a class of games with real-valued strategies and payoff information available only in the form of data from a given sample of strategy profiles. Solving such games with respect to the underlying strategy space requires generalizing from the data to a complete payoff-function representation. We address payoff-function learning as a standard regression problem, with provision for capturing known structure (symmetry) in the multiagent environment. To measure learning performance, we consider the relative utility of prescribed strategies, rather than the accuracy of payoff functions per se. We demonstrate our approach and evaluate its effectiveness on two examples: a two-player version of the first-price sealed-bid auction (with known analytical form), and a five-player market-based scheduling game (with no known solution).

1 Introduction

Game-theoretic analysis typically begins with a complete description of strategic interactions, that is, *the game*. We consider the prior question of determining what the game actually is, given a database of game experience rather than any direct specification. This is one possible target of *learning* applied to games [Shoham *et al.*, 2003]. When agents have few available actions and outcomes are deterministic, the game can be identified through systematic exploration. For instance, we can ask the agents to play each strategy profile in the entire joint strategy set and record the payoffs for each. If the joint action space is small enough, limited nondeterminism can be handled by sampling. Coordinating exploration of the joint set does pose difficult issues. Brafman and Tennenholtz, for example, address these carefully for the case of common-interest stochastic games [Brafman and Tennenholtz, 2003], as well as the general problem of maintaining an equilibrium among learning algorithms [Brafman and Tennenholtz, 2004].

Further difficulties are posed by intractably large (or infinite) strategy sets. We can make this problem tractable by reducing the number of profiles that agents are allowed to play, but this comes at the cost of transforming the game of interest into a different game entirely. Instead, we seek to

identify the full game (or at least a less restrictive game) from limited data, entailing some generalization from observed instances. Approximating payoff functions using supervised learning (regression) methods allows us to deal with continuous agent strategy sets, providing a payoff for an arbitrary strategy profile. In so doing, we adopt functional forms consistent with prior knowledge about the game, and also admit biases toward forms facilitating subsequent game analysis (e.g., equilibrium calculation).

In this paper, we present our first investigation of approximating payoff functions, employing regression to low-degree polynomials. We explore two example games, both with incomplete information and real-valued actions. First is the standard first-price sealed bid auction, with two players and symmetric value distributions. The solution to this game is well-known [Krishna, 2002], and its availability in analytical form proves useful for benchmarking our learning approach. Our second example is a five-player market-based scheduling game [Reeves *et al.*, 2005], where time slots are allocated by simultaneous ascending auctions [Milgrom, 2000]. This game has no known solution, though previous work has identified equilibria on discretized subsets of the strategy space.

2 Preliminaries

2.1 Notation

A generic normal form game is formally expressed as $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$, where I refers to the set of players and $m = |I|$ is the number of players. S_i is the set of strategies available to player $i \in I$, and the set $\Delta(S_i)$ is the simplex of mixed strategies over S_i . Finally, $u_i(s) : S_1 \times \dots \times S_m \rightarrow \mathcal{R}$ is the payoff function of player i when all players jointly play $s = (s_1, \dots, s_m)$, with each $s_j \in S_j$. As is common, we assume von Neumann-Morgenstern utility, allowing an agent i 's payoff for a particular mixed strategy profile to be $u_i(\sigma) = \sum_{s \in S} [\sigma_1(s_1) \dots \sigma_m(s_m)] u_i(s)$, where $\sigma_j : S_j \rightarrow [0, 1]$ is a mixed strategy of player j , assigning a probability to each pure strategy $s_j \in S_j$ such that all probabilities over the agent's strategy set add to 1 (i.e., $\sigma_j \in \Delta(S_j)$).

It will often be convenient to refer to the strategy (pure or mixed) of player i separately from that of the remaining players. To accommodate this, we use s_{-i} to denote the joint strategy of all players other than player i .

2.2 Nash Equilibrium

In this paper, we are concerned with one-shot normal-form games, in which players make decisions simultaneously and accrue payoffs, upon which the game ends. This single-shot nature may seem to preclude learning from experience, but in fact repeated episodes are allowed, as long as actions cannot affect future opportunities, or condition future strategies. Game payoff data may also be obtained from observations of other agents playing the game, or from simulations of hypothetical runs of the game. In any of these cases, learning is relevant despite the fact that the game is to be played only once.

Faced with a one-shot game, an agent would ideally play its best strategy given those played by the other agents. A configuration where all agents play strategies that are best responses to the others constitutes a *Nash equilibrium*.

Definition 1 A strategy profile $s = (s_1, \dots, s_m)$ constitutes a (pure-strategy) Nash equilibrium of game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ if for every $i \in I$, $s'_i \in S_i$, $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$.

A similar definition applies when mixed strategies are allowed.

Definition 2 A strategy profile $\sigma = (\sigma_1, \dots, \sigma_m)$ constitutes a mixed-strategy Nash equilibrium of game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ if for every $i \in I$, $\sigma'_i \in \Delta(S_i)$, $u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma'_i, \sigma_{-i})$.

In this study we devote particular attention to games that exhibit symmetry with respect to payoffs.

Definition 3 A game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ is symmetric if $\forall i, j \in I$, (a) $S_i = S_j$ and (b) $u_i(s_i, s_{-i}) = u_j(s_j, s_{-j})$ whenever $s_i = s_j$ and $s_{-i} = s_{-j}$.

Symmetric games have relatively compact descriptions and may present associated computational advantages. Given a symmetric game, we may focus on the subclass of symmetric equilibria, which are arguably most natural [Kreps, 1990], and avoid the need to coordinate on roles. In fairly general settings, symmetric games do possess symmetric equilibria [Nash, 1951].

3 Payoff Function Approximation

3.1 Problem Definition

We are given a set of data points (s, v) , each describing an instance where agents played strategy profile s and realized value $v = (v_1, \dots, v_m)$. For deterministic games of complete information, v is simply u . With incomplete information or stochastic outcomes, v is a random variable, more specifically an independent draw from a distribution function of s , with expected value $u(s)$.

The *payoff function approximation task* is to select a function $\hat{u}$ from a candidate set $\mathcal{U}$ minimizing some measure of deviation from the true payoff function u . Because the true function u is unknown, of course, we must base our selection on evidence provided by the given data points.

Our goal in approximating payoff functions is typically not predicting payoffs themselves, but rather in assessing strategic behavior. Therefore, for assessing our results, we measure

approximation quality not directly in terms of a distance between $\hat{u}$ and u , but rather in terms of the *strategies dictated by $\hat{u}$* evaluated with respect to u . For this we appeal to the notion of approximate Nash equilibrium.

Definition 4 A strategy profile $\sigma = (\sigma_1, \dots, \sigma_m)$ constitutes an ϵ -Nash equilibrium of game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$ if for every $i \in I$, $\sigma'_i \in \Delta(S_i)$, $u_i(\sigma_i, \sigma_{-i}) + \epsilon \geq u_i(\sigma'_i, \sigma_{-i})$.

We propose using ϵ in the above definition as a measure of approximation error of $\hat{u}$, and employ it in evaluating our learning methods. When u is known, we can compute ϵ in a straightforward manner. Let s_i^* denote i 's *best-response correspondence*, defined by $s_i^*(\sigma_{-i}) = \{x : x \in \arg \max_{s_i} u_i(s_i, \sigma_{-i})\}$. For clarity of exposition, we take $s_i^*(\sigma_{-i})$ to be single-valued. Let $\hat{\sigma}$ be a solution (e.g., a Nash equilibrium) of game $[I, \{\Delta(S_i)\}, \{\hat{u}_i(s)\}]$. Then $\hat{\sigma}$ is an ϵ -Nash equilibrium of the true game $[I, \{\Delta(S_i)\}, \{u_i(s)\}]$, for $\epsilon = \max_{i \in I} [u_i(s_i^*(\hat{\sigma}_{-i}), \hat{\sigma}_{-i}) - u_i(\hat{\sigma}_i, \hat{\sigma}_{-i})]$.

Since in general u will either be unknown or not amenable to this analysis, we developed a method for estimating ϵ from data. We will describe it in some detail below.

For the remainder of this report, we focus on a special case of the general problem, where action sets are real-valued intervals, $S_i = [0, 1]$. Moreover, we restrict attention to symmetric games and further limit the number of variables in payoff-function hypotheses by using some form of aggregation of other agents' actions.¹ The assumption of symmetry allows us to adopt the convention for the remainder of the paper that payoff $u(s_i, s_{-i})$ is to the agent playing s_i .

3.2 Polynomial Regression

One class of models we consider are the *n th-degree separable polynomials*:

$$u(s_i, \phi(s_{-i})) = a_n s_i^n + \dots + a_1 s_i + b_n \phi^n(s_{-i}) + \dots + b_1 \phi(s_{-i}) + d, \quad (1)$$

where $\phi(s_{-i})$ represents some aggregation of the strategies played by agents other than i . For two-player games, ϕ is simply the identity function. We refer to polynomials of the form (1) as *separable*, since they lack terms combining s_i and s_{-i} . We also consider models with such terms, for example, the *non-separable quadratic*:

$$u(s_i, \phi(s_{-i})) = a_2 s_i^2 + a_1 s_i + b_2 \phi^2(s_{-i}) + b_1 \phi(s_{-i}) + c s_i \phi(s_{-i}) + d. \quad (2)$$

Note that (2) and (1) coincide in the case $n = 2$ and $c = 0$. In the experiments described below, we employ a simpler version of non-separable quadratic that takes $b_1 = b_2 = 0$.

One advantage of the quadratic form is that we can analytically solve for Nash equilibrium. Given a general non-separable quadratic (2), the necessary first-order condition for an interior solution is $s_i = -(a_1 + c\phi(s_{-i}))/2a_2$. This reduces to $s_i = -a_1/2a_2$ in the separable case. For the non-separable case with *additive aggregation*, $\phi_{\text{sum}}(s_{-i}) =$

¹ Although none of these restrictions are inherent in the approach, one must of course recognize the tradeoffs in complexity of the hypothesis space and generalization performance.

$\sum_{j \neq i} s_j$, we can derive an explicit first-order condition for symmetric equilibrium: $s_i = -a_1/(2a_2 + (m-1)c)$.

While a pure-strategy equilibrium will necessarily exist for any separable polynomial model, it is only guaranteed to exist in the non-separable case when the learned quadratic is concave. In the experiments that follow, when the learned non-separable quadratic does not have a pure Nash equilibrium, we generate an arbitrary symmetric pure profile as the approximate Nash equilibrium.

Another difficulty arises when a polynomial of a degree higher than three has more than one Nash equilibrium. In such a case we select an equilibrium arbitrarily.

3.3 Local Regression

In addition to polynomial models, we explored learning using two local regression methods: locally weighted average and locally weighted quadratic regression [Atkeson *et al.*, 1997]. Unlike model-based methods such as polynomial regression, local methods do not attempt to infer model coefficients from data. Instead, these methods weigh the training data points by distance from the query point and estimate the answer—in our case, the payoff at the strategy profile point—using some function of the weighted data set. We used a Gaussian weight function: $w = e^{-d^2}$, where d is the distance of the training data point from the query point and w is the weight that is assigned to that training point.

In the case of locally weighted average, we simply take the weighted average of the payoffs of the training data points as our payoff at an arbitrary strategy profile. Locally weighted quadratic regression, on the other hand, fits a quadratic regression to the weighted data set for each query point.

3.4 Support Vector Machine Regression

The third category of learning methods we used was Support Vector Machines (SVMs). For details regarding this learning method, we refer an interested reader to [Vapnik, 1995]. In our experiments, we used *SVM light* package [Joachims, 1999], which is an open-source implementation of SVM classification and regression algorithms.

3.5 Finding Mixed Strategy Equilibria

In the case of polynomial regression, we were able to find either analytic or simple and robust numeric methods for computing pure Nash equilibria. With local regression and SVM learning we are not so fortunate, as we do not have access to a closed-form description of the function we are learning. Furthermore, we are often interested in mixed strategy approximate equilibria, and our polynomial models and solution methods yield pure strategy equilibria.

When a particular learned model is not amenable to a closed-form solution, we can approximate the *learned* game with a finite strategy grid and find a mixed-strategy equilibrium of the resulting finite game using a general-purpose finite-game solver. We employed replicator dynamics [Fudenberg and Levine, 1998], which searches for a symmetric mixed equilibrium using an iterative evolutionary algorithm. We treat the result after a fixed number of iterations as an approximate Nash equilibrium of the learned game.

3.6 Strategy Aggregation

As noted above, we consider payoff functions on two-dimensional strategy profiles in the form $u(s_i, s_{-i}) = f(s_i, \phi(s_{-i}))$. As long as $\phi(s_{-i})$ is invariant under different permutations of the same strategies in s_{-i} , the payoff function is symmetric. Since the actual payoff functions for our example games are also known to be symmetric, we constrain that $\phi(s_{-i})$ preserve the symmetry of the underlying game.

In our experiments, we compared three variants of $\phi(s_{-i})$. First and most compact is the simple sum, $\phi_{sum}(s_{-i})$. Second is the ordered pair (ϕ_{sum}, ϕ_{ss}) , where $\phi_{ss}(s_{-i}) = \sum_{j \neq i} (s_j)^2$. The third variant, $\phi_{identity}(s_{-i}) = s_{-i}$, simply takes the strategies in their direct, unaggregated form. To enforce the symmetry requirement in this last case, we sort the strategies in s_{-i} .

4 First-Price Sealed-Bid Auction

In the standard first-price sealed-bid (FPSB) auction game [Krishna, 2002], agents have private valuations for the good for sale, and simultaneously choose a bid price representing their offer to purchase the good. The bidder naming the highest price gets the good and pays the offered price. Other agents receive and pay nothing. In the classic setup first analyzed by Vickrey [1961], agents have identical valuation distributions, uniform on $[0, 1]$, and these distributions are common knowledge. The unique (Bayesian) Nash equilibrium of this game is for agent i to bid $\frac{m-1}{m} x_i$, where x_i is i 's valuation for the good.

Note that strategies in this game (and generally for games of incomplete information), $b_i : [0, 1] \rightarrow [0, 1]$, are functions of the agent's private information. We consider a restricted case, where bid functions are constrained to the form $b_i(x_i) = k_i x_i$, $k_i \in [0, 1]$. This constraint transforms the action space to a real interval, corresponding to choice of parameter k_i . We can easily see that the restricted strategy space includes the known equilibrium of the full game, with $s_i = k_i = \frac{m-1}{m}$ for all i , which is also an equilibrium of the restricted game in which agents are constrained to strategies of the given form.

We further focus on the special case $m = 2$, with corresponding equilibrium at $s_1 = s_2 = 1/2$. For the two-player FPSB, we can also derive a closed-form description of the actual expected payoff function:

$$u(s_1, s_2) = \begin{cases} 0.25 & \text{if } s_1 = s_2 = 0, \\ \frac{(s_1 - 1)[(s_2)^2 - 3(s_1)^2]}{6(s_1)^2} & \text{if } s_1 \geq s_2, \\ \frac{s_1(1 - s_1)}{3s_2} & \text{otherwise.} \end{cases} \quad (3)$$

The availability of known solutions for this example facilitates analysis of our learning approach. Our results are summarized in Figure 1. For each of our methods (classes of functional forms), we measured average ϵ for varying training set sizes. For instance, to evaluate the performance of separable quadratic approximation with training size N , we independently draw N strategies, $\{s^1, \dots, s^N\}$, uniformly on $[0, 1]$. The corresponding training set comprises $O(N^2)$ points: $((s^i, s^j), u(s^i, s^j))$, for $i, j \in \{1, \dots, N\}$, with u as given by (3). We find the best separable quadratic fit $\hat{u}$ to

these points, and find a Nash equilibrium corresponding to $\hat{u}$. We then calculate the least ϵ for which this strategy profile is an ϵ -Nash equilibrium with respect to the actual payoff function u . We repeat this process 200 times, averaging the results over strategy draws, to obtain each value plotted in Figure 1.

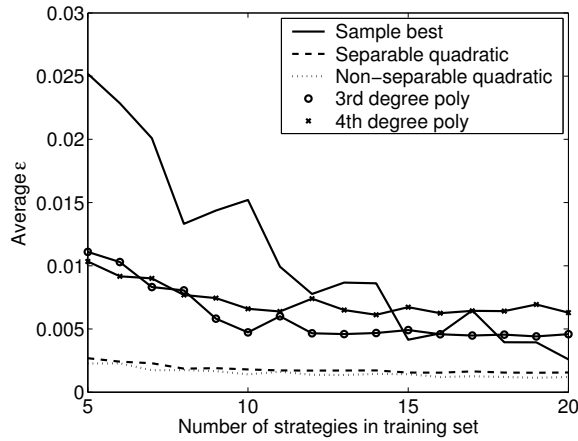


Figure 1: Epsilon versus number of training strategy points for different functional forms.

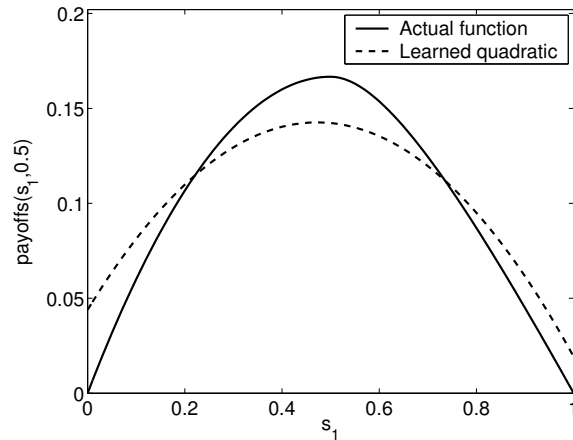


Figure 2: Learned and actual payoff function when the other agent plays 0.5. The learned function is the separable quadratic, for a particular sample with $N = 5$.

As we can see, both second-degree polynomial forms we tried do quite well on this game. For $N < 20$, quadratic regression outperforms the model labeled “sample best”, in which the payoff function is approximated by the discrete training set directly. The derived equilibrium in this model is simply a Nash equilibrium over the discrete strategies in the training set. At first, the success of the quadratic model may be surprising, since the actual payoff function (3) is only piecewise differentiable and has a point of discontinuity. However, as we can see from Figure 2, it appears quite smooth and well approximated by a quadratic polynomial. The higher-degree polynomials apparently overfit the data, as

indicated by their inferior learning performance displayed in this game.

The results of this game provide an optimistic view of how well regression might be expected to perform compared to discretization. This game is quite easy for learning since the underlying payoff function is well captured by our lower-degree model. Moreover, our experimental setup eliminated the issue of noisy payoff observations, by employing the actual expected payoffs for selected strategies.

5 Market-Based Scheduling Game

The second game we investigate presents a significantly more difficult learning challenge. It is a five-player symmetric game, with no analytic characterization, and no (theoretically) known solution. The game hinges on incomplete information, and training data is available only from a simulator that samples from the underlying distribution.

The game is based on a market-based scheduling scenario [Reeves *et al.*, 2005], where agents bid in simultaneous auctions for time-indexed resources necessary to perform their given jobs. Agents have private information about their job lengths, and values for completing their jobs by various deadlines. Note that the full space of strategies is quite complex: it is dependent on multi-dimensional private information about preferences as well as price histories for all the time slots. As in the FPSB example, we transform this policy space to the real interval by constraining strategies to a parametrized form. In particular, we start from a simple myopic policy—*straightforward bidding* [Milgrom, 2000], and modify it by a scalar parameter (called “sunk awareness”, and denoted by k) that controls the agent’s tendency to stick with slots that it is currently winning. Although the details and motivation for sunk awareness are inessential to the current study, we note that $k \in [0, 1]$, and that the optimal setting of k involves tradeoffs, generally dependent on other agents’ behavior.

To investigate learning for this game, we collected data for all strategy profiles over the discrete set of values $k \in \{0, 0.05, \dots, 1\}$. Accounting for symmetry, this represents 53,130 distinct strategy profiles. For evaluation purposes, we treat the sample averages for each discrete profile as the true expected payoffs on this grid.

The previous empirical study of this game by Reeves *et al.* [2005] estimated the payoff function over a discrete grid of profiles assembled from the strategies $\{0.8, 0.85, 0.9, 0.95, 1\}$, computing an approximate Nash equilibrium using replicator dynamics. We therefore generated a training set based on the data for these strategies (300000 samples per profile), regressed to the quadratic forms, and calculated empirical ϵ values with respect to the entire data set by computing the maximum benefit from deviation within the data: $\epsilon_{emp} = \max_{i \in I} \max_{s_i \in S_i} [u_i(s_i, \hat{s}_{-i}) - u_i(\hat{s})]$, where S_i is the strategy set of player i represented within the data set. Since the game is symmetric, the maximum over the players can be dropped, and all the agent strategy sets are identical.

From the results presented in Table 1, we see that the Nash equilibria for the learned functions are quite close to that produced by replicator dynamics, but with ϵ values quite a bit

lower. (Since 0.876 is not a grid point, we determined its ϵ post hoc, by running further profile simulations with all agents playing 0.876, and where one agent deviates to any of the strategies in $\{0, 0.05, \dots, 1\}$.)

Method	Equilibrium s_i	ϵ
Separable quadratic	0.876	0.0027
Non-separable quadratic	0.876	0.0027
Replicator Dynamics	(0,0.94,0.06,0,0)	0.0238

Table 1: Values of ϵ for the symmetric pure-strategy equilibria of games defined by different payoff function approximation methods. The quadratic models were trained on profiles confined to strategies in $\{0.8, 0.85, 0.9, 0.95, 1\}$.

In a more comprehensive trial, we collected 2.2 million additional samples per profile, and ran our learning algorithms on 100 training sets, each uniformly randomly selected from the discrete grid $\{0, 0.05, \dots, 1\}$. Each training set included profiles generated from between five and ten of the twenty-one agent strategies on the grid. Since in this case the profile of interest does not typically appear in the complete data set, we developed a method for estimating ϵ for pure symmetric approximate equilibria in symmetric games based on a mixture of neighbor strategies that do appear in the test set. Let us designate the pure symmetric equilibrium strategy of the approximated game by $\hat{s}$. We first determine the closest neighbors to $\hat{s}$ in the symmetric strategy set S represented within the data. Let these neighbors be denoted by s' and s'' . We define a mixed strategy α over support $\{s', s''\}$ as the probability of playing s' , computed based on the relative distance of $\hat{s}$ from its neighbors: $\alpha = 1 - |\hat{s} - s'|/|s' - s''|$. Note that symmetry allows a more compact representation of a payoff function if agents other than i have a choice of only two strategies. Thus, we define $U(s_i, j)$ as the payoff to a (symmetric) player for playing strategy $s_i \in S$ when j other agents play strategy s' . If $m - 1$ agents each independently choose whether to play s' with probability α , then the probability that exactly j will choose s' is given by

$$\Pr(\alpha, j) = \binom{m-1}{j} \alpha^j (1 - \alpha)^{m-1-j}.$$

We can thus approximate ϵ of the mixed strategy α by

$$\max_{s_i \in S} \sum_{j=0}^{m-1} \Pr(\alpha, j) (U(s_i, j) - \alpha U(s', j) - (1 - \alpha) U(s'', j)).$$

Using this method of estimating ϵ on the complete data set, we compared results from polynomial regression to the method which simply selects from the training set the pure strategy profile with the smallest value of ϵ . We refer to this method as “sample best”, differentiating between the case where we only consider symmetric pure profiles (labeled “sample best (symmetric)”) and all pure profiles (labeled “sample best (all)”).²

²It is interesting to observe in Figures 3 and 4 that when we restrict the search for a best pure strategy profile to symmetric profiles, we on average do better in terms of ϵ than when this restriction is not imposed.

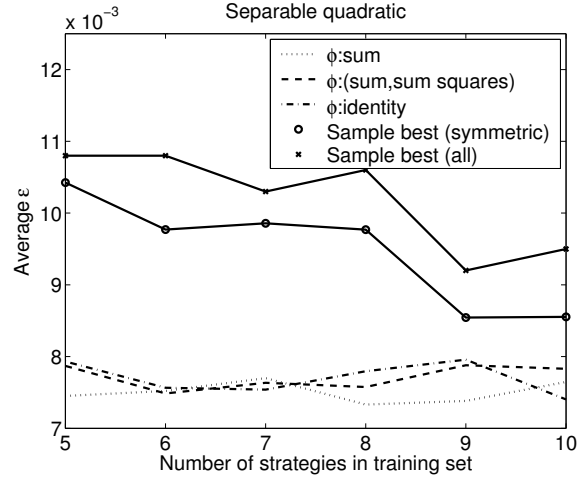


Figure 3: Effectiveness of learning a separable quadratic model with different forms of $\phi(s_i)$.

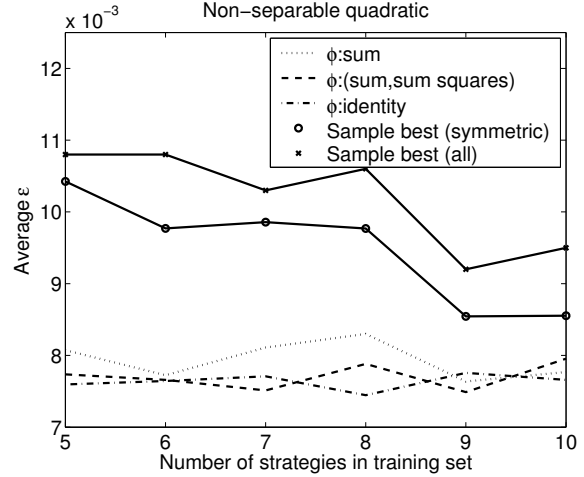


Figure 4: Effectiveness of learning a non-separable quadratic model with different forms of $\phi(s_i)$.

From Figure 3 we see that regression to a separable quadratic produces a considerably better approximate equilibrium when the size of the training set is relatively small. Figure 4 shows that the non-separable quadratic performs similarly. The results appear relatively insensitive to the degree of aggregation applied to the representation of other agents’ strategies.

The polynomial regression methods we employed yield pure-strategy Nash equilibria. We further evaluated four methods that generally produce mixed-strategy equilibria: two local regression learning methods, SVM with a Gaussian radial basis kernel, and direct estimation using the training data. As discussed above, we computed mixed strategy equilibria by applying replicator dynamics to discrete approximations of the learned payoff functions.³ Since we ensure that

³In the case of direct estimation from training data, the data itself

the support of any mixed strategy equilibrium produced by these methods is in the complete data set, we can compute ϵ of the equilibria directly.

As we can see in Figure 5, locally weighted average method appears to work better than the other three for most data sets that include between five and ten strategies. Additionally, locally weighted regression performs better than replicator dynamics on four of the six data set sizes we considered, and SVM consistently beats replicator dynamics for all six data set sizes.⁴

It is somewhat surprising to see how irregular our results appear for the local regression methods. We cannot explain this irregularity, although of course there is no reason for us to expect otherwise: even though increasing the size of the training data set may improve the quality of fit, improvement in quality of equilibrium approximation does not necessarily follow.

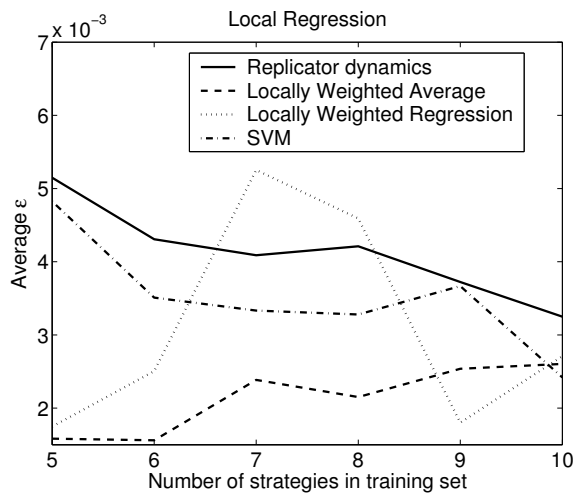


Figure 5: Effectiveness of learning local and SVM regression to estimate mixed strategy symmetric equilibria, with $\phi(s_{-i}) = (\phi_{sum}, \phi_{ss})$.

6 Conclusion

While there has been much work in game theory attempting to solve particular games defined by some payoff functions, little attention has been given to approximating such functions from data. This work addresses the question of payoff function approximation by introducing regression learning techniques and applying them to representative games of interest. Our results in both the FPSB and market-based scheduling games suggest that when data is sparse, such methods

was used as input to the replicator dynamics algorithm. For the other three methods we used a fixed ten-strategy grid as the discretized approximation of the learned game.

⁴Note that we do not compare these results to those for the polynomial regression methods. Given noise in the data set, mixed-strategy profiles with larger supports may exhibit lower ϵ simply due to the smoothing effect of the mixtures.

can provide better approximations of the underlying game—at least in terms of ϵ -Nash equilibria—than discrete approximations using the same data set.

Regression or other generalization methods offer the potential to extend game-theoretic analysis to strategy spaces (even infinite sets) beyond directly available experience. By selecting target functions that support tractable equilibrium calculations, we render such analysis analytically convenient. By adopting functional forms that capture known structure of the payoff function (e.g., symmetry), we facilitate learnability. This study provides initial evidence that we can sometimes find models serving all these criteria.

In future work we expect to apply some of the methods developed here to other challenging domains.

References

- [Atkeson *et al.*, 1997] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.
- [Brafman and Tennenholtz, 2003] Ronen I. Brafman and Moshe Tennenholtz. Learning to coordinate efficiently: A model-based approach. *Journal of Artificial Intelligence Research*, 19:11–23, 2003.
- [Brafman and Tennenholtz, 2004] Ronen I. Brafman and Moshe Tennenholtz. Efficient learning equilibrium. *Artificial Intelligence*, 159:27–47, 2004.
- [Fudenberg and Levine, 1998] D. Fudenberg and D. Levine. *The Theory of Learning in Games*. MIT Press, 1998.
- [Joachims, 1999] Thorsten Joachims. Making large-scale SVM learning practical. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1999.
- [Kreps, 1990] David M. Kreps. *Game Theory and Economic Modelling*. Oxford University Press, 1990.
- [Krishna, 2002] Vijay Krishna. *Auction Theory*. Academic Press, 2002.
- [Milgrom, 2000] Paul Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Journal of Political Economy*, 108:245–272, 2000.
- [Nash, 1951] John Nash. Non-cooperative games. *Annals of Mathematics*, 54:286–295, 1951.
- [Reeves *et al.*, 2005] Daniel M. Reeves, Michael P. Wellman, Jeffrey K. MacKie-Mason, and Anna Osepayshvili. Exploring bidding strategies for market-based scheduling. *Decision Support Systems*, 39:67–85, 2005.
- [Shoham *et al.*, 2003] Yoav Shoham, Rob Powers, and Trond Grenager. Multi-agent reinforcement learning: A critical survey. Technical report, Stanford University, 2003.
- [Vapnik, 1995] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.
- [Vickrey, 1961] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

A Logical Model of Nash Bargaining Solution

Dongmo Zhang

Intelligent Systems Laboratory

School of Computing and Information Technology

University of Western Sydney, Australia

dongmo@cit.uws.edu.au

Abstract

This paper presents a logical extension of Nash's Cooperative Bargaining Theory. We introduce a concept of entrenchment measurement, which maps propositions to real numbers, as a vehicle to represent agent's belief states and attitudes towards bargaining situations. We show that Nash's bargaining solution can be restated in terms of bargainers belief states. Negotiable items, bargaining outcomes and conflicting arguments can then be explicitly expressed in propositional logic meanwhile Nash's numerical solution to bargaining problem is still applicable.

1 Introduction

Negotiation or bargaining is a process to settle disputes and reach mutually beneficial agreements. Typical situations of negotiation are characterized by two or more agents who have common interests in cooperating but have conflicting interests in the way of doing so. The outcomes of negotiation depend on agents' attitudes towards their bargaining items and their expectations from the negotiation. The representation of bargainer's attitudes in game theory is implicit via a pair of von Neumann-Morgenstern utility functions [Nash 1950]. The conflicting of interests between bargainers is then implied by certain mathematical conditions of the utility functions. However, agents' attitudes can be more explicitly described in terms of logic. Negotiable items, conflicting claims, arguments of disputation and negotiation protocols are all expressible by logical statements [Sycara 1990][Sierra *et al.* 1997][Kraus *et al.* 1998][Wooldridge and Parsons 2000][Meyer *et al.* 2004a][Zhang *et al.* 2004]. One difficulty of logical frameworks of negotiation is that quantitative criteria are harder to be applied to bargaining processes, which seem necessary in the analysis of negotiation situations. This paper attempts to bridge the gap between the quantitative analysis of game-theory and qualitative description and reasoning of logic.

One of our basic assumption to negotiation is that outcomes of negotiation are determined by bargainers' belief states. Representation of agent's belief states is normally by a set of beliefs and a revision operator over the belief set in terms of the theory of belief revision [Gärdenfors 1988]. Alternatively, belief states can also be specified by an order-

ing over propositions, referred to as *epistemic entrenchment* [Gärdenfors 1988]. It is well-known that such two modelings of belief states is equivalent. The main finding of the paper is that *the ordering of epistemic entrenchment can be extended into a numerical measurement over propositions so that a von Neumann-Morgenstern utility function is induced*. With this quantitative representation of bargainers' belief states, Nash's numerical requirements on bargaining solutions can be restated in terms of logical properties. Bargaining items, negotiation outcomes and conflicting arguments can be also explicitly expressed via the extended concept of belief states.

The structure of the paper is the following. After a short motivation in this section, we extend the AGM's epistemic entrenchment to a map from propositions to real numbers. Bargaining problem is then defined as a pair of bargainers' belief states and outcomes of bargaining is specified by a pair of concessions made by two bargainers. In order to induce a von Neumann-Morgenstern utility function over negotiation outcomes, the standard game-theoretical process of randomization over possible outcomes is applied. When all these done, Nash's bargaining solution is ready to be restated in terms of bargainers belief states. Finally we argue that our solution to bargaining problem can not be replaced by the belief revision based solution even though they share many common logical properties.

We will work on a propositional languages $\mathcal{L}$ with finite many propositional variables. A set K of sentences in $\mathcal{L}$ is said to be *logically closed* or to be a *belief set* if $K = Cn(K)$, where $Cn(K) = \{\varphi \in \mathcal{L} : K \vdash \varphi\}$. If F_1, F_2 are two sets of sentences, $F_1 + F_2$ denotes $Cn(F_1 \cup F_2)$. We shall use $\top$ and $\perp$ to denote the propositional constants **true** and **false**, respectively.

2 Entrenchment Measurement

In the theory of belief revision, the belief state of an agent consists of a belief set - *the beliefs held by the agent* - and a revision operator - *which takes a piece of new information as input and outputs a new belief set*. We know that the belief state is uniquely determined by the agent's epistemic entrenchment - *an ordering on beliefs* [Gärdenfors 1988]. However, we shall argue that such an ordering is not enough to specify agent's belief states in bargaining. We will extend the concept of epistemic entrenchment into a more general concept so that a quantitative measurement can be used to capture bargainers' attitudes towards their bargaining items.

Definition 1 An entrenchment measure is a pair of (ρ, e) where e is a real number and ρ is a function from $\mathcal{L}$ to the real number set $\mathbb{R}$ which satisfies the following condition:

(LC) If $\varphi_1, \dots, \varphi_n \vdash \psi$, $\min\{\rho(\varphi_1), \dots, \rho(\varphi_n)\} \leq \rho(\psi)$.

Let $K = \{\varphi \in \mathcal{L} : \rho(\varphi) \geq e\}$, called the *derived belief set* of the entrenchment measure. It is easy to see that K is logically closed, i.e., $K = Cn(K)$, so it is a belief set. Obviously, K is consistent iff $\rho(\perp) < e$. We shall call ρ the *belief state* of an agent and e is the *bottom line* of the belief state.

The following proposition shows that an entrenchment measure uniquely determines an epistemic entrenchment, thus also determines a unique belief revision function over the derived belief set.

Proposition 1 Let (ρ, e) be an entrenchment measure and K be its derived belief set. Define an ordering $\leq$ over $\mathcal{L}$ as follows: for any $\varphi, \psi \in \mathcal{L}$,

1. if $\vdash \varphi, \varphi \leq \psi$ iff $\vdash \psi$;¹
2. if $\not\vdash \varphi$ and $\varphi \in K$, $\varphi \leq \psi$ iff $\rho(\varphi) \leq \rho(\psi)$;
3. if $\varphi \notin K$, $\varphi \leq \psi$ for any ψ .

Then $\leq$ satisfies all postulates (EE1)-(EE5) for epistemic entrenchment (See [Gärdenfors 1988]).

Based on the result, we can define a belief revision operator $*$ that satisfies all AGM postulates: $\psi \in K * \varphi$ if and only if $\psi \in K + \{\varphi\}$ and either $\rho(\neg\varphi) < \rho(\neg\varphi \vee \psi)$ or $\vdash \neg\varphi \vee \psi$.

We shall call $*$ the *revision function derived from (ρ, e)* .

Since our language is finite, it is easy to define a multiple revision operator $\otimes$ by the singleton revision function:

$$K \otimes F = K * (\wedge F)$$

where $(\wedge F)$ is the conjunction of all the sentences in F . This will facilitate the comparison of our results with [Meyer et al. 2004b] and [Zhang et al. 2004].

We remark that any epistemic entrenchment ordering can be extended to an entrenchment measure while keeping the ordering on sentences. Clearly such an entrenchment measure is not unique. An entrenchment measure contains richer structure than an epistemic entrenchment ordering. Interestingly, the extra structure of entrenchment measures cannot be captured by belief revision operations.

Theorem 1 (Independence of Monotone Transformations) Let τ be a strictly increasing monotonic transformation over $\mathbb{R}$, i.e., for any $x, y \in \mathbb{R}$, $x < y$ if and only if $\tau(x) < \tau(y)$. Let $*$ and $*'$ be the derived revision functions from (ρ, e) and $(\tau \circ \rho, \tau(e))$, respectively². Then for any φ , $K * \varphi = K *' \varphi$, where K is the belief set of (ρ, e) .

3 Bargaining Problem

We shall restrict us to the bargaining situations where only two parties are involved. A bargaining situation is a situation in which both bargainers bring their negotiable items to the negotiation table expecting to reach a common agreement. Whenever a conflict presents, concessions from one or both agents are required in order to reach a compromise. Formally, a *bargaining situation* or a *bargaining game* is a pair, $((\rho_1, e_1), (\rho_2, e_2))$, of entrenchment measures over $\mathcal{L}$ such that each belief set derived by the entrenchment measures is logically consistent, i.e. $K_i = \{\varphi \in \mathcal{L} : \rho_i(\varphi) \geq e_i\} \not\vdash \perp$.

¹Thanks to the anonymous reviewers for pointing out an error in the original version.

² $\circ$ is the composition operator on real number functions.

Example 1 Consider the following negotiation scenario: a buyer (agent 1) negotiates with a seller (agent 2) about the price of a product. The buyer's reserve price is less than \$16 and the seller's reserve price is no less than \$10. Let's discretize the problem as follows:

- $p_1 = \{\text{the price is less than \$10}\}.$
- $p_2 = \{\text{the price is less than \$12}\}.$
- $p_3 = \{\text{the price is less than \$14}\}.$
- $p_4 = \{\text{the price is less than \$16}\}.$

Assume that the entrenchment measure of each agent is respectively the following³:

$$\rho_1(\top) = \rho_1(p_1 \rightarrow p_2) = \rho_1(p_2 \rightarrow p_3) = \rho_1(p_3 \rightarrow p_4) = \rho_1(p_4) = 4, \rho_1(p_3) = 3, \rho_1(p_2) = 2, \rho_1(p_1) = 1, \rho_1(\neg p_j) = 0 \text{ for } j = 1, \dots, 4.$$

$$\rho_2(\top) = \rho_2(p_1 \rightarrow p_2) = \rho_2(p_2 \rightarrow p_3) = \rho_2(p_3 \rightarrow p_4) = \rho_2(p_3) = 4, \rho_2(\neg p_2) = 3, \rho_2(\neg p_3) = 2, \rho_2(\neg p_4) = 1, \rho_2(p_j) = 0 \text{ for } j = 1, \dots, 4.$$

Suppose that both sides set their bottom lines to be 1. Then $K_1 = Cn(\{p_1, p_2, p_3, p_4\})$ and $K_2 = Cn(\{\neg p_1, \neg p_2, \neg p_3, \neg p_4\})$. ¶

Note that the values of entrenchment measures reflect the degree of entrenchment on negotiable items of each agent rather than the agent's preference. For instance, the buyer in above example entrenches p_4 more firmly than p_1 even though p_1 is more profitable because p_4 is much easier to keep than p_1 ⁴.

Definition 2 A bargaining game $B = ((\rho_1, e_1), (\rho_2, e_2))$ is said to be a subgame of another bargaining game $B' = ((\rho'_1, e'_1), (\rho'_2, e'_2))$, denoted by $B \sqsubseteq B'$, if, for each $i = 1, 2$,

1. $\rho_i(\varphi) = \rho'_i(\varphi)$ for any $\varphi \in K_i$;
2. $e_i \geq e'_i$.

Note that $B \sqsubseteq B'$ implies $K_i \subseteq K'_i$ ($i = 1, 2$). Therefore B can be viewed as a kind of *conservative* concession to each other in the sense that each player gives up part of their initial demands by raising their negotiable bottom lines while keeping their entrenchment measurements on negotiable items unchanged.

4 Possible Outcomes of Bargaining

We now consider the possible outcomes of a bargaining game. Apparently the outcome of a bargaining process is the agreements that are reached in the negotiation. In most situations when the demands of two agents conflict, concessions from players are required. The final negotiated outcome then consists of the combination of those demands that each agent chooses to retain. In this section we shall first consider all possible compromises a bargaining process could reach and then combine these outcomes in a probabilistic fashion.

³To make the presentation simple, we do not give the entrenchment measurements for the whole language. The readers are invited to complete the measurements with reasonable ranking for the rest of sentences.

⁴This exactly follows the original explanation of epistemic entrenchment. For instance, φ is no less entrenched than $\varphi \wedge \psi$ because φ is easier to retain than $\varphi \wedge \psi$, so keeping φ is less costly than $\varphi \wedge \psi$ from the information economics point of view.

4.1 Pure deals of bargaining

Following [Meyer et al. 2004a] and [Zhang et al. 2004], we shall define an outcome of bargaining as a pair of subsets of two agents' bargaining item sets, interpreted as the concessions made by both agents. Considering the real-life bargaining, bargainers normally intend to keep their highly entrenched negotiable items and to give up those less entrenched items if necessary. This idea leads to the following definition of possible bargaining outcomes.

Definition 3 Let $B = ((\rho_1, e_1), (\rho_2, e_2))$ be a bargaining game. K_i ($i = 1, 2$) is the belief set derived by (ρ_i, e_i) . A pure deal of B is a pair (D_1, D_2) satisfying:

1. $D_i \subseteq K_i$
2. $\top \in D_i$
3. $D_1 \cup D_2 \not\vdash \perp$
4. if $\varphi \in K_i$ and $\exists \psi \in D_i (\rho_i(\varphi) \geq \rho_i(\psi))$, then $\varphi \in D_i$

where $i = 1, 2$. The set of all pure deals of B is denoted by $\Omega(B)$.

There is little need to comment on the first three conditions. They are just the statements on the type of compromises desired. The last one expresses the idea that the procedure of concession by both agents is in the order from lower entrenched items to higher entrenched items: *higher entrenched items should be retained before any lower entrenched items are considered to be given up*. Combining this condition with the second implies the *individual rationality* – if an agent entrench an item as firmly as a tautology, this item will never be given up in any bargaining situation. Such an item is called *reserved item*.

We are now ready to define the utility of pure deals for each player. We will evaluate an agent's gain from a deal by measuring the length of the remained negotiable items of the agent.

Definition 4 For any $D \in \Omega(B)$, define the utility of the deal, denoted by $u(D) = (u_1(D), u_2(D))$, as follows:

$$u_i(D) = \rho_i(\top) - \min\{\rho_i(\varphi) : \varphi \in D_i\}$$

Example 2 Consider Example 1 again. All possible pure deals of the bargaining game are:

$$\begin{aligned} D^1 &= (Cn(\{p_2, p_3, p_4\}), Cn(\{\neg p_1\})) \\ D^2 &= (Cn(\{p_3, p_4\}), Cn(\{\neg p_1, \neg p_2\})) \\ D^3 &= (Cn(\{p_4\}), Cn(\{\neg p_1, \neg p_2, \neg p_3\})) \end{aligned}$$

The utility values of the deals are:

$$u(D^1) = (2, 0), u(D^2) = (1, 1), u(D^3) = (1, 0).$$

Note that $(Cn(\{p_1, p_2, p_3, p_4\}), Cn(\emptyset))$ and $(Cn(\emptyset), Cn(\{\neg p_1, \neg p_2, \neg p_3, \neg p_4\}))$ are not pure deals because p_4 and $\neg p_1$ are reserved items of agent 1 and 2, respectively. ¶

Given a bargaining game B , if $\Omega(B)$ is nonempty, it must contain two extreme cases of pure deals, called *all-or-nothing deals*:

$$\overline{D} = (K_1, R_2) \text{ and } \overline{D} = (R_1, K_2)$$

where $R_i = \{\varphi \in K_i : \rho_i(\varphi) = \rho_i(\top)\}$, i.e., the reserved items of agent i .

The utilities of the all-or-nothing deals give the up-bound and low-bound of utility values for each agent:

$$u(\overline{D}) = (\rho_1(\top) - e_1, 0) \text{ and } u(\overline{D}) = (0, \rho_2(\top) - e_2).$$

We say that a pure deal D *dominates* another pure deal D' , denoted by $D \succ D'$, if either $D_1 \supset D'_1 \wedge D_2 \supseteq D'_2$ or $D_1 \supseteq D'_1 \wedge D_2 \supset D'_2$. D *weekly dominates* D' , denoted by $D \succeq D'$, if $D_1 \supseteq D'_1 \wedge D_2 \supseteq D'_2$.

A pure deal D of a bargaining game B is said to be *Pareto optimal* over $\Omega(B)$ if there does not exist another pure deal in $\Omega(B)$ which dominates D .

The following theorem shows that if a negotiation function takes a Pareto optimal pure deal as its outcome, then it satisfies all logical requirements put on negotiation functions by [Meyer et al. 2004a] and all the postulates except *Iteration* proposed in [Zhang et al. 2004].

Theorem 2 For any $(D_1, D_2) \in \Omega(B)$, if it is Pareto optimal, then it satisfies all the postulates introduced by [Meyer et al. 2004a] for so-called *concession-permissible deals*⁵:

- (C1) $D_i = Cn(D_i)$ for $i = 1, 2$.
- (C2) $D_i \subseteq K_i$ for $i = 1, 2$.
- (C3) If $K_1 \cup K_2$ is consistent, $D_i = K_i$ for $i = 1, 2$.
- (C4) $D_1 \cup D_2$ is consistent.
- (C5) If either $K_1 \cup D_2$ or $K_2 \cup D_1$ is consistent, then $K_1 \cap K_2 \subseteq Cn(D_1 \cup D_2)$.

Moreover, if $\otimes_1$ and $\otimes_2$ are the belief revision operators derived by the two entrenchment measures of B , respectively, then D satisfies the following conditions introduced by [Zhang et al. 2004]:

1. $Cn(D_1 \cup D_2) = (K_1 \otimes_1 D_2) \cap (K_2 \otimes_2 D_1)$.
2. $K_1 \cap (K_2 \otimes_2 D_1) \subseteq K_1 \otimes_1 D_2$ and $K_2 \cap (K_1 \otimes_1 D_2) \subseteq K_2 \otimes_2 D_1$.

The following lemma shows that the utility value of a pure deal uniquely determines the deal. The fact will be used in the next section.

Lemma 1 For $D, D' \in \Omega(B)$,
 $u(D) = u(D')$ iff $D = D'$.

4.2 Mixed deals

In many cases, a bargaining game could end in a tie that no single agreement reaches. A standard method to deal with the problem in game theory is to play a lottery to determine the outcome of a bargaining game. Following Nash's utility theory, we call $rD' + (1 - r)D''$ a *mixed deal* if D' and D'' are two pure deals or mixed deals, meaning the lottery which has the two possible outcomes, D' and D'' , with probabilities r and $1 - r$, respectively. The set of all mixed deals is denoted by $\overline{\Omega}(B)$ ⁶.

For any mixed deal $D = rD' + (1 - r)D''$, the utility of the deal can be defined as

$$u(D) = ru(D') + (1 - r)u(D'').$$

The concept of domination then can be extended to mixed deals: for any mixed deals $D, D' \in \overline{\Omega}(B)$,

$$D \succ (\succeq) D' \text{ iff } u(D) > (\geq) u(D').^7$$

⁵In [Meyer et al. 2004a], six postulates were introduced to specify the rationality of concessionary. However the sixth, (C6), holds trivially provided K_1 and K_2 are logically closed.

⁶More precisely, $\overline{\Omega}(B)$ is a conservative extension of $\Omega(B)$ by allowing lottery deals in the form $rD' + (1 - r)D''$.

⁷For any $x, y \in \mathbb{R}^2$, $x > y$ denotes $x_1 > y_1 \wedge x_2 \geq y_2$ or $x_1 \geq y_1 \wedge x_2 > y_2$. $x \geq y$ denotes that $x_i \geq y_i$.

Since the language we consider is finite, the set of pure deals of a bargaining game is also finite. The following lemma is easy to prove according to the construction of mixed deals.

Lemma 2 Let $D^1, \dots, D^m \in \Omega(B)$ be all pure deals of a bargaining game B . Any mixed deal $D \in \overline{\Omega}(B)$ can be expressed as a linear combination of $D^1, \dots, D^m$, i.e., there exist real numbers, $\alpha_1, \dots, \alpha_m$, satisfying

1. $\alpha_j \geq 0, j = 1, \dots, m$;
2. $\sum \alpha_j = 1$;
3. $D = \sum \alpha_j D^j$.

Moreover, $u(D) = \sum \alpha_j u(D^j)$.

In other words, $\overline{\Omega}(B)$ is a convex hull of $\Omega(B)$ if we identify a deal with its utility pair. Obviously, the utility function over mixed deals defined above is a pair of von Neumann-Morgenstern utility functions.

We have seen from Lemma 1 that a pure deal can be uniquely determined by its utility values. However, a deal can be dublicately represented in the form of mixed deals. For instance, any pure deal D can be represented as $rD + (1-r)D$. To solve the problem we can either use Lemma 2 to obtain a unique representation of a deal by orthogonalizing representation of pure deals (or utility pairs) or apply equivalence classification by defining an equivalent relation: $D \sim D'$ iff $u(D) = u(D')$. To avoid too much complexity, we omit the details of the mathematical treatment but simply assume that for any $D, D' \in \overline{\Omega}(B)$, $D \equiv D'$ iff $u(D) = u(D')$.

5 Bargaining solution

The target of bargaining theory is to find theoretical predictions of what agreement, if any, will be reached by the bargainers. John Nash in his path-breaking papers introduced an axiomatic method which permitted a unique feasible outcome to be selected as the solution of a given bargaining problem [Nash 1950][Nash 1953]. He formulated a list of properties, or axioms, that he thought solution should satisfy, and established the existence of a unique solution satisfying all the axioms. In this section we shall give a brief summary of Nash's theory and then extend Nash's result in terms of bargainer's belief states.

5.1 The Nash bargaining solution

A bargaining game in Nash's terminology is a pair (Ω, d) , where $\Omega \subseteq \mathbb{R}^2$ is a set of possible utility pairs and $d \in \mathbb{R}^2$ the disagreement point. It was assumed that Ω is convex and compact. All bargaining games satisfying the conditions are collected in the set $\mathcal{B}$. A bargaining solution is then a function $f : \mathcal{B} \rightarrow \mathbb{R}^2$ such that $f(\Omega, d) \in \Omega$. Nash proposed that a bargaining solution should possess the following four properties [Nash 1953]⁸:

- PO** (Pareto-Optimality) There does not exist $(u_1, u_2) \in \Omega$ such that $(u_1, u_2) > f(\Omega, d)$.
- SYM** (Symmetry) If (Ω, d) is symmetric, then $f_1(\Omega, d) = f_2(\Omega, d)$.

⁸See more modern treatments and detailed explanation from [Roth 1979][Owen 1995][Houba and Bolt 2002][Napel 2002].

IEUR (Independence of Equivalent Utility Representation)

Let (Ω', d') be obtained from (Ω, d) by a strictly increasing affine transformation $\tau(x) = (\tau_1(x_1), \tau_2(x_2))$ ⁹. Then $f(\Omega', d') = \tau(f(\Omega, d))$.

IIA (Independence of Irrelevant Alternatives) If $\Omega' \subseteq \Omega$ and $f(\Omega, d) \in \Omega'$, then $f(\Omega', d) = f(\Omega, d)$.

Theorem 3 (Nash's Theorem) A bargaining solution f satisfies all the four properties if and only if $f = F$, where F is defined by $F(\Omega, d) = u^*$ where u^* solves the maximization problem $\max\{u_1 u_2 : u = (u_1, u_2) \in \Omega, u \geq d\}$.

5.2 Bargaining solution with belief states

We might have noticed that a solution to bargaining problem is to provide a general model of the bargaining process rather than a selection function that picks up a point as outcome from all feasible solutions for a particular bargaining situation. A surprise with Nash's theorem is that the axioms shown above seem so natural that any negotiation process should follow whereas the solution is so specific that it uniquely determines an outcome in any bargaining situation. The question now is that whether we can restate these plausible requirements in terms of bargainers' belief states instead of utility functions.

Given a finite propositional language, let $\mathcal{B}$ denote all bargaining games in the same propositional language with nonempty set of deals. A bargaining solution is defined as a function f which maps a bargaining game $B \in \mathcal{B}$ to a mixed deal in $\overline{\Omega}(B)$. Following Nash's approach, we propose the following axioms:

PO (Pareto Optimality) There does not exist a deal $D \in \overline{\Omega}(B)$ such that $D \succ f(B)$.

MD (Midpoint Domination) $f(B) \succeq \frac{1}{2} \overleftarrow{D} + \frac{1}{2} \overrightarrow{D}$.

IEEM (Independence of Equivalent Entrenchment Measurement) Given a bargaining game $B = ((\rho_1, e_1), (\rho_2, e_2))$, for any strictly increasing affine transformations τ ,

$$f(\tau(B)) \equiv f(B)$$

where $\tau(B) = ((\tau_1 \circ \rho_1, \tau_1(e_1)), (\tau_2 \circ \rho_2, \tau_2(e_2)))$.

IIA (Independence of Irrelevant Alternatives) If $B' \subseteq B$ and $f(B) \in \overline{\Omega}(B')$, then $f(B') \equiv f(B)$.

We can easily see the similarities between the above axioms and Nash's. The differences are also observable. Firstly, the above axioms are presented in terms of bargainers' belief states, i.e., the entrenchment measurement of two agents. This is significant not only because bargainers' belief states contains the logical implication of negotiation demands but also because the entrenchment measurements are more obtainable than the utility functions over all possible outcomes (combinatorial explosion can happen with possible outcomes). Secondly, our axioms are more intuitive and easier to verify. For instance, the MD, which is a replacement of Nash's SYM, expresses the idea that a minimal amount of co-operation among the players should allow them to do at least as well as the average of their preferred outcomes. When no easy agreement on a deterministic outcome is obtained, the tossing of a coin is always an option to determine a winner

⁹A strictly increasing affine transformation is a pair of linear real functions with positive slope, i.e., $\tau_i(x) = a_i x + b_i$ where $a_i > 0$.

of the negotiation (see a counterpart of MD in game theory in [Thomson 1994]). Additionally, our IIA is much weaker than Nash's IIA because not every convex subset of $\mathbb{R}^2$ can be a set of utility pairs of mixed deals.

Exactly like the Nash's theorem, there exists a unique bargaining solution possessing the above properties.

Theorem 4 *There is a unique solution f which satisfies PO, IEEM, MD and IIA. Moreover, the solution is the function $F(B) \equiv D^*$ such that $D^* \in \overline{\Omega}(B)$ and $u_1(D^*)u_2(D^*) = \max_{D \in \overline{\Omega}(B)} (u_1(D)u_2(D))$.*

Proof: The whole proof of the theorem is quite lengthy. To save space, we shall only present a sketch of the second half of the proof. We shall prove that if f is a bargaining solution possessing PO, IEEM, MD and IIA, then $f = F$.

Given a bargaining game $B = ((\rho_1, e_1), (\rho_2, e_2))$, let K_i be the derived belief sets of B . In the case that $K_1 \cup K_2$ is consistent, it is easy to show that $f(B) \equiv F(B) \equiv (K_1, K_2)$. So we assume that $K_1 \cup K_2$ is inconsistent.

Let $D^* \equiv F(B)$. Suppose that B' is the bargaining game derived from B by changing its entrenchment measurement with a strictly increasing affine transformation τ so that $u'(D^*) = (\frac{1}{2}, \frac{1}{2})$. By IEEM, $f(B) \equiv D^*$ iff $f(B') \equiv D^*$. Since F satisfies IEEM, D^* also maximizes $u'_1(D)u'_2(D)$ over $\overline{\Omega}(B')$. Therefore we are left to prove $f(B') \equiv D^*$.

Let $B' = ((\rho'_1, e'_1), (\rho'_2, e'_2))$. Following Nash's argument, it is not hard to verify that for any $D \in \overline{\Omega}(B')$, $(u'_1(D), u'_2(D)) \in \{(x, y) : x \geq 0, y \geq 0 \text{ and } x + y \leq 1\}$ because $\overline{\Omega}(B')$ is convex. Particularly, $\rho'_i(\top) - e'_i \leq 1$.

Since $K_1 \cup K_2$ is inconsistent and $\Omega(B)$ is nonempty, we can pick up two Pareto optimal pure deals from $\Omega(B')$ in the form (K_1, U_2) and (U_1, K_2) . Construct a game $B'' = ((\rho''_1, e''_1), (\rho''_2, e''_2))$ as following: for each $i = 1, 2$,

1. for any $\varphi \in K_i$, $\rho''_i(\varphi) = \rho'_i(\varphi)$;
2. for any $\varphi \in Cn(K_i \cup \{\neg(\wedge U_i)\}) \setminus K_i$, $\rho''_i(\varphi) = \rho'_i(\top) - 1$;
3. $e''_i = \rho'_i(\top) - 1$.

It is easy to prove that B' is a subgame of B'' and $K''_i = K_i \cup \{\neg(\wedge U_i)\}$ ($i = 1, 2$). According to MD, $u''(f(B'')) \geq u''(\frac{1}{2}\overline{D}'' + \frac{1}{2}\overline{D}''') = (\frac{1}{2}, \frac{1}{2})$. On the other hand, for any mixed deal $D \in \overline{\Omega}(B'')$, by Lemma 2, $D \equiv \sum \alpha_j D^j$, where $D^j \in \Omega(B'')$. For each j , if both $D^j_i \subseteq K_i$ ($i = 1, 2$), then $u''(D^j) = u'(D^j)$; otherwise either $D^j = \overline{D}''$ or $D^j = \overline{D}'''$, which implies $u''_1(D^j) + u''_2(D^j) = 1$. Thus $u''_1(D) + u''_2(D) \leq 1$. This implies $u''_1(f(B'')) + u''_2(f(B'')) \leq 1$. Therefore $u''(f(B'')) = (\frac{1}{2}, \frac{1}{2})$. It follows that $f(B'')$ is expressible in B' , or $f(B'') \in \overline{\Omega}(B')$. According to IIA, $f(B'') \equiv f(B')$. We conclude that $f(B) \equiv f(B') \equiv D^* \equiv F(B)$. $\blacksquare$

With the theorem we can easily see that the solution to the bargaining situation described in Example 1 is the equivalent class of the pure deal $D^2 = (Cn(\{p_3, p_4\}), Cn(\{\neg p_1, \neg p_2\}))$, under which both agents agree that the price of the product is less than \$14 but no less than \$12. Note that a more refined discretization of the problem might give a mixed deal as the solution of the bargaining game.

The uniqueness of bargaining solution depends on the assumption that both bargainers agree to randomize between outcomes, i.e., whenever the solution of a bargaining game

gives a mixed deal, a lottery will be played to decide which pure deal will be the agreement. However, the assumption is not generally applicable to most of real-life bargaining. In this case, we could use a pure deal that maximizes the product of utility pairs as an approximate prediction of bargaining outcomes even though it is not necessarily unique (Note that such a solution must be Pareto optimal).

6 Discussion

We have observed that a bargaining game consists of two independent entrenchment measurements on negotiable items of two individuals. The actual values of entrenchment measures can be "freely" scaled without changing the solution of the bargaining game. How could these "random" assignments of an entrenchment measure determine a unique solution? Before we answer the question, let's explore another example.

Example 3 [Owen 1995] *Two men are offered \$100 if they can decide how to divide the money. The first man is assumed to be very rich, while the second man has \$50 capital in all. As the first player is very rich, it is assumed that the utility of \$x for the first player, with $0 \leq x \leq 100$, is proportional to x, i.e. $u_1(x) = x$. To the second player, it is assumed that the utility of a sum of money is proportional to its (natural) logarithm, i.e., $u_2(x) = \ln(150 - x) - \ln 50$.*

The game-theoretical solution to the problem is simply maximizing the product of two agents' utilities : $u_1(x)u_2(x) = x \ln(\frac{150-x}{50})$, which gives approximately $x = 57.3$, meaning that the first player receives \$57.3 and the other takes \$43.7.

The logical solution to the problem needs a process of discretization. Let $P(x)$ denote the proposition that the first player receives no less than \$x, where x is a natural number ($0 \leq x \leq 100$)¹⁰.

The negotiable item set of each agent is then:

$K_1 = Cn(\{P(x) : 0 \leq x \leq 100\})$ and $K_2 = Cn(\{\neg P(x) : 0 \leq x \leq 100\})$.

Let $C = \{P(x) \rightarrow P(x-1) : 0 < x \leq 100\}$, which is the common knowledge of both players. The associated entrenchment measure of the first player can be defined as:

- $\rho_1(\varphi) = 100$ for each $\varphi \in \{\top\} \cup C$;
- $\rho_1(P(x)) = 100 - x$ for any $0 \leq x \leq 100$;
- $\rho_1(\varphi) = -1$ for any $\varphi \notin K_1$.

The entrenchment measure of the second player is:

- $\rho_2(\varphi) = 100$ for each $\varphi \in \{\top\} \cup C$;
- $\rho_2(\neg P(x)) = 100 - \ln(\frac{150-x}{50})$ for any $0 \leq x \leq 100$;
- $\rho_2(\varphi) = -1$ for any $\varphi \notin K_2$.

It is easy to verify that the bargaining solution of the game is $(Cn(\{P(57)\} \cup C), Cn(\{\neg P(58)\} \cup C))$, which gives the similar result as the game-theoretical solution.

The result seems strange: the poor man receives less money than the rich man. This is because the poor man is so eager to get money that he highly entrench each single dollars ($98 \leq \rho_2(\neg P(x)) \leq 100$ for any $0 \leq x \leq 100$)¹¹. If the second agent linearly entrenches its gain as the first agent, the

¹⁰Note that we treat $P(x)$ as a proposition rather than a predicate.

¹¹Game-theory explains such a phenomenon as the interpersonal differences of attitudes towards threats or negotiation power of dif-

negotiation would end with 50 to 50. This means that the non-linearity of the second agent's entrenchment measurement results the non-balanced allocation. In other words, the distribution of entrenchment measurement reflects the bargaining power of players. Note that the logarithm function is strictly monotone. Therefore the non-linearity of entrenchment measurement cannot be captured by any belief change operations (see Theorem 1). The AGM's entrenchment ordering is not enough to measure players' bargaining power. In fact, the entrenchment measurement plays two rules in bargaining: *its ordering determines a unique belief revision operation for minimizing the loss of bargainer's negotiable items and the distribution of entrenchment values decides the negotiating power of bargainers.*

7 Conclusion and Related Work

We have presented a logical framework for Nash's bargaining solution. To do so, we introduced a concept of entrenchment measure, which maps a proposition to a real number, and used it as a vehicle to convey agents' belief states in different bargaining situations. We have shown that Nash bargaining solution can be restated in terms of the extended concept of belief states. Negotiable items, bargaining outcomes and conflicting arguments can then be expressed in propositional logic meanwhile Nash's numerical solution to bargaining problem is still applicable. This offers a combinative approach of qualitative and quantitative analysis to bargaining situations. As a direct application of the result, a logic-based solution to automated negotiation can be proposed. Given a two-agent bargaining situation, we invite both agents to describe their negotiable items in terms of propositional logic and to provide their own measurements of entrenchment on their negotiable items. All the information comes to an arbitrator (or a server) who will announce the outcome of the negotiation after conducting a process of belief revision and a calculation of Nash's solution. This procedure should be considered to be fair because the result of negotiation is in effect determined by the participants' belief states rather by the arbitrator.

This paper is closely related to the work on the belief revision model of negotiation [Booth 2001][Meyer et al. 2004a][Meyer et al. 2004b][Zhang et al. 2004]. We have proved that if the bargaining solution takes a pure deal as the outcome, it will satisfy almost all assumptions that are put on the negotiation function constructed by belief revision operators (see Theorem 2). However, as we have discussed in the last section, belief revision operation cannot measure bargainers' negotiation power. Therefore a negotiation function defined purely by belief revision operation is unable to determine negotiation outcomes.

The model we presented in the paper is a logical extension of Nash's *cooperative model* of bargaining, which deals with the bargaining situations described by a set of abstract rules. *Non-cooperative models* of bargaining, in contrast, analyze interaction which is based on explicit rules of bargaining [Binmore et al. 1992]. A series of work have been done that attempts to represent negotiation procedures in terms of

ferent negotiators [Owen 1995][Houba and Bolt 2002][Fatima et al. 2005].

logic [Kraus et al. 1998][Sycara 1990][Sierra et al. 1997]. It is interesting to see whether the classical results about the relationship between cooperative solutions and noncooperative solutions of bargaining problem can be extended to the logical models of bargaining.

Nash's axiomatic approach to bargaining has reached a high sophistication through the development of last five decades [Roth 1979][Thomson 1994][Napel 2002]. A variety of alternative assumptions have been proposed to derive a given solution concept. It could be a promising research topic to investigate logical properties of these alternative solutions. On the other hand, the logical implication of the distribution of entrenchment measurements is also worthwhile to be further explored.

References

- [Binmore et al. 1992] K. Binmore, M. Osborne and A. Rubinstein, Noncooperative models of bargaining, *Handbook of Game Theory*, Vol 1, 180-225.
- [Booth 2001] R. Booth, A negotiation-style framework for non-prioritized revision, *TARK'01*, 137-150.
- [Fatima et al. 2005] S. Fatima, M. Wooldridge and N. R. Jennings, N. R., Bargaining with incomplete information, to appear in *Annals of Mathematics and Artificial Intelligence*.
- [Gärdenfors 1988] P. Gärdenfors, *Knowledge in Flex: Modeling the Dynamics of Epistemic States*, The MIT Press, 1988.
- [Houba and Bolt 2002] H. Houba and W. Bolt, *Credible Threats in Negotiations*, Springer.
- [Kraus et al. 1998] S. Kraus, K. Sycara, A. Evenchik, Reaching agreements through argumentation: a logical model and implementation, *Artificial Intelligence*, 104, 1-69.
- [Meyer et al. 2004a] T. Meyer, N. Foo, R. Kwok and D. Zhang, Logical foundations of negotiation: outcome, concession and adaptation, *AAAI-04*, 293-298.
- [Meyer et al. 2004b] T. Meyer, N. Foo, R. Kwok and D. Zhang, Logical foundations of negotiation: strategies and preferences, *KR'04*, 311-318.
- [Nash 1950] J. Nash, The bargaining problem, *Econometrica*, 18(2), 155-162.
- [Nash 1953] J. Nash, Two-person cooperative games, *Econometrica*, 21(1), 129-140.
- [Napel 2002] S. Napel, *Bilateral Bargaining*, Springer.
- [Owen 1995] G. Owen, *Game Theory*, Academic Press.
- [Roth 1979] A. Roth, *Axiomatic Models of Bargaining*, Springer-Verlag.
- [Sierra et al. 1997] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons, 1997. A framework for argumentation-based negotiation, *ATAL-97*, 177-192.
- [Sycara 1990] K. Sycara, Persuasive argumentation in negotiation, *Theory and Decision*, 28:203-242.
- [Thomson 1994] W. Thomson, Cooperative models of bargaining, *Handbook of Game Theory*, Volume 2, Elsevier Science B.V., 1237-1284.
- [Wooldridge and Parsons 2000] M. Wooldridge and S. Parsons, Languages for negotiation, *ECAI2000*, 393-400.
- [Zhang et al. 2004] D. Zhang, N. Foo, T. Meyer and R. Kwok, Negotiation as mutual belief revision, *AAAI-04*, 317-322.