

POSTER PAPERS

LEARNING AND INFORMATION EXTRACTION

A Study of Selection Noise in Collaborative Web Search*

Oisín Boydell and Barry Smyth

Adaptive Information Cluster

Smart Media Institute, Department of Computer Science,
University College Dublin, Belfield, Dublin 4, Ireland
{oisin.boydell, barry.smyth}@ucd.ie

Cathal Gurrin and Alan F. Smeaton

Adaptive Information Cluster, Centre for Digital Video Processing,
Dublin City University, Glasnevin, Dublin 9, Ireland
{cathal.gurrin, alan.smeaton}@computing.dcu.ie

Abstract

Collaborative Web search uses the past search behaviour (queries and selections) of a community of users to promote search results that are relevant to the community. The extent to which these promotions are likely to be relevant depends on how reliably past search behaviour can be captured. We consider this issue by analysing the results of collaborative Web search in circumstances where the behaviour of searchers is unreliable.

1 Introduction

Traditional approaches to Web search are term-based; they assume that relevance is best understood from the terms within documents. It has been shown that typical Web users are not experts when it comes to searching for documents - they routinely provide ambiguous queries [Lawrence and Giles, 1998], making it difficult for search engines to predict their needs. In such an environment, query and document terms are not always a good indication of relevance and so content-based approaches often fail [Furnas *et al.*, 1987]. Recent innovations have seen the introduction of new sources of relevance knowledge. For example, the work of [Brin and Page, 1998] has famously demonstrated the usefulness of inter-document relationships by exploiting link-connectivity information. More recently researchers have also focused on ways to exploit context in Web search as a way to disambiguate vague queries, either by explicitly establishing the search context up-front or by implicitly inferring the context as part of the search process (see [Lawrence, 2000]). Yet others have focused on leveraging knowledge about the query-space to directly address the correspondence problem, by mining query logs in order to identify useful past queries that may help the current searcher (e.g., [Glance, 2001; Raghavan and Sever, 1995; Wen *et al.*, 2002]). The work

of [Smyth *et al.*, In Pressb] has suggested another source of relevance knowledge: in the *Collaborative Web Search* (CWS) approach patterns of queries and result selections from a community of users are used to re-rank the results of an underlying search engine. A community can be a well-defined group of users, such as the employees of a given company, or it can be an ad-hoc group of individuals, such as the users of some topical Web site, for example a motor-ing Web site. The point is that many searches can be traced back to communities of users and our research indicates that these communities are likely to submit similar queries and select the same types of results [Smyth *et al.*, In Pressb]. In a recent analysis of search logs from a variety of sources we have found that within communities of searchers up to 70% or more of queries share at least 50% of their query terms with past queries from the community; see [Smyth *et al.*, In Pressb]. For example, in [Smyth *et al.*, In Pressa] we describe the results of a recent trial of CWS within a corporate context and a community of about 50 searchers (the employees of an Irish software company). The data collected as part of this trial indicates that 57% of queries share half of their query terms with previous queries. The results of this trial go on to show that CWS can take good advantage of this type of repetition and regularity to significantly enhance search performance.

CWS relies on a relevance assumption—that the selection of a page for some query is a reliable indicator of page relevance—which cannot be fully relied upon. Search engines often return pages that turn out not to be relevant, and users often mistakenly select them. The question is: how sensitive are the results of CWS to selection noise? We will answer this question in what follows by analysing the impact of selection noise on result precision.

2 Collaborative Web Search

CWS is a form of meta-search, relying on the search services of a set of underlying search engines, but manipulating their results in response to the learned preferences of a given community of users. A key data structure in CWS is the *hit ma-*

*This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361.

trix, H . It represents the search behaviour of a given community of users and each time a member of a community selects a result p_j in response to some query q_i the entry in cell H_{ij} is incremented. In turn, the *relevance* of a page p_j to q_i can be estimated as the relative number of selections p_j has received in the past for q_i ; see Equation 1.

$$Relevance(p_j, q_i) = \frac{H_{ij}}{\sum_{\forall j} H_{ij}} \quad (1)$$

$$WRel(p_j, q_T, q_1, \dots, q_n) = \frac{\sum_{i=1 \dots n} Relevance(p_j, q_i) \bullet Sim(q_T, q_i)}{\sum_{i=1 \dots n} Exists(p_j, q_i) \bullet Sim(q_T, q_i)} \quad (2)$$

More generally, the relevance of a page p_j to some query q_T can be calculated as the weighted sum of its relevance to a set of queries that are similar to q_T , namely $q_1, \dots, q_n$, with each individual relevance value discounted by the similarity between q_T and the query in question; see Equation 2. The similarity between q_T and the query in question, q_i is calculated as in Equation 3; it is worth noting that we have recently evaluated a variety of alternative and more sophisticated similarity metrics, the results of which are presented in [Balfe and Smyth, 2005; In Press].

$$Sim(q_T, q_i) = \frac{|q_T \cap q_i|}{|q_T \cup q_i|} \quad (3)$$

Thus on receipt of some target query, q_T , CWS dispatches it to a set of underlying search engines and their results are combined to form a meta-search result-list, R_M . At the same time, q_T is compared to other queries in the hit-matrix to select a set of similar queries, $q_1, \dots, q_n$, that are selected if their similarity to q_T exceeds some set threshold. The results selected for these queries in the past are ranked by their weighted relevance to q_T , according to Equation 2, to produce a new *promoted* result-list, R_P . R_P is combined with R_M to produce R_T , which is then returned to the user; in our implementation $R_T = R_P \cup R_M$. For further detail on the technical details of the CWS architecture and operation see [Smyth *et al.*, In Pressb].

3 Evaluation

The success of CWS depends on the quality of its promoted results, R_P , and their quality in turn depends on the reliability of the user selections that underpin the core relevancy calculations. In this evaluation we will assess the quality of these promotions under different degrees of selection noise with an experiment using the TREC Terabyte collection (see <http://www-nlpir.nist.gov/projects/terabyte>) and the Físréal search engine [Blott *et al.*, In Press].

3.1 The TREC Terabyte Collection

This collection consists of over 25 million Web pages (approx, 426GB) crawled from the .gov domain during early 2004. The TREC Terabyte track included 50 random topics as target search topics. Each topic included a short textual description and during the evaluation competing search engines

were evaluated with respect to their ability to locate pages relevant to these topics. After the evaluation a *relevance engine* was made available to help with the evaluation of new search techniques. This engine provides ground-truth relevance for the topics and allows for the detailed analysis of result-lists in relation to the test queries.

3.2 Training the Hit-Matrix

Normally in CWS the hit-matrix is trained by the searches of a given community of users, but we need a mechanism for manipulating selection noise and so we need a more controlled experimental framework. For this we require a set of queries and a method for judging the relevance of the results that are returned and that might be selected by users. We generate queries by extracting subsets of terms from the TREC topic descriptions, after first removing commonly occurring stop words; for each topic we generate 50 queries with between 2 and 8 terms each. To simulate the action of a searcher we use the official TREC relevance engine, which is capable of identifying all result pages that are relevant for a given topic. Thus, for a training query q generated from topic t , we can identify the k pages returned by a baseline search engine (see [Blott *et al.*, In Press]) that are relevant to t as the basis for updating our hit-matrix for topic t . During the update we use two types of noise. *TypeA* noise occurs when additional non-relevant results are selected and thus during training we assume that the user selects all k retrieved documents that are relevant and an additional $n\%$ of k non-relevant pages. *TypeB* noise occurs when users fail to select all of the relevant results returned and thus during training we assume that the user only selects $(100 - n)\%$ of the k relevant results and $n\%$ of k irrelevant results returned for each query.

3.3 Precision Analysis

To assess the precision of CWS we use the official TREC test queries that formed the basis of the TREC 2004 evaluation; none of these queries were used in training. Two different versions of CWS are used for query similarity thresholds of > 0 and ≥ 0.5 , with different types and levels of selection noise introduced. Each test query is replayed and the percentage of relevant results for different result-list sizes is computed. The average of these precision values is computed to produce an overall *mean average precision* (MAP) for each version of CWS. We also compute a baseline MAP for the TREC Benchmark search engine used in [Blott *et al.*, In Press], which serves as the underlying engine for CWS.

The results are presented in Figures 1 and 2. Each graph shows the MAP for the baseline and for CWS with Type A and B noise. They show that CWS offers significant precision increases over the baseline. For example, for the > 0 threshold, at the 0% noise level, CWS delivers over twice the precision of the baseline (0.34 vs. 0.15). We also see that these precision benefits degrade gracefully as noise is increased, and the benefits are preserved even for high noise levels. For instance, even with 100% Type A noise we see that CWS delivers a MAP of 0.17, which is greater than the corresponding baseline MAP (0.14). We see that CWS is more sensitive to Type B noise, especially for the ≥ 0.5 threshold. Here, CWS MAP falls below the baseline for 80% noise;

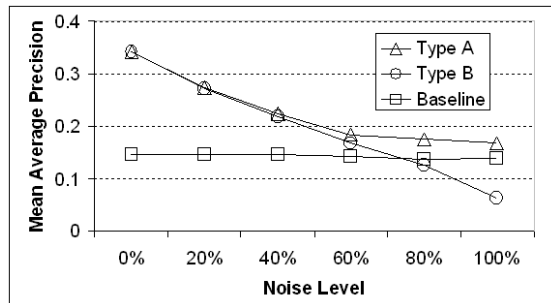


Figure 1: Similarity Threshold > 0

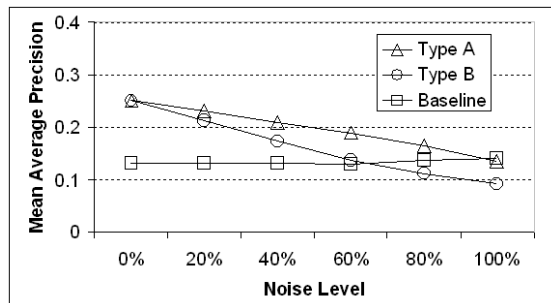


Figure 2: Similarity Threshold ≥ 0.5

this corresponds to a situation where in many search sessions no relevant results are selected by users.

4 Conclusions

CWS personalizes search results for a community of like-minded searchers based on their prior search histories. It is appropriate in a wide variety of search scenarios because many Web searches are initiated within a community context; for example, the searches initiated from a search box on some themed Web site or the searches of some online social group. When responding to a new search query, CWS exploits the results of searches for similar queries that have taken place in the past (for a given community) and actively promotes those results. Thus, results that are consistently selected for queries will tend to be promoted, although care must be taken to limit the influence of promotional biases as a result of selection noise or malicious users.

In this paper we have evaluated the performance of CWS when search histories contain result selections that are unreliable. Obviously, if users select results that are not relevant to their query then there is the risk that these results will be promoted in the future. The question that we have attempted to answer is the degree to which CWS is sensitive to such noisy selections. Our results indicate that CWS is relatively robust to noise with significant precision benefits available even for very high levels of selection noise. This bodes well for CWS and is in agreement with the performance benefits witnessed in live-user trials, in which noisy selections are likely to occur. In finishing, it is worth highlighting that the CWS approach has been fully implemented in the I-SPY system which can be accessed at <http://ispy.ucd.ie>.

References

- [Balfe and Smyth, 2005] Evelyn Balfe and Barry Smyth. An analysis of query similarity in collaborative web search. In *Proceedings of the 27th European Conference on Information Retrieval*, pages 330–344, 2005.
- [Balfe and Smyth, In Press] Evelyn Balfe and Barry Smyth. A Comparative Analysis of Query Similarity Metrics for Community-Based Web Search. In *Proceedings of the 6th International Conference on Case-Based Reasoning*, (In Press).
- [Blott et al., In Press] Stephen Blott, Oisín Boydell, Fabrice Camous, Paul Ferguson, Georgina Gaughan, Cathal Gurrin, Noel Murphy, Noel O’Connor, Alan F. Smeaton, Barry Smyth, and Peter Wilkins. Experiments In Terabyte Searching, Genomic Retrieval And Novelty Detection For TREC-2004. In *Proceedings of the Thirteenth Text REtrieval Conference*, (In Press).
- [Brin and Page, 1998] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.
- [Furnas et al., 1987] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987.
- [Glance, 2001] Natalie S. Glance. Community Search Assistant. In *Proceedings of the International Conference on Intelligent User Interfaces*, pages 91–96. ACM Press, 2001.
- [Lawrence and Giles, 1998] Steve Lawrence and C. Lee Giles. Context and Page Analysis for Improved Web Search. *IEEE Internet Computing*, July-August:38–46, 1998.
- [Lawrence, 2000] Steve Lawrence. Context in Web Search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.
- [Raghavan and Sever, 1995] Vijay V. Raghavan and Hayri Sever. On the reuse of past optimal queries. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 344–350. ACM Press, 1995.
- [Smyth et al., In Pressa] Barry Smyth, Evelyn Balfe, Oisín Boydell, Keith Bradley, Peter Briggs, Maurice Coyle, and Jill Freyne. A Live-User Evaluation of Collaborative Web Search. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-05*, (In Press).
- [Smyth et al., In Pressb] Barry Smyth, Evelyn Balfe, Jill Freyne, Peter Briggs, Maurice Coyle, and Oisín Boydell. Exploiting Query Repetition & Regularity in an Adaptive Community-based Web Search Engine. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, (In Press).
- [Wen et al., 2002] Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Query clustering using user logs. *ACM Transactions on Information Systems*, 20(1):59–81, 2002.

Image Retrieval and Disambiguation for Encyclopedic Web Search

Atsushi Fujii and Tetsuya Ishikawa

Graduate School of Library, Information and Media Studies

University of Tsukuba

1-2 Kasuga, Tsukuba, 305-8550, Japan

Abstract

To produce multimedia encyclopedic content, we propose a method to search the Web for images associated with a specific word sense. We use text in an HTML file which links to an image as a pseudo-caption for the image and perform text-based indexing and retrieval. We use term descriptions in a Web search site called “CYCLONE” as queries and correspond images and texts based on word senses.

1 Introduction

The World Wide Web, which contains an enormous volume of up-to-date information, is a promising source to obtain encyclopedic knowledge. It has become common to consult the Web for specific keywords, instead of using conventional dictionaries and encyclopedias. However, existing Web search engines often retrieve extraneous pages containing low-quality, unreliable, and misleading information.

Fujii and Ishikawa [2001] proposed an automatic method to extract term descriptions from the Web and classify multiple descriptions into domains and word senses. Using this method, Fujii and Ishikawa have built a Web search site called “CYCLONE”¹, in which a user can efficiently obtain an encyclopedic term description in a specific word sense. Over 700,000 Japanese terms are currently indexed as headwords.

However, to explain certain headwords, specifically those related to an entity, such as devices and animals, it is effective to present a picture depicting the entity, in addition to text descriptions.

In this paper, we propose a method to integrate images on the Web and text descriptions in CYCLONE. We resolve the ambiguity of the meaning of an image by text analysis, so that images for a polysemous word, such as “hub (network device and center of wheel)”, are classified on the basis of word senses. Our research is a step toward the automatic compilation of multimedia encyclopedias, such as Encarta².

2 Description-based Image Disambiguation

Existing search engines, such as Google and Yahoo!, use texts in HTML files for retrieval purposes, instead of performing

content-based image retrieval. Given a text query, an image file, such as GIF and JPEG files, linked from an HTML file including one or more query terms is selected as a candidate image. In other words, text content in an HTML file is used as a “pseudo-caption” for a target image.

However, the text-based image retrieval cannot distinguish images depicting different entities, if a query term is polysemous. For example, images of network devices and images of axles can potentially be retrieved together in response to the query “hub”.

To resolve this problem, we use a text description for a specific word sense. Term descriptions in CYCLONE, which are organized on the basis of word senses, provide informative contexts for word sense disambiguation. For example, if an HTML file includes words “LAN” and “cable”, an image linked from this HTML file is likely to depict a network device more than an axle.

We need to crawl the Web and cache a large number of images and HTML files linking to those images. We shall call these HTML files “hyper HTML files”. However, this process is computationally expensive. We experimentally use Yahoo! Japan³ to obtain pairs of an image and its hyper HTML file, by submitting a target term (e.g., “hub”) as a query. We discard all HTML tags in the hyper HTML files and extract the text content, from which we produce a text index.

For indexing and retrieval purposes, any best-match text retrieval method, such as the vector space model and probabilistic model, can be used. We experimentally use Okapi BM25. We use content words, such as nouns, extracted from text as index terms, and perform word-based indexing. We use the ChaSen morphological analyzer⁴ to extract content words, because Japanese sentences lack lexical segmentation. The same method is used to extract terms from queries.

However, unlike a text-based image retrieval which uses well-organized captions [Smeaton and Quigley, 1996], not all words in an HTML file are related to an image. We use different term weights depending on the region in an HTML file. In principle, a decreasing function which assigns a value to each word depending on the proximity between the word and an anchor (i.e., and <A> in HTML) to a target image can be used. In practice, we multiply the weight of a word M

¹<http://cyclone.slis.tsukuba.ac.jp/>

²<http://encarta.msn.com/>

³<http://www.yahoo.co.jp/>

⁴<http://chasen.aist-nara.ac.jp/>



Figure 1: Example text descriptions and images for “habu”.

times, if the number of characters between the word and the anchor to the image is less than N . We shall call this method “proximity-based term weighting (PBTW)”. The values of M and N are determined empirically in Section 3.

Figure 1 depicts a successful example result for the Japanese term “habu”, which is associated with multiple word senses, such as a network device, snake in *Okinawa*, airport, and center. In Figure 1, the first two paragraphs describe device and snake, respectively. The top three image candidates are associated with each paragraph.

3 Experimentation

To evaluate the accuracy of our method, we produced a test collection. First, we collected polysemous words used as test terms. For each test term, at least two word senses must be able to be depicted by image, because our purpose is to disambiguate the meaning of images. We collected 22 test terms.

Second, for each test term, we used Yahoo! Japan to search the Web for the top one hundred images and their hyper HTML files. Third, each image was manually annotated with a word sense, disregarding as to whether or not the sense is included in CYCLONE. The annotator was able to read the content of a hyper HTML file for the decision, if necessary. The images not associated with any word sense are annotated with “irrelevant”. Fourth, for each test term, a query is produced from each description in CYCLONE. We used the top descriptions classified into each domain. The descriptions not associated with any word sense annotated to the images were discarded. The total number of queries was 155.

We compared the following three methods:

- a baseline method, which sorts candidate images for each query according to the ranking in Yahoo! Japan (“Baseline”),
- our method (the description-based disambiguation method), which uses descriptions in CYCLONE as queries (“DBD”),
- our method, which uses descriptions in CYCLONE as

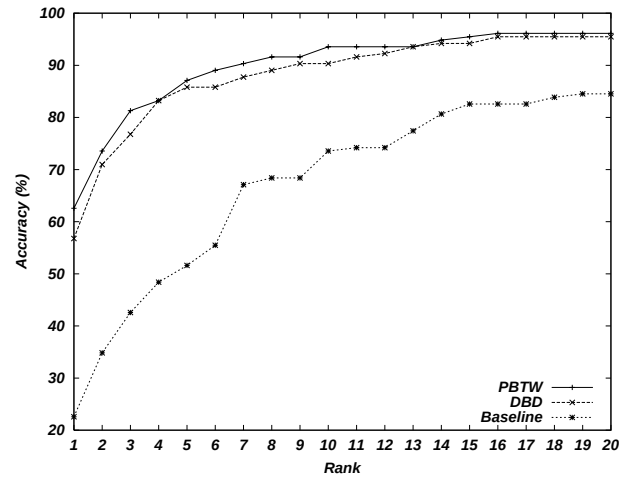


Figure 2: Rank-accuracy graphs for different methods.

queries and performs the proximity-based term weighting method (“PBTW”).

For the baseline method, which did not perform image disambiguation, the same list of candidate images was associated with the different senses for each test term. Through a preliminary experiment, we set $M = 7$ and $N = 400$ for PBTW.

Figure 2 shows the accuracy for each method in different ranks. The accuracy for rank r is the number of queries for which a correct image was found in the top r candidates and the total number of queries. DBD and PBTW significantly improved on the accuracy of the baseline method, irrespective of the rank. PBTW improved on the accuracy of DBD. As predicted, the words in close proximity to an anchor for a target image were important to disambiguate the meaning of the image. PBTW retrieved at least one correct image in the top ten candidates for 93.6% of the queries.

4 Conclusion

To compile multimedia encyclopedic content on the Web, we proposed a method to associate text descriptions in CYCLONE and image files based on word senses, and showed its effectiveness by means of experiments.

References

- [Fujii and Ishikawa, 2001] Atsushi Fujii and Tetsuya Ishikawa. Organizing encyclopedic knowledge based on the Web and its application to question answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 196–203, 2001.
- [Smeaton and Quigley, 1996] Alan F. Smeaton and Ian Quigley. Experiments on using semantic distances between words in image caption retrieval. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–180, 1996.

Learning Complex Event Descriptions by Abstraction

Ugo Galassi¹, Attilio Giordana¹, Lorenza Saitta¹ and Marco Botta²

¹Università Amedeo Avogadro, Dipartimento di Informatica
Via Bellini 25/G, 15100 Alessandria, Italy

²Università di Torino, Dipartimento di Informatica
C.so Svizzera 185, 10149 Torino, Italy

{galassi, attilio, saitta}@mf.n.unipmn.it, botta@di.unito.it

Abstract

The presence of long gaps dramatically increases the difficulty of detecting and characterizing complex events hidden in long sequences. In order to cope with this problem, a learning algorithm based on an abstraction mechanism is proposed: it can infer a Hierarchical Hidden Markov Model, from a learning set of sequences. The induction algorithm proceeds bottom-up, progressively coarsening the sequence granularity, and letting correlations between subsequences, separated by long gaps, naturally emerge. As a case study, the method is evaluated on an application of user profiling. The results show that the proposed algorithm is suitable for developing real applications in network security and monitoring.

1 Introduction

This paper addresses the task of discovering complex events (CEs) occurring sparsely in long sequences. It is assumed that a CE is a partially ordered set of short chains (episodes) of atomic events (AEs), interleaved with gaps, where irrelevant facts may occur. Moreover, the presence of noise can make episodes hard to recognize. Episodes are represented as strings of symbols, being a symbol the label assigned to an atomic event. In a recent paper [Botta *et al.*, 2004], a method for automatically inferring a Hierarchical Hidden Markov Model (HHMM) [Fine *et al.*, 1998] from a database of sequences has been proposed. Here, an improved version of the algorithm is applied to a non trivial application of user profiling in computer security.

2 User Profiling

User profiling is widely used to detect intrusions in computer networks or in telephony networks. The possibility of automatically building a profile for *users* or for *network services* reflecting their temporal behavior would offer a significant help to the deployment of adaptive Intrusion Detection Systems (IDSs) [Lee *et al.*, 2002].

The experiments described in the following investigate the possibility of automatically constructing a user profile from

the logs of its activity. The selected task consists in learning to identify a user from his/her typing style on a keyboard. The basic assumption is that every user has a different way of typing, which becomes particularly evident when he/she types specific words, or sequences of characters. The goal is not challenging the results previously obtained [Bleha *et al.*, 1990; Brown and Rogers, 1993], but investigating the possibility of synthesizing automatically profiles from the activity logs. The specific application has been selected because the data are easy to acquire. In other words, if the methodology described so far succeeds in building up a HHMM for this kind of user profiling, it is likely that it will succeed in other cases as well. Two experiments, described in the following subsections, have been performed.

3 Key Phrase Typing Model

In the first experiment, the goal was to construct a model for a user typing a *key phrase*, discriminant enough to recognize the user among others. A selected sentence of 22 syllables has been typed many times on the same keyboard, while a transparent program recorded the duration of each stroke, and the delay between two consecutive strokes. Then, every repetition of the sentence generated a temporal sequence. Four volunteers provided 140 sequences each, and, for every one of them, a model has been built up using 100 traces (for each user) as learning set. The four learned models have been tested against the remaining 160 traces. For each model λ and for each trace s , the probability of λ generating s has been computed using the forward-backward algorithm [Rabiner, 1989]. Then, s has been assigned to the model with the highest probability. The results reported only one commission error and two rejection errors (no decision taken), when a trace was not recognized by any one of the models.

4 Text Typing Model

The second experiment addressed the more general problem of modeling a user during a text editing activity. A corpus of several paragraphs, selected from newspapers and books, has been collected. The total number of words was 2280, and the number of typed keys 14273. Again, four users typed the entire corpus in several different sessions, without any constraint, in order not to modify their natural typing style. In this kind of application, a user model should be centered not on

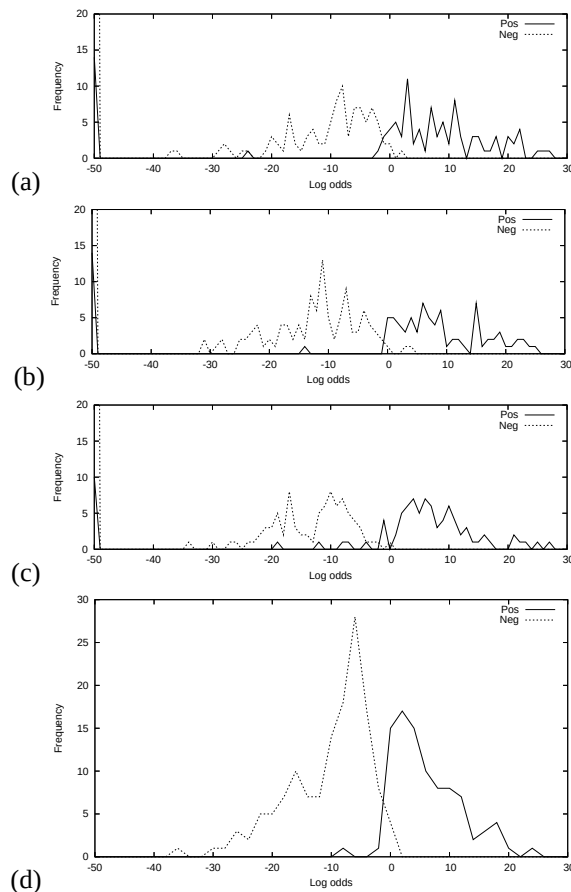


Figure 1: User profiling statistics. Graphs (a), (b), (c) and (d) refer each one to a different user profile. The continuous line "Pos" reports the scoring, measured in log odds, for the sequences belonging to the profile. The dotted line "Neg" refers to the sequences not belonging to the profile.

the specific words he/she types, but on the user typing style, which, in turns, depends on the position of the keys on the keyboard. Therefore, a standard keyboard subdivision into regions, used in dactylography, has been considered. On this basis, keys have been grouped into 10 classes. In this way, transition from one region of the keyboard to another should be emphasized. Afterwards, the sequences generated during a typing session have been rewritten by replacing every character with the name of the class it has been assigned to. Finally, long sequences deriving from an editing session have been segmented into shorter sequences, setting the breakpoint in correspondence of long gaps. The idea is that typical delays due to the user typing style cannot go beyond a given limit. Longer delays are imputable to different reasons, such as thinking or changing of the focus of attention. In this way a set of about 1350 subsequences has been obtained. For every user, a subset of 220 subsequences has been extracted in order to learn the corresponding model. The remaining ones have been used for testing. As in the previous case, the probability of generating each one of the sequences in the test set has been computed for every model. The results are summa-

rized in Figure 1, where the distribution of the scoring rate on the test sequences is reported for every model. The scoring rate is measured in $\log odds$ ¹. The continuous line, labelled "Pos", represents the distribution of the scores assigned to the correct model (user), whereas the other one, labelled "Neg", represents the distribution of the scores assigned to all other models (users), considered together. The sequences on the extreme left have been rejected. It is evident from the figure that sequences belonging to the model are well separated from the other ones. Referring to the data in the test set, a monitoring system using the simple rule that, in a set of three consecutive sequences generated by a user at least two must have a score higher than '0', would give a perfect discrimination of the legal user without rising false alarms.

It is worth noting that the results have been obtained as a first shot, without requiring any tuning of the algorithm. This means that the method is robust and easy to apply to this kind of problems.

5 Conclusion

An improved version of the algorithm [Botta *et al.*, 2004] for inferring complex HHMM from sequences has been applied to a non trivial user profiling problem. Even if this case study is just a preliminary investigation, the results are promising. In fact, the considered case is highly representative of many other similar problems found in intrusion detection systems, and the results show that HHMM is a suitable tool for building profiles. The methodology is effective, robust and easy to apply.

References

- [Bleha *et al.*, 1990] S. Bleha, C. Slivinsky, and B. Hussein. Computer-access security systems using keystroke dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(12):1217–1222, 1990.
- [Botta *et al.*, 2004] M. Botta, U. Galassi, and A. Giordana. Learning complex and sparse events in long sequences. In *Proceedings of the European Conference on Artificial Intelligence, ECAI-04*, Valencia, Spain, August 2004.
- [Brown and Rogers, 1993] M. Brown and S.J. Rogers. User identification via keystroke characteristics of typed names using neural networks. *International Journal of Man-Machine Studies*, 39:999–1014, 1993.
- [Fine *et al.*, 1998] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32:41–62, 1998.
- [Lee *et al.*, 2002] W. Lee, W. Fan, M. Miller, S.J. Stolfo, and E. Zadok. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security*, 10:5–22, 2002.
- [Rabiner, 1989] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

¹The logarithm of the ratio between the probability that the observed sequence is generated by the model and the probability that it is generated by a random process.

CONDITIONAL VISUOMOTOR LEARNING AND VIABILITY THEORY

Fadila Hadj-Bouziane^{1*}, H     Frankowska², Martine Meunier¹ and Driss Boussaoud¹

1. INCM-CNRS, 31 Chemin Joseph Aiguier, 13402 Marseille, France. Email: boussaoud@incm.cnrs-mrs.fr, meunier@incm.cnrs-mrs.fr

2. CNRS, CREA, Ecole Polytechnique, 1 rue Descartes, 75005 Paris, France. Email: franko@shs.polytechnique.fr

* Present address: NIMH, Laboratory of Brain and Cognition, Bethesda, MD, USA. Email: hadjf@mail.nih.gov

Key words: Basal ganglia, striatum, visuomotor learning, viability theory, electrophysiology, monkey, learning strategy
Content areas: rule learning, reinforcement learning

Address for correspondence: boussaoud@incm.cnrs-mrs.fr

Abstract

In conditional visuomotor learning, several arbitrary associations between visual cues and motor responses have to be learned by trial and error at the same time. Monkeys, as humans, do not achieve this task by randomly trying each possible association. Rather, they use a strategy that organizes sequentially the acquisition of individual stimulus-response associations. Accordingly, neuronal recordings in the monkey striatum, the main basal ganglia structure, reveals two forms of plasticity during learning, a transient one that could constitute the neuronal correlate of the strategy, and a long-lasting one that could reflect the slow neuronal implementation of individual associations. Existing models of basal ganglia function based on reinforcement learning cannot account for this dual process. Hence, we developed a mathematical model of conditional visuomotor learning, inspired from viability theory, which implements both the formation of individual associations and the use of strategies to organize learning.

Introduction

Studies of arbitrary visuomotor learning in laboratory situations generally use several stimuli taken from the same category (colors, pictures, positions etc.) and an equivalent number of motor responses (hand postures, lever displacements, etc.). Subjects are required to learn by trial and error, conditional rules such as 'if green go right, if red go left, if blue, go up', etc. This requires on-line monitoring of the associations already tried and of their outcome (correct or incorrect). With large sets of items and actions to associate, this monitoring represents a high memory load, which can be decreased by using specific learning strategies.

Conditional associative learning: behavioral and neuronal data

The striatum, the larger structure composing the basal ganglia, constitutes a key neural substrate for arbitrary visuo-motor learning (see e.g. Hadj-Bouziane et al., 2003 for review). Hence, in a recent study, we recorded single cell activity in the striatum of two rhesus monkeys during learning of a conditional visuomotor task that required to associate complex visual cues with joystick movements (Hadj-Bouziane & Boussaoud, 2003). The animals searched, by trial and error, for the correct joystick movement associated with each of four visual cues. The four associations comprising a set were presented concurrently, that is, each appeared once, in randomized order, within each block of four consecutive trials.

Behavior. Although the four associations were presented concurrently within each block of 4 trials, monkeys learned them sequentially, focusing on one association at a time in order to decrease the memory load. The use of strategies to organize learning was also observed in humans in a separate study (Hadj-Bouziane et al., 2004).

Neuronal activity: Neuronal activity in the striatum showed two main types of modulations during learning. One was transient, consisting in a time-limited increase in activity during early learning stages. The other was long-lasting, consisting in a gradual increment of activity over the entire course of the learning session. The former could underlie the development and use of learning strategies, whereas the latter could be responsible for long term storage of the individual associations.

Mathematical model: the striatum and viability theory

Several not mutually exclusive computational models of basal ganglia have been proposed in the literature (e.g. Houk et al., 1995; Barto, 1995). Many of these models use the reinforcement learning (RL) algorithm. For learning associations by RL algorithms one would need to build a cost function adapted to this particular learning task and then to find a way to maximize this function. However, as described above, learning arbitrary visuomotor associations is not based on random selection of a behavioral response, but rather implies strategies. Mathematically the learning of associations leads to the so called target problem of control theory and there are different tools (cf. viability theory and capture basin algorithms, time optimal control, adaptive control) that allow to find policies theoretically and construct solutions numerically. The learning of policies leading to a target can be derived from viability theory (Aubin 1991). Instead of a repetitive improvement of the cumulative reward to make it as close as possible to a maximum (as it is done in RL algorithms), the viability kernel algorithms are based on the systematic exploration of boundaries of sets. Indeed when we investigate the boundary of a set, instead of the set itself, we reduce the dimension of the problem (Cardaliaguet et al., 1999). We propose that the use of strategies such as the sequential strategy evoked above correspond to investigation of boundaries. From the mathematical view point the viability and capture basin algorithms are based on elimination: taking out points from which it is impossible to get to a target and in the same time constructing the winning strategies for Capture Basin (e.g. Frankowska & Quincampoix, 1991). This elimination is based on a local analysis of constraints. Instead of applying exploration type arguments and trying different controls from the same initial condition, the viability kernel algorithms consider only critical points, that is the boundary of the set of constraints and check which points of the boundary are not viable.

Summary and conclusions

Learning arbitrary visuo-motor associations implies to learn the correct stimulus-movement associations and then to store them in long term memory. One has to learn by trial and error to select the appropriate movement according to the stimulus presented. During early stages of learning, processes such as discrimination, recognition and memorization of the stimuli are required, but also monitoring of events and behavioral outcome. Learning time can be reduced by the use of particular strategies such as focusing on a single cue at a time or repeating a motor response until correct.

We propose a computational model based on 1) these behavioral data showing the use learning strategies, 2) the possible correlates found in the dynamics of neuronal

changes in the striatum, and 3) viability theory. This model complements existing ones, by taking into account the strategies, viewed as a way to determine the boundaries of the learning space. In our model the learning strategies artificially reduce the dimension of learning space (number of actions or number of images), a process that simplifies the task. At the neuronal level, we suggest that the strategy choice may be carried out by the transient activities, whereas the boundary encodings may be accomplished using the rules proposed in Bar-Gad et al. (2003), and local analysis done using the reinforcement signal. We thus propose an additional role of the basal ganglia, that of reducing the dimension of learning space.

Acknowledgements

This work was supported by the French Ministry of Research (ACI Cognition et traitement de l'information), and by the CNRS (PI TCAN), and a Bettencourt-Schuller foundation fellowship for FHB.

References

- Aubin JP. Viability Theory, Birkhauser, Boston, Basel, Berlin, 1991.
- Barto A. Adaptive critics and the basal ganglia. In Models of Information Processing in the Basal Ganglia, Eds. Houk J.C., Davis J.L., and Beiser D.G., MIT Press, Cambridge, MA, pages 215-232, 1995.
- Bar-Gad I, Morris G., and Bergman H. Information processing, dimensionality reduction and reinforcement learning in the basal ganglia. *Progress in Neurobiology*, 71: 439-473, 2003.
- Cardaliaguet P, Quincampoix M and Saint-Pierre P. Set-valued numerical methods for optimal control and differential games, in Stochastic and differential games. Theory and numerical methods, In *Annals of the International Society of Dynamical Games*, Birkhauser, pages 177-247, 1999.
- Frankowska H and Quincampoix M. Viability kernels of differential inclusions with constraints: algorithm and applications. *J. Math. Systems, Estimation and Control*, 1:371-388, 1991.
- Hadj-Bouziane F and Boussaoud D. Activity of neurons in the striatum during learning of arbitrary visuo-motor associations. *Exp Brain Res*, 18:269-277, 2003.
- Hadj-Bouziane F, Meunier M and Boussaoud D. Conditional visuo-motor learning in primates: a key role for the basal ganglia. *Journal of Physiology-Paris*, 97:567-579, 2003.
- Hadj-Bouziane F, Benatru I, Klinger H, Brousolle E, Boussaoud D and Meunier M. Conditional associative learning in advanced Parkinson's disease. *Soc. Neurosci. Abstr.* 710.4, 2004.
- Houk JC, Adams JL and Barto AG. A model of how the basal ganglia generate and use reward signals that predict reinforcement. In JC Houk, JL Davis and DG Beiser (Eds.), *Cambridge: MIT Press Models of information processing in the basal ganglia*, pages 249-270, 1995.

fMRI Analysis via One-class Machine Learning Techniques

David R. Hardoon

University of Southampton, ECS, ISIS Research Group
Southampton, UK SO17 1BJ. drh@ecs.soton.ac.uk

Larry M. Manevitz

University of Haifa, Department of Computer Science
Haifa, Israel 31905. manevitz@cs.haifa.ac.il

Abstract

We show how one-class compression Neural Networks and one-class SVM can be applied to fMRI data to learn the classification of brain activity associated with a specific motor activity. For comparison purposes, we use two labeled data and see what degree of classification ability is lost compared with the usual two-class SVM.

1 Introduction

Functional magnetic resonance imaging (fMRI) allows the carrying out of specific non-invasive studies within a given subject while providing an important insight to the neural basis of brain processes. Neurons, which are the basic functional unit of the brain, consume a higher level of oxygen when active, hence blood with a higher level of oxygenation is supplied to those active neurons. fMRI makes indirect use of this effect by detecting areas of the brain which have an elevated consumption of oxygen. This effect can be used to identify areas of the brain associated with specific functions.

The current methodology used to identify such regions is to compare, using various mathematical techniques, the elevation of oxygen consumption during a task with that used during a resting state. [Mitchell *et al.*, 2004] applied machine learning techniques to this problem, when considering the classification of the cognitive state of a human subject. Thus, in order to determine the elevation of oxygen consumption during a task, images acquired during a resting state are required. In order to keep the alternation between activity, a reference time-course is needed, where the resting and active states are embedded. In this paper we further consider the problem of identifying fMRI scans that have only been acquired during the “active” state, i.e. scans acquired during the duration when the human subject has performed the given task. The basic intuitions are that, if available, two-class classification should perform better; although not always. However, as is the case under consideration here, often we have some reasonable sampling of the positive examples; i.e. the distribution of positive examples can be estimated; while the negative examples are either non-existent or episodic; i.e. not necessarily representative.

Obtaining good results under this assumption is known to be quite challenging; nonetheless it is often the most realistic

assumption. For the fMRI classification described above, this problem is particularly non-trivial as we expect the data to be of very high dimension and extremely noisy, as the brain concurrently works on many given tasks. It is also quite natural to assume that there is only representative data of the task of interest; and not necessarily representative data of the negation of this task thus making the one-class learning techniques appropriate. In this work, we use two major one-class learning techniques - “bottleneck” or compression neural networks [Manevitz and Yousef, 2001] and a common version of the one-class Support Vector Machine (SVM) [Scholkopf *et al.*, 1999]. We point out that we use the entire brain slice, with no pre-filtering - i.e. the data is the entire slice, labeled with the task.

2 Experiments

The fMRI scans are of a volunteer¹ flexing their index finger on the right hand inside a MR-scanner while 12 image slices of the brain were obtained from a T2*-weighted MR scanner. The time-course reference of the flexing is built from the subject performing a sequence of 20 total actions and rests consisting of rest, flex, rest, . . . flex. Two hundred fMRI scans are taken over this sequence; ten for each action and rest. The individual fMRI images are dicom² format of size 128×128 . Each image is labelled as either 1 (active) or -1 (inactive).

Thus, in our data we have 100 positive and 100 negative images for each of the 12 slices. For the bottleneck neural network 80 positive samples were chosen randomly and presented for training and 40 samples, consisting of the remaining 20 positive and 20 random negative samples, were used for testing. This experiment was redone with ten independent random runs. The limitation to 20 negative samples out of a possible 100 was chosen to keep the testing fair between the positive and negative classes. We manually cropped the non-brain background from the scans; resulting in a slightly different input/output size for each slice of about 8,300 inputs and outputs. The compression percentage arising from the bottleneck was chosen by experimenting with different possible values. A uniform compression of about 60% gave the best results for the hidden layer. The irrelevant (non-brain) image data was cropped for each slice resulting in a slightly

¹Provided by Ola Friman [Friman, 2003].

²For information regarding dicom see <http://medical.nema.org/>

different input/output size for the network for each slice. The network was trained to the identity using 20 epochs on the above chosen data. Following training the network was used as a classification filter, with an input value being classified as positive if the error level was lower or equal to a threshold chosen heuristically from training and classified as negative. We used the same protocol in a one-class SVM. Additionally, we used the two-class SVM where we randomly selected 160 training images and the remaining 40 for testing. This was also repeated 10 times.

2.1 First Experimental Results

The obtained results are an average over all the slices. Each slice was averaged over 10 repeats where in each repeat a random split of training testing was selected. Both SVM classifiers were used in their default setting as set by the OSU-SVM 3.00 package³. In addition, the two-class SVM was used with the unnormalised data as we have experimentally found that with normalised data the overall results were significantly worse. In Table 1 we are able to observe that while the one-class SVM performs better with the RBF kernel, the two-class SVM is better with the linear kernel. In Table 2

Table 1: SVM Results (success).

Method	Linear kernel	RBF kernel
One-class SVM	49.12% $\pm$ 0.86%	59.18% $\pm$ 1.47%
Two-class SVM	68.06% $\pm$ 2.10%	44.70% $\pm$ 1.12%

we compare the one-class to two-class techniques. As initially expected we are able to observe that the two-class approaches outperform those of the one-class. The one-class SVM is slightly better than the bottleneck compression NN. We further analyse the statistics of the methods i.e. the sepa-

Table 2: Methods success results.

Method	Result on slices
BN - NN	56.19% $\pm$ 1.26%
One-class SVM	59.18% $\pm$ 1.47%
Two-class SVM	68.06% $\pm$ 2.10%

ration of the classified samples to their true classes. In Table 3⁴ we compute and show the statistics of the fMRI images of the Positive samples that were classified as positive, denoted as true-positive, and the positive samples that were classified as negative, denoted as false-negative and the statistics of the negative fMRI images samples that were classified as negative, denoted as true-negative, and those that were classified as positive, denoted as false-positive. We are able to observe in that the compression NN is able to find a higher rate of true-positive fMRI images than the one-class SVM and the two-class methods even though they have obtained an higher overall success rate. Also we observe with regard to the negative samples that the two-class methods perform better than

the one-class. This is expected as the one-class methods make no use of the negative samples and eminently will have a lower ability in classifying it. Additional experiments have

Table 3: Methods Statistics

Method	True-Positive	False-Negative	std
BN - NN	78.96%	21.04%	$\pm$ 3.15%
One-class SVM	72.83%	27.17%	$\pm$ 1.98%
Two-class SVM	71.55%	28.45%	$\pm$ 3.21%
	True-Negative	False-Positive	
BN - NN	33.42%	66.58%	$\pm$ 3.45%
One-class SVM	39.25%	60.75%	$\pm$ 3.25%
Two-class SVM	65.64%	54.46%	$\pm$ 3.02%

been performed on visual instead of motor tasks with very similar results.

3 Conclusions

We have found that one class classification can be done, even with the "noisy" data and with the full slices of the brain scan. Comparable results (about 58% accuracy) were obtained under both one-class SVM and Compression-Based NN techniques. For future work we would further investigate automated feature reduction as it might be fruitful, compare scans of the same individual across different fMRI sessions. The work presented here was for one individual. We intend to compare training and classification across individuals.

Acknowledgments

This collaboration was supported by the Caesarea Rothschild Foundation, and by the HIACS Research Center, the University of Haifa. The first author is funded by the European project LAVA num. IST-2001-34405 and PASCAL network of excellent num. IST-2002-506778. We also thank Ola Friman of Harvard Medical School for his generous sharing of his data.

References

- [Friman, 2003] Ola Friman. *Adaptive Analysis of Functional MRI Data*. PhD thesis, Linkoping Studies in Science and Technology, 2003.
- [Manevitz and Yousef, 2001] Larry Manevitz and Malik Yousef. One-class svms for document classification. *Journal of Machine Learning Research* 2, pages 139–154, 2001.
- [Mitchell et al., 2004] T.M. Mitchell, R. Hutchinson, R.S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 1-2:145–175, 2004.
- [Scholkopf et al., 1999] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. Technical Report 99-87, Microsoft Research., 1999.

³OSU SVMs Toolbox http://www.ece.osu.edu/~maj/osu_svm/

⁴std stands for Standard Deviation

Adaptive Support Vector Machine for Time-Varying Data Streams Using Martingale

Shen-Shyang Ho and Harry Wechsler

George Mason University

Department of Computer Science

4400 University Drive, Fairfax, VA 22030

{sho, wechsler}@gmu.edu

Abstract

A martingale framework is proposed to enable support vector machine (SVM) to adapt to time-varying data streams. The adaptive SVM is a one-pass incremental algorithm that (i) does not require a sliding window on the data stream, (ii) does not require monitoring the performance of the classifier as data points are streaming, and (iii) works well for high dimensional, multi-class data streams. Our experiments show that the novel adaptive SVM is effective at handling time-varying data streams simulated using both a synthetic dataset and a multi-class real dataset.

1 Introduction

In this paper we propose an efficient adaptive support vector machine (SVM) for time-varying data streams based on the martingale approach [Vovk *et al.*, 2003] and using adiabatic incremental learning [Cauwenberghs and Poggio, 2000]. When a new data point is observed, hypothesis testing decides whether any change has occurred. Once a change is detected, historical information about previous data is removed from the memory.

2 Change Detection Using Martingale

[Vovk *et al.*, 2003] introduced the idea of testing exchangeability online using martingale. The sequence of random variables $z_1, \dots, z_n$ is exchangeable if the joint distribution $p(z_1, \dots, z_n)$ is invariant under any permutation of the indices of the random variables. When a new labeled example, z_n , is observed, testing exchangeability of the sequence of examples $z_1, z_2, \dots, z_n$ consists of two main steps [Vovk *et al.*, 2003]:

1. Extract the randomized p-value for z_n :

$$p_n = \frac{\#\{i : \alpha_i > \alpha_n\} + \theta_n \#\{i : \alpha_i = \alpha_n\}}{n} \quad (1)$$

where α_i is the strangeness measure (for SVM, one can use either the Lagrange multipliers or the distances from the hyperplane) for z_i , $i = 1, 2, \dots, n$ and θ_n is a random number between 0 and 1.

2. Construct the randomized martingale:

$$M_n^{(\epsilon)} = \prod_{i=1}^n (\epsilon p_i^{\epsilon-1}) \quad (2)$$

where p_i are p-values, a constant $\epsilon \in [0, 1]$, and the initial martingale $M_1^{(\epsilon)} = 1$.

p_i and $M_i^{(\epsilon)}$, for $i = 2, \dots, n-1$, are computed using Step 1 and 2.

The exchangeability of a data sequence is a slightly weaker assumption compared to the randomness (i.i.d.) assumption. Hypothesis testing using the randomized martingale is suitable to check for the randomness of a data sequence which, in turn, can detect changes.

One tests the null hypothesis H_0 : “no change in the data stream” against the alternative hypothesis H_1 : “change in the data stream”. The martingale test continues as long as

$$0 < M_n^{(\epsilon)} < \lambda \quad (3)$$

One rejects the null hypothesis when $M_n^{(\epsilon)} \geq \lambda$.

The martingale test is justified using the Doob’s Maximal Inequality [Steele, 2001] which states that for a martingale $\{M_k\}$ and any $\lambda > 0$,

$$\lambda P \left(\max_{k \leq n} M_k \geq \lambda \right) \leq E(M_n) \quad (4)$$

Hence, if $E(M_n) = E(M_1) = 1$, then

$$P \left(\max_{k \leq n} M_k \geq \lambda \right) \leq \frac{1}{\lambda} \quad (5)$$

This inequality means that it is unlikely for M_n to have a high value. Based on this inequality, λ is appropriately set to reject the null hypothesis.

3 Experiments

Experiments are first conducted on a binary-class two-dimensional rotating hyperplane data stream [Hulten *et al.*, 2001]. To show the performance of the adaptive SVM on high dimensional multi-class data stream, experiments are carried out using a modified three-digit data stream based on the USPS handwritten digit dataset [LeCun *et al.*, 1989]. For the SVMs, $C = 10$ and the Gaussian kernel are used.

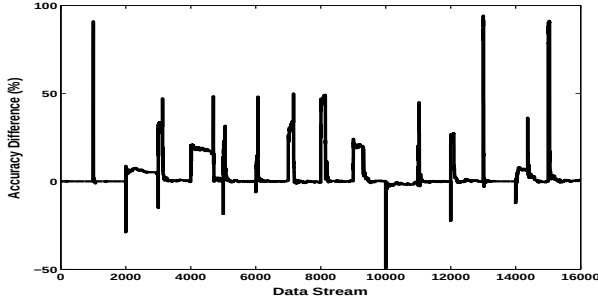


Figure 1: The difference between baseline accuracy and the accuracy of the adaptive SVM. Negative values imply better performance than the baseline accuracy.

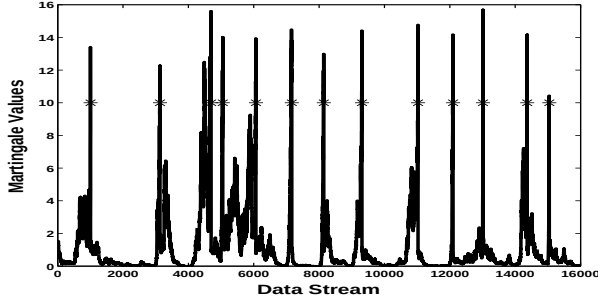


Figure 2: The martingale values of the data stream. Detected change points (denoted *): 1007, 3133, 4697, 5054, 6068, 7165, 8144, 9312, 11029, 12099, 13015, 14368, 15045. Miss detections: 2001, 10001. The median of the time between true change points and their corresponding detected change points is 99 time units.

A. 2-D Rotating Hyperplane Synthetic Dataset

For the two-dimensional rotating hyperplane data stream, the sequence consists of 16,000 data points with changes in the data distribution occurring at points $(1000 \times i) + 1$, for $i = 1, 2, \dots, 15$. For each segment of a steady data distribution, 500 points from the same data distribution are used to assess the performance of the SVM classifier.

The adaptive SVMs using $\lambda = 16$, for the martingale test (3), displays very good performance as shown in Figure 1. The difference in accuracy from baseline accuracy (performance of the classifier that adapts to the given true change points) is near zero most of the time. Short bursts of accuracy discrepancy occur between true change points and their corresponding detected change points. Longer accuracy discrepancies happen when a significant change is detected late (e.g. change point 4001) as shown in Figure 1. In Figure 2, we observe that the martingale value is small and stable when no change in the data distribution is detected. The martingale value increases abruptly when change is suspected.

B. Three-digit Data Stream

The USPS handwritten digits dataset, which consists of 10 classes of dimension 256 and includes 7,291 data points, is modified to form a data stream as follows. There are four different data segments. Each segment draws from a fixed

Segment	Digit 1	Digit 2	Digit 3	Total	Change Point
1	597/359 (0)	502/264 (1)	731/198 (2)	1830/821	1831
2	597/359 (0)	658/166 (3)	652/200 (4)	1907/725	3738
3	503/264 (1)	556/160 (5)	664/170 (6)	1723/594	5461
4	645/147 (7)	542/166 (8)	644/177 (9)	1831/490	-

Table 1: **Three-Digit Data Stream: TR/TS (D):** For the true digit class D, TR and TS are the number of data points used for segment construction and for testing respectively.

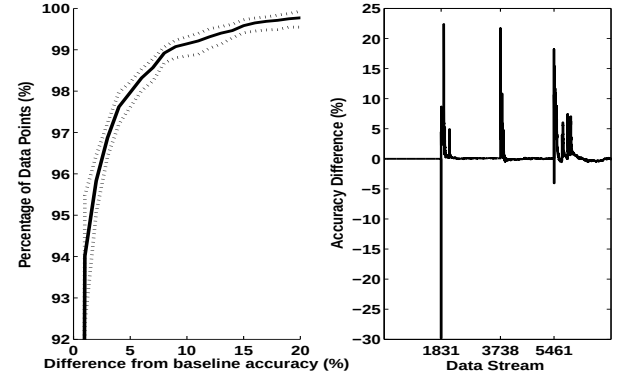


Figure 3: Left Graph: The proportion of data points with the difference from the baseline accuracy less than or equal to some percentage value. The maximum, minimum and mean percentage of data points at each accuracy difference are shown. Right Graph: The mean difference between the baseline accuracy and the accuracy of the adaptive SVM.

set of three different digits in a random fashion. The three-digit set changes from one segment to the next. The composition of the data stream and ground truth for the change points are summarized in Table 1. Ten simulations were performed on the simulated data stream using the one-against-the-rest multi-class SVM and the martingale test with $\lambda = 10$.

For the three-class data stream there are three martingale values that have to be computed at each point in order to detect change. When one of the martingale values is greater than λ , the SVM will detect and adapt to the change. One can see from Figure 3 that the adaptive SVM performs extremely well. About 94% of the data points have a difference from baseline accuracy less than or equal to 1%. There are no miss detections for all the ten simulations.

References

- [Cauwenberghs and Poggio, 2000] G. Cauwenberghs and T. Poggio. Incremental support vector machine learning. In *Advances in Neural Information Processing Systems 13*, pages 409–415. MIT Press, 2000.
- [Hulten et al., 2001] Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *KDD*, pages 97–106, 2001.
- [LeCun et al., 1989] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. J. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.
- [Steele, 2001] M. Steele. *Stochastic Calculus and Financial Applications*. Springer Verlag, 2001.
- [Vovk et al., 2003] Vladimir Vovk, Ilya Nourtdinov, and Alexander Gammerman. Testing exchangeability on-line. In Tom Fawcett and Nina Mishra, editors, *ICML*, pages 768–775. AAAI Press, 2003.

Automatic Hypertext Keyphrase Detection

Daniel Kelleher, Saturnino Luz

University of Dublin, Trinity College

Department of Computer Science, O'Reilly Institute, Trinity College, Dublin 2, Ireland

{dkellehe, luzs}@cs.tcd.ie

Abstract

This paper describes initial experiments in applying knowledge derived from hypertext structure to domain-specific automatic keyphrase extraction. It is found that hyperlink information can improve the effectiveness of automatic keyphrase extraction by 50%. However, the primary goal of this project is to apply similar techniques to information retrieval tasks such as web searching. These initial results show promise for the applicability of these techniques to more far-reaching tasks.

1 Introduction

The associative nature of the Web has been under-exploited so far. This paper describes the initial steps taken towards developing a framework that will take advantage of the associative hyperlink structure of the web to improve information retrieval and document classification. In this project, the automatic keyphrase extraction program, KEA, was adapted for use on web corpora and an extra feature was added that takes advantage of any semantic similarity that may exist between linked web documents.

1.1 KEA

KEA [Witten *et al.*, 1999] is an automatic keyphrase extraction algorithm for documents based on a domain-specific machine-learning model [Frank *et al.*, 1999]. It compiles a list of phrases from a training set of documents with annotated keyphrases and generates a naïve Bayes classifier based on two default features of these phrases:

- TFXIDF [Salton, 1988], which is a value based on the ratio of probability of a phrase appearing in the current document and the probability that it appears in any document, and is given by:

$$TF \times IDF(P, D) = \frac{\text{No. of occur. of } P \text{ in } D}{\text{No. of phrases in } D} \times \log\left(\frac{\text{total number of documents}}{\text{number of docs that contain } P}\right)$$

- Distance - the ratio of the number of words before the first appearance of the phrase in the document and the total number of words in the document.

The classifier is then used to extract potential keyphrases from a test set of documents.

1.2 Hypertext

The concept behind hypertext is that text content (or other media, in fact) is connected by associations or 'links' from document to document, forming a directed graph structure. The associations will usually (although not always) be based on some semantic similarity or relevance (of varying strength) between two documents.

2 Method

The link structure of web documents is included in KEA by introducing the "Semantic Ratio" (SR) feature. SR is similar to the TFXIDF feature, in that it is a frequency ratio. However, the SR of a phrase is calculated by dividing the number of occurrences of that phrase in the current document by the number of times it occurs in all documents directly linked to that document (i.e. those that are the targets of hyperlinks in the document).

$$SR(P, D) = \frac{\text{Frequency of } P \text{ in } D}{\text{Frequency of } P \text{ in documents linked to } D}$$

The reasoning behind including this feature in KEA is based on the intuition that the content of a web document is frequently semantically related to its neighbours (in the context of a graph structure, in other words, the documents linked to it) and that the subject matter (identified by the keyphrases) of the document is therefore in some way relative to their contents. The inclusion of the SR feature is a first step in testing, and subsequently, modelling this intuition.

A low SR value (< 1) indicates that a potential keyphrase occurs more frequently in the document's neighbours than in the document itself. The higher the SR value, therefore, the more specific the phrase to this document, relative to its immediate surroundings. Note that this is different to the TFXIDF score as only a subset of the documents are used to compute it, namely those documents that form a localised subgraph with paths of length 1 from the original document.

3 Testing

The new version of KEA with the SR feature included was tested on four web corpora taken from the WWW. These corpora were chosen because a sufficient number of docu-

ments in each site contained annotated keyphrases in the form of the Meta Keyword HTML tag and were therefore suitable for empirical tests on the accuracy of automatic keyphrase extraction.

The corpus was then split into a training set and a test set of roughly equal size, with the restriction that no document in the test set should be linked to a document in the training set. The new KEA algorithm (called KEAWeb) was then trained on the training set and then performed automatic keyphrase extraction on the test set, and the average number of correct keyphrases found in each corpus was recorded and is presented below.

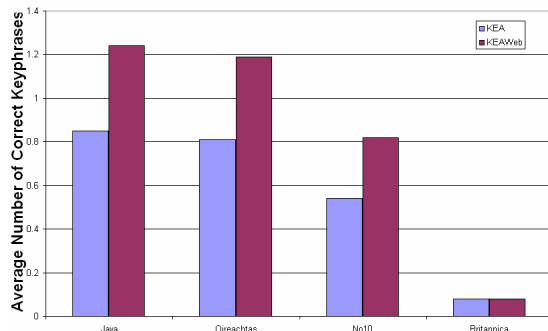


Figure 1: Comparison of KEA and KEA Web

4 Results

Figure 1 shows that the inclusion of the SR feature in the naïve Bayes classifier improves the success of the KEA algorithm by between 45% and 52% in three out of four of the test corpora. KEA performs badly on the Britannica corpus whether the SR feature is included or not. This is due to a number of factors based on the fact that the license-free version of the site severely restricts both the size and the number of links in a document. It can therefore be safely assumed that the corpus is an atypical example of a web site and that the three other corpora are more representative examples.

Initial analysis of the SR distribution in keyphrases suggests that phrases with extremal SR values are more likely to be keyphrases. In other words, phrases that appear frequently in surrounding documents (low SR) have high relevance for the document in question. Also, phrases that occur very rarely in surrounding documents (high SR) will also have high relevance, suggesting that they indicate a topic that is specific to the current document.

5 Conclusions & Future Work

The SR feature shows initial promise as an indicator of semantic relations between linked pages. The latent semantic information that causes the improvement in KEA is expected to be transferable to other domains, particularly Web searching. Future work therefore involves (in order of immediacy):

- Further analysis of the distribution of SR in keyphrases and adapting the KEA algorithm to use a more suitable classifier than the naïve Bayes. It is clear that the SR and TFxIDF features are not independent, as the naïve Bayes classifier requires. Furthermore, while phrases with low TFxIDF are generally less likely to be keyphrases, this is not typically the case with SR. Therefore, the independence assumption will, in some cases result in a less accurate classification. Also, the normal density function assumed by KEA may be suboptimal for the continuous features used in this project. Current work therefore involves testing variations on the naïve Bayes classifier (such as the ‘flexible’ naïve Bayes described in [John *et al.*, 1995]).
- Experimentation with the SR feature is required in order to determine if a more suitable feature or number of features exist. In addition, the SR feature will be more generalised to take into account more distant documents than those directly related to the document in question, perhaps including a link-weighting or spreading-activation mechanism for retrieving the documents.
- The SR distribution will be further analysed and used to adapt a term-frequency-based document retrieval program to the Web and to improve existing web search engines by assigning scores to documents according to the probability that a given search term is a keyphrase of that document. These projects will be applied initially in a domain-specific, supervised learning environment, and ultimately in a more general and universal environment.

The initial results mentioned in this paper lend encouragement to the hope that a system based on hyperlink analysis can make significant improvements to existing Web search technology.

References

- [Frank *et al.*, 1999] Frank E., Paynter G.W., Witten I.H., Gutwin C. and Nevill-Manning C.G. (1999) Domain-specific keyphrase extraction *Proc. Sixteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA, pp. 668-673.
- [John *et al.*, 1999] John, G.H., Langley, P., (1995) Estimating Continuous Distributions in Bayesian Classifiers. *Proc. Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann Publishers, San Mateo.
- [Salton, 1988] Salton, Gerard. Automatic text processing: the transformation, analysis, and retrieval of information by computer. Reading, Mass. Wokingham : Addison-Wesley, 1988.
- [Witten *et al.*, 1999] Witten I.H., Paynter G.W., Frank E., Gutwin C. and Nevill-Manning C.G. (1999) KEA: Practical automatic keyphrase extraction. *Proc. DL '99*, pp. 254-256. (Poster presentation)

Relational Learning for Email Task Management*

Rinat Khossainov and Nicholas Kushmerick

Department of Computer Science
University College Dublin, Ireland
{rinat, nick}@ucd.ie

1 Introduction and Background

Today's email clients were designed for yesterday's email. Originally, email was merely a communication medium. Today, email has become a "habitat" [Ducheneaut and Bellotti, 2001]—an environment where users engage in a variety of complex *activities*. Our goal is to develop automated techniques to help people manage complex activities or *tasks* in email. In many cases, such activities manifest the user's participation in various structured processes or workflows. The central challenge is that most processes are distributed over multiple emails, yet email clients are designed mainly to manipulate individual messages.

A task-oriented email client would allow the user to manage activities rather than separate messages. For instance, the user would be able to quickly inquire about the current status of unfinished e-commerce transactions or check the outcome of recent project meetings. Some process steps could be automated, such as automatically sending reminders for earlier user's requests. Similarly, the email client could remind the user when her/his input is required in some activity.

Previous work in this area has mainly focused on two distinct problems: finding related messages and semantic message analysis. The goal of *finding related messages* is to group emails according to tasks and possibly establish conversational links between emails in a task (e.g. extract a task from email given a seed message [Dredze, 2005]). Note that tasks need not correspond to folders (folders can be orthogonal to tasks); and conversations need not correspond to syntactic threads (users can use the "Reply" button or the same subject to start a semantically new conversation).

Semantic message analysis involves generating metadata for individual messages in a task that provides a link between the messages and the changes in the status of the underlying process, or the actions of the user in the underlying workflow. For example, [Cohen *et al.*, 2004] proposed machine learning methods to classify emails according to the intent of the sender, expressed in an ontology of "*email speech acts*".

Our key innovation compared to related work is that we exploit the *relational structure* of these two tasks. The idea is that related messages in a task provide a valuable context that can be used for semantic message analysis. Similarly, the activity-related metadata in separate messages can pro-

vide relational clues that can be used to establish links between emails and group them into tasks. Instead of treating these two problems separately, we propose a *synergetic approach* where identifying related emails is used to assist semantic message analysis and vice versa.

Our contributions are as follows:

(1) We propose a new method for identifying relations between emails, based on pair-wise message similarity. We extend the similarity function to take into account available structured information in email.

(2) We propose a relational learning approach [Neville and Jensen, 2000] to email task management. We investigate how (a) features of related emails in the same task can assist with classification of speech acts, and how (b) information about speech acts can assist with finding related messages. Combining these two methods yields an iterative relational algorithm for speech act classification and relation identification.

(3) We evaluate our methods on a real-life email corpus.

2 Problem Decomposition and Email Corpora

In a non-relational approach, we would use the content of a message to assign speech acts, and some content similarity between messages to identify relations. In a relational approach to speech act classification, we can use both the message content and features of the related messages from the same task. For example, if a message is a response to a meeting proposal, then it is more likely to be a meeting confirmation or refusal. Similarly, we can use messages' speech acts to improve relations identification, e.g. a request followed by a delivery are more likely to be related than two requests.

Therefore, we can identify four sub-problems: (P1) find relations in email using content similarity only (i.e. without using messages' speech acts); (P2) classify messages into speech acts (semantic message analysis) using only the message content (i.e. without using information about the related messages); (P3) use the identified related emails to improve the quality of speech acts classification for a given message; (P4) use the messages' speech acts to improve identification of relations (links) between emails. These four sub-problems can be combined into a synergetic approach to task management based on an iterative relational classification algorithm illustrated in Figure 1.

We used the PW CALO email corpus [Cohen *et al.*, 2004] for our study. It was generated during a 4-day exercise conducted at SRI specifically to generate an email corpus. During

*This research was funded by the Science Foundation Ireland and the US Office of Naval Research.

```

1: Identify initial relations (P1)
2: Generate initial speech acts (P2)
loop
3: Use related emails in the task to clarify speech acts (P3)
4: Use speech acts to clarify relations between emails (P4)
5: Update messages relations
6: Update messages speech acts
end loop

```

Figure 1: Iterative relational algorithm for task management

Table 1: Identifying relations		Precision	Recall	F1
P1	No time decay, thresh. prune	0.83	0.80	0.81
	Time decay, thresh. prune	0.84	0.80	0.82
	No time decay, threads prune	0.83	0.81	0.82
	Time decay, threads prune	0.84	0.82	0.83
P4	Using speech acts	0.91	0.80	0.85

this time a group of six people assumed different work roles (project leader, finance manager, researcher, etc) and performed a number of activities. Each email has been manually annotated with labels linking it to other emails and also with labels showing the intent of the sender, expressed in a verb-noun ontology of “*email speech acts*” [Cohen *et al.*, 2004]. Examples of speech acts are “Propose meeting”, “Deliver information”. For this study we only use 5 most frequent verbs (“Propose”, “Request”, “Deliver”, “Commit”, “Amend”) as speech acts. To perform experiments, we need to ensure our training and testing sets are unrelated. So we generated 2 non-overlapping corpora with messages received by 2 different users (“User 1”, 160 emails; and “User 2”, 33 emails).

3 Solutions and Results

Identifying Relations without Using Speech Acts (P1). For each email in the corpus, we find the most similar preceding (in time) email using a pair-wise message similarity. Our similarity functions takes into account not only the textual similarity between messages, but also the available structured information in email, such as send dates, and message subjects. The textual similarity is defined as the TF/IDF cosine similarity between email texts. However, the terms appearing in the subject get a higher weight, since people often summarise email content in the subject making subject terms more important. Similarly, related messages tend to be sent around the same time. So, two messages with a large send time difference are less likely to be related. We use the following formula: $Sim(m_1, m_2) = Cosine_Sim(m_1, m_2) * \exp(-\alpha * Norm_time_diff(m_1, m_2))$, where $Cosine_Sim$ is the cosine message similarity, $Norm_time_diff(m_1, m_2)$ is the time difference between messages divided by the maximum time difference, and α is a time decay parameter.

There may be multiple pairs of messages with non-zero similarity in a corpus, however, not all are actually related. Hence, we would like to be able to prune the links suggested by the similarity function. One way is to use some threshold value: if the similarity is below the threshold, the messages are not related. Another way is to use email threads: messages from different threads are not related. Table 1 compares different methods on the “User 1” corpus.

Identifying Relations Using Speech Acts (P4). We treat the problem of finding related messages using speech acts as a supervised learning task. We assume that we have access to a training set, which provides the correct labels for both speech acts and message relations. The goal is to use this information to improve our performance on an unseen email corpus. From the given labelled email corpus, we produce a set of training instances as follows. For each message in the corpus (child), we identify the most similar preceding message (parent) using the previously defined similarity function. For each such pair of messages, we create one training instance with one numeric feature for the similarity between messages, and two subsets of binary features for each possible speech act (10 features in total). The first binary subset is filled with speech acts of the parent message: 1 if the message has this speech act, 0 otherwise. The second binary subset is filled with speech acts of the child message. The class label for the instance is positive if the corresponding messages are related and negative otherwise. The resulting classifier can then be used to identify links in an unseen email corpus.

To evaluate the potential for improvement from using speech acts, we tried to train and test a classifier on the same “User 1” corpus. We use the SMO as our classification algorithm [Platt, 1999]. As shown in Table 1, using speech acts worked here as a more effective pruning method resulting in the increase in precision with only marginal loss in recall.

Classifying Speech Acts without Related Messages (P2). As in the previous case, we treat the problem of email speech act classification as a supervised learning task. We use the standard text classification methods with bag-of-words document representations similar to [Cohen *et al.*, 2004], and SMO as the classification algorithm.

Classifying Speech Acts Using Related Messages (P3). We adopt here the relational learning terminology from [Neville and Jensen, 2000]. Each email message is represented by a set of features: *intrinsic features*, derived from the content of the given message; and *extrinsic features* derived from the properties of related messages in the same task. To represent the intrinsic features of a message, we use the raw term frequencies as in P2. To represent the extrinsic features of a message, we use the speech acts of related messages. We want to know whether speech acts of “surrounding” messages can help in classifying speech acts of a given message. For each speech act, we produce a separate binary classification problem where the goal is to identify whether the message has this act or not.

Each message can be viewed as a response to its parent message and as a cause for its children messages. In addition to looking at the immediate ancestors and descendants of a message, we can also include features from several “generations” of ancestors and descendants (e.g. parents, grandparents, children, grandchildren). For each “generation” of related ancestor and descendant messages, we use a separate set of extrinsic features with one feature per each possible speech act. The number of generations included into extrinsic features is regulated by the *depth of lookup* parameters: one for ancestor messages and one for descendant messages (0 lookup depth means we use only intrinsic features).

We evaluated speech act classification using the human-

Table 2: Speech acts classification				
Ancest./Descend. lookup	0/0	0/1	1/0	1/1
Amend (p=0.43)	0.40	0.36	0.45 v	0.40
Commit (p=0.05)	0.17	0.20	0.28	0.37 v
Deliver (p=0.24)	0.22	0.29 v	0.21	0.29 v
Propose (p=0.21)	0.08	0.11	0.13 v	0.15
Request (p=0.05)	0.09	0.17	0.14	0.31 v

for all speech acts a **do**

Train C_a on the training set to classify speech act a using only intrinsic features

Train R_a on the training set to classify speech act a using intrinsic+extrinsic features

end for

Train L on the training set to classify email links

/*Problem 1*/

Set relations in the test set using similarity function

/*Iterative classification*/

for $Iteration = 1 \dots I$ **do**

/*Problem 2*/

Use classifiers C_a to set speech acts in the test set

/*Problem 3*/

$Threshold = 1$

for $Subiteration = 1 \dots K$ **do**

for all messages m in the test set **do**

for all speech acts a **do**

Obtain confidence for “ m has a ” using R_a

Obtain confidence for “ m has no a ” using R_a

end for

end for

For all cases where confidence for “ m has/has no a ” is greater than $Threshold$ update speech acts of m

$Threshold = Threshold / 2$

Evaluate performance for speech acts

end for

/*Problem 4*/

Use L to find links between emails in the test set

Evaluate performance for relations

end for

Figure 2: Iterative relational algorithm (detailed version)

annotated (correct) relations between messages and the correct speech acts for related messages on the “User 1” corpus. Notice that using the correct speech acts for related messages does not mean that we use the class label of an instance among its features. Each message uses only the speech acts of the related messages, but not its own speech acts. Classification accuracy is not a good measure of performance on imbalanced data sets, so we use the Kappa statistics instead [Cohen *et al.*, 2004]. The results in Table 2 are obtained in 5-fold cross-validations repeated 10 times for statistical significance testing (paired T-test). “V” marks statistically significant improvements over the *base line* (both lookup depths are 0).

Iterative Algorithm for Task Management (P1 + P2 + P3 + P4). The results for P3 and P4 have demonstrated promise for our synergetic approach to task management. Therefore, here we combine the described methods for solving P1–P4 into the algorithm shown in Figure 2 (which is a detailed version of the algorithm in Figure 1).

In our experiments, we used the “User 1” corpus as the training set and the “User 2” corpus as the test set. To ob-

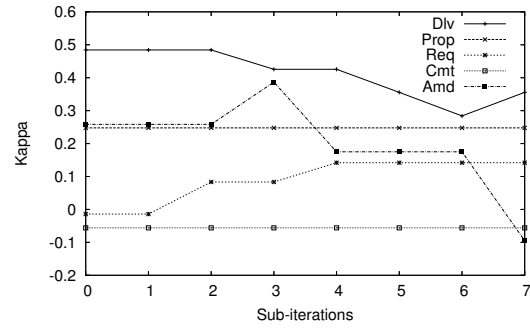


Figure 3: Speech acts classification, Iteration 1

tain confidence scores for SMO, we used the distance from the hyper-plane normalised over all test instances. We use the similarity function with time decay and threshold-based pruning to identify the initial links between messages (P1). We repeated the inner speech act classification loop 10 times ($K = 10$) and the outer iteration loop 2 times ($I = 2$).

The initial links identification resulted in precision = recall = F1 = 0.95. It improved after the first iteration to precision=1.0; recall=0.95; F1=0.98, and remained the same after the second iteration. Figure 3 shows how the speech acts classification performance was changing during the first main iteration. Once the links improved after the first iteration, we were able to further improve the performance for the “Request” speech act at the second iteration to Kappa=0.23.

Discussion. Our experiments demonstrated that: (1) structured features in email, such as message subject and send dates, can be very useful for identification of related messages and grouping them into email tasks; (2) the properties of related messages in the same task can be used to improve the semantic message analysis. In particular, the features of related messages in a task can improve the performance of the email speech acts classification; (3) the semantic metadata in messages can be used to improve the quality of task identification. In particular, taking into account speech acts of messages improves identification of links between emails.

Finally, our combined iterative classification algorithm was able to simultaneously improve performance on both speech acts and message relations. These results provide a good empirical evidence in favour of the proposed synergetic approach to email task management.

References

- [Cohen *et al.*, 2004] W. Cohen, V. Carvalho, and T. Mitchell. Learning to classify email into “speech acts”. In *Empirical Methods in Natural Language Processing*, 2004.
- [Dredze, 2005] M. Dredze. Taxie: Automatically identifying tasks in email. In *Manuscript avail. from the author*, 2005.
- [Ducheneaut and Bellotti, 2001] N. Ducheneaut and V. Bellotti. Email as habitat: An exploration of embedded personal information management. *ACM Interactions*, 8(1), 2001.
- [Neville and Jensen, 2000] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statistical Models from Relational Data*, 2000.
- [Platt, 1999] John C. Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimization*. MIT Press, 1999.

A Fast Normalized Maximum Likelihood Algorithm for Multinomial Data

Petri Kontkanen, Petri Myllymäki

Complex Systems Computation Group (CoSCo)

Helsinki Institute for Information Technology (HIIT)

University of Helsinki & Helsinki University of Technology

P.O. Box 9800, FIN-02015 HUT, Finland.

{Firstname}. {Lastname}@hiit.fi

Abstract

Stochastic complexity of a data set is defined as the shortest possible code length for the data obtainable by using some fixed set of models. This measure is of great theoretical and practical importance as a tool for tasks such as model selection or data clustering. In the case of multinomial data, computing the modern version of stochastic complexity, defined as the Normalized Maximum Likelihood (NML) criterion, requires computing a sum with an exponential number of terms. Furthermore, in order to apply NML in practice, one often needs to compute a whole table of these exponential sums. In our previous work, we were able to compute this table by a recursive algorithm. The purpose of this paper is to significantly improve the time complexity of this algorithm. The techniques used here are based on the discrete Fourier transform and the convolution theorem.

1 Introduction

The *Minimum Description Length (MDL)* principle developed by Rissanen [Rissanen, 1978; 1987; 1996] offers a well-founded theoretical formalization of statistical modeling. The main idea of this principle is to represent a set of models (model class) by a single model imitating the behaviour of any model in the class. Such representative models are called *universal*. The universal model itself does not have to belong to the model class as often is the case.

From a computer science viewpoint, the fundamental idea of the MDL principle is *compression of data*. That is, given some sample data, the task is to find a description or *code* of the data such that this description uses less symbols than it takes to describe the data literally. Intuitively speaking, this approach can in principle be argued to produce the best possible model of the problem domain, since in order to be able to produce the most efficient coding of data, one must capture all the regularities present in the domain.

The MDL principle has gone through several evolutionary steps during the last two decades. For example, the early realization of the MDL principle, the two-part code MDL [Rissanen, 1978], takes the same form as the Bayesian BIC criterion [Schwarz, 1978], which has led some people to incor-

rectly believe that MDL and BIC are equivalent. The latest instantiation of the MDL is *not* directly related to BIC, but to the formalization described in [Rissanen, 1996]. Unlike Bayesian and many other approaches, the modern MDL principle does not assume that the chosen model class is correct. It even says that there is no such thing as a true model or model class, as acknowledged by many practitioners. The model class is only used as a technical device for constructing an efficient code. For discussions on the theoretical motivations behind the modern definition of the MDL see, e.g., [Rissanen, 1996; Merhav and Feder, 1998; Barron *et al.*, 1998; Grünwald, 1998; Rissanen, 1999; Xie and Barron, 2000; Rissanen, 2001].

The most important notion of the MDL principle is the *Stochastic Complexity (SC)*, which is defined as the shortest description length of a given data relative to a model class $\mathcal{M}$. The modern definition of SC is based on the Normalized Maximum Likelihood (NML) code [Shtarkov, 1987]. Unfortunately, with multinomial data this code involves a sum over all the possible data matrices of certain length. Computing this sum, usually called the *regret*, is obviously exponential. Therefore, practical applications of the NML have been quite rare,

In our previous work [Kontkanen *et al.*, 2003; 2005], we presented a polynomial time (quadratic) method to compute the regret. In this paper we improve our previous results and show how mathematical techniques such as discrete Fourier transform and convolution can be used in regret computation. The idea of applying these techniques for computing a single regret term was first suggested in [Koivisto, 2004], but as discussed in [Kontkanen *et al.*, 2005], in order to apply NML to practical tasks such as clustering, a whole table of regret terms is needed. We will present here an efficient algorithm for this specific task. For a more detailed discussion of this work, see [Kontkanen and Myllymäki, 2005].

2 NML for Multinomial Data

The most important notion of the MDL is the *Stochastic Complexity (SC)*. Intuitively, stochastic complexity is defined as the shortest description length of a given data relative to a model class. To formalize things, let us start with a definition of a model class. Consider a set $\Theta \in \mathbb{R}^d$, where d is a positive integer. A class of parametric distributions indexed by the elements of Θ is called a *model class*. That is, a model

class $\mathcal{M}$ is defined as

$$\mathcal{M} = \{P(\cdot | \theta) : \theta \in \Theta\}. \quad (1)$$

Consider now a discrete data set (or matrix) $\mathbf{x}^N = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ of N outcomes, where each outcome $\mathbf{x}_j$ is an element of the set $\mathcal{X}$ consisting of all the vectors of the form $(a_1, \dots, a_m)$, where each variable (or attribute) a_i takes on values $v \in \{1, \dots, n_i\}$. Given a model class $\mathcal{M}$, the *Normalized Maximum Likelihood (NML)* distribution [Shtarkov, 1987] is defined as

$$P_{NML}(\mathbf{x}^N | \mathcal{M}) = \frac{P(\mathbf{x}^N | \hat{\theta}(\mathbf{x}^N), \mathcal{M})}{\mathcal{R}_{\mathcal{M}}^N}, \quad (2)$$

where $\hat{\theta}(\mathbf{x}^N)$ denotes the *maximum likelihood* estimate of data $\mathbf{x}^N$, and $\mathcal{R}_{\mathcal{M}}^N$ is given by

$$\mathcal{R}_{\mathcal{M}}^N = \sum_{\mathbf{x}^N} P(\mathbf{x}^N | \hat{\theta}(\mathbf{x}^N), \mathcal{M}), \quad (3)$$

and the sum goes over all the possible data matrices of size N . The term $\mathcal{R}_{\mathcal{M}}^N$ is called the *regret*. The definition (2) is intuitively very appealing: every data matrix is modeled using its own maximum likelihood (i.e., best fit) model, and then a penalty for the complexity of the model class $\mathcal{M}$ is added to normalize the distribution.

The stochastic complexity of a data set $\mathbf{x}^N$ with respect to a model class $\mathcal{M}$ can now be defined as the negative logarithm of (2), i.e.,

$$SC(\mathbf{x}^N | \mathcal{M}) = -\log \frac{P(\mathbf{x}^N | \hat{\theta}(\mathbf{x}^N), \mathcal{M})}{\mathcal{R}_{\mathcal{M}}^N} \quad (4)$$

$$= -\log P(\mathbf{x}^N | \hat{\theta}(\mathbf{x}^N), \mathcal{M}) + \log \mathcal{R}_{\mathcal{M}}^N. \quad (5)$$

As in [Kontkanen *et al.*, 2005], in the sequel we focus on a multi-dimensional model class suitable for cluster analysis. The selected model class has also been successfully applied to mixture modeling [Kontkanen *et al.*, 1996], case-based reasoning [Kontkanen *et al.*, 1998], Naive Bayes classification [Grünwald *et al.*, 1998; Kontkanen *et al.*, 2000b] and data visualization [Kontkanen *et al.*, 2000a].

Let us assume that we have m variables, $(a_1, \dots, a_m)$, and we also assume the existence of a special variable c (which can be chosen to be one of the variables in our data or it can be latent). Furthermore, given the value of c , the variables $(a_1, \dots, a_m)$ are assumed to be independent. The resulting model class is denoted by $\mathcal{M}_T$. Suppose the special variable c has K values and each a_i has n_i values. The NML distribution for the model class $\mathcal{M}_T$ is now

$$P_{NML}(\mathbf{x}^N | \mathcal{M}_T) = \left[\prod_{k=1}^K \left(\frac{h_k}{N} \right)^{h_k} \prod_{i=1}^m \prod_{k=1}^K \prod_{v=1}^{n_i} \left(\frac{f_{ikv}}{h_k} \right)^{f_{ikv}} \right] \cdot \frac{1}{\mathcal{R}_{\mathcal{M}_T, K}^N}, \quad (6)$$

where h_k is the number of times c has value k in $\mathbf{x}^N$, f_{ikv} is the number of times a_i has value v when $c = k$, and $\mathcal{R}_{\mathcal{M}_T, K}^N$ is the regret term. In [Kontkanen *et al.*, 2005] it was proven

that an efficient way to compute the regret term is via the following recursive formula:

$$\mathcal{R}_{\mathcal{M}_T, K}^N = \sum_{r=0}^N \frac{N!}{r!(N-r)!} \left(\frac{r}{N} \right)^r \left(\frac{N-r}{N} \right)^{N-r} \cdot \mathcal{R}_{\mathcal{M}_T, k_1}^r \cdot \mathcal{R}_{\mathcal{M}_T, k_2}^{N-r}, \quad (7)$$

where $k_1 + k_2 = K$.

As discussed in [Kontkanen *et al.*, 2005], in order to apply NML to the clustering problem, we need to compute a whole table of regret terms. This table consists of the terms $\mathcal{R}_{\mathcal{M}_T, k}^n$ for $n = 0, \dots, N$ and $k = 1, \dots, K$, where K is the maximum number of clusters.

The procedure of computing the regret table starts by filling the first column, i.e., the case $k = 1$, which is trivial (see [Kontkanen *et al.*, 2005]). To compute the column k , for $k = 2, \dots, K$, the recursive formula (7) can be used by choosing $k_1 = k - 1$, $k_2 = 1$. The time complexity of filling the whole table is $\mathcal{O}(K \cdot N^2)$. For more details, see [Kontkanen *et al.*, 2005; Kontkanen and Myllymäki, 2005].

In practice, the quadratic dependency on the size of data limits the applicability of NML to small or moderate size data sets. In the next section, we will present a novel, significantly more efficient method for computing the regret table.

3 The Fast NML Algorithm

In this section we will derive a very efficient algorithm for the regret table computation. The new method is based on the Fast Fourier Transform algorithm. As mentioned in the previous section, the calculation of the first column of the regret table is trivial. Therefore, we only need to consider the case of calculating the column k given the first $k - 1$ columns. Let us define two sequences $\mathbf{a}$ and $\mathbf{b}$ by

$$a_n = \frac{n^n}{n!} \mathcal{R}_{\mathcal{M}_T, k-1}^n, \quad b_n = \frac{n^n}{n!} \mathcal{R}_{\mathcal{M}_T, 1}^n, \quad (8)$$

for $n = 0, \dots, N$. Evaluating the convolution of $\mathbf{a}$ and $\mathbf{b}$ gives

$$(\mathbf{a} * \mathbf{b})_n = \sum_{h=0}^n \frac{h^h}{h!} \mathcal{R}_{\mathcal{M}_T, k-1}^h \frac{(n-h)^{n-h}}{(n-h)!} \mathcal{R}_{\mathcal{M}_T, 1}^{n-h} \quad (9)$$

$$= \frac{n^n}{n!} \sum_{h=0}^n \frac{n!}{h!(n-h)!} \left(\frac{h}{n} \right)^h \left(\frac{n-h}{n} \right)^{n-h} \cdot \mathcal{R}_{\mathcal{M}_T, k-1}^h \mathcal{R}_{\mathcal{M}_T, 1}^{n-h} \quad (10)$$

$$= \frac{n^n}{n!} \mathcal{R}_{\mathcal{M}_T, k}^n, \quad (11)$$

where the last equality follows from the recursion formula (7). This derivation shows that the column k can be computed by first evaluating the convolution (11), and then multiplying each term by $n!/n^n$.

The standard *convolution theorem* states that convolutions can be evaluated via the (discrete) Fourier transform, which in turn can be computed efficiently with the Fast Fourier Transform algorithm (see [Kontkanen and Myllymäki, 2005] for details). It follows that the time complexity of computing the

whole regret table drops to $\mathcal{O}(N \log N \cdot K)$. This is a major improvement over $\mathcal{O}(N^2 \cdot K)$ obtained by the recursion method of Section 2.

4 Conclusion And Future Work

The main result of this paper was a derivation of a novel algorithm for the regret table computation. The theoretical time complexity of this algorithm allows practical applications of NML in domains with very large datasets. With the earlier quadratic-time algorithms, this was not possible.

In the future, we plan to conduct an extensive set of empirical tests to see how well the theoretical advantage of the new algorithm transfers to practice. On the theoretical side, our goal is to extend the regret table computation to more complex cases like general graphical models. We will also research supervised versions of the stochastic complexity, designed for supervised prediction tasks such as classification.

Acknowledgements

This work was supported in part by the Academy of Finland under the projects Minos and Civi and by the National Technology Agency under the PMMA project. In addition, this work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

References

- [Barron *et al.*, 1998] A. Barron, J. Rissanen, and B. Yu. The minimum description principle in coding and modeling. *IEEE Transactions on Information Theory*, 44(6):2743–2760, October 1998.
- [Grünwald *et al.*, 1998] P. Grünwald, P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Minimum encoding approaches for predictive modeling. In G. Cooper and S. Moral, editors, *Proceedings of the 14th International Conference on Uncertainty in Artificial Intelligence (UAI'98)*, pages 183–192, Madison, WI, July 1998. Morgan Kaufmann Publishers, San Francisco, CA.
- [Grünwald, 1998] P. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty*. PhD thesis, CWI, ILLC Dissertation Series 1998-03, 1998.
- [Koivisto, 2004] M. Koivisto. *Sum-Product Algorithms for the Analysis of Genetic Risks*. PhD thesis, Report A-2004-1, Department of Computer Science, University of Helsinki, 2004.
- [Kontkanen and Myllymäki, 2005] P. Kontkanen and P. Myllymäki. Computing the regret table for multinomial data. Technical Report 2005-1, Helsinki Institute for Information Technology (HIIT), 2005.
- [Kontkanen *et al.*, 1996] P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report NC-TR-97-003, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1996.
- [Kontkanen *et al.*, 1998] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. On Bayesian case matching. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning, Proceedings of the 4th European Workshop (EWCBR-98)*, volume 1488 of *Lecture Notes in Artificial Intelligence*, pages 13–24. Springer-Verlag, 1998.
- [Kontkanen *et al.*, 2000a] P. Kontkanen, J. Lahtinen, P. Myllymäki, T. Silander, and H. Tirri. Supervised model-based visualization of high-dimensional data. *Intelligent Data Analysis*, 4:213–227, 2000.
- [Kontkanen *et al.*, 2000b] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. On predictive distributions and Bayesian networks. *Statistics and Computing*, 10:39–54, 2000.
- [Kontkanen *et al.*, 2003] P. Kontkanen, W. Buntine, P. Myllymäki, J. Rissanen, and H. Tirri. Efficient computation of stochastic complexity. In C. Bishop and B. Frey, editors, *Proceedings of the Ninth International Conference on Artificial Intelligence and Statistics*, pages 233–238. Society for Artificial Intelligence and Statistics, 2003.
- [Kontkanen *et al.*, 2005] P. Kontkanen, P. Myllymäki, W. Buntine, J. Rissanen, and H. Tirri. An MDL framework for data clustering. In P. Grünwald, I.J. Myung, and M. Pitt, editors, *Advances in Minimum Description Length: Theory and Applications*. The MIT Press, 2005.
- [Merhav and Feder, 1998] N. Merhav and M. Feder. Universal prediction. *IEEE Transactions on Information Theory*, 44(6):2124–2147, October 1998.
- [Rissanen, 1978] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:445–471, 1978.
- [Rissanen, 1987] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society*, 49(3):223–239 and 252–265, 1987.
- [Rissanen, 1996] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, January 1996.
- [Rissanen, 1999] J. Rissanen. Hypothesis selection and testing by the MDL principle. *Computer Journal*, 42(4):260–269, 1999.
- [Rissanen, 2001] J. Rissanen. Strong optimality of the normalized ML models as universal codes and information in data. *IEEE Transactions on Information Theory*, 47(5):1712–1717, July 2001.
- [Schwarz, 1978] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- [Shtarkov, 1987] Yu M. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23:3–17, 1987.
- [Xie and Barron, 2000] Q. Xie and A.R. Barron. Asymptotic minimax regret for data compression, gambling, and prediction. *IEEE Transactions on Information Theory*, 46(2):431–445, March 2000.

Using Neutral Examples for Learning Polarity

Moshe Koppel, Jonathan Schler

Bar-Ilan University
Computer Science Department
Ramat-Gan, 52900, ISRAEL
{ koppel ,schlerj }@cs.biu.ac.il

Abstract

Sentiment analysis is an example of polarity learning. Most research on learning to identify sentiment ignores “neutral” examples and instead performs training and testing using only examples of significant polarity. We show that it is crucial to use neutral examples in learning polarity for a variety of reasons and show how neutral examples help us obtain superior classification results in two sentiment analysis test-beds.

Many machine-learning problems involve predicting an example’s *polarity*: is it (significantly) greater than or less than some standard. One canonical example of learning polarity is *sentiment analysis*, the determination of whether a particular text expresses positive or negative sentiment regarding some issue.

The problem of how to exploit a labeled corpus to learn models for sentiment analysis has attracted a good deal of interest in recent years [Dave et al 2003, Pang et al 2002, Shanahan et al 2005]. One common characteristic of almost all this work has been the tendency to define the task as a two-category problem: positive versus negative. In almost all actual polarity problems, including sentiment analysis, there are, however, *three* categories that must be distinguished: positive, negative and *neutral*. Not every comment on a product or experience expresses purely positive or negative sentiment. Some – in many cases, most – comments might report objective facts without expressing any sentiment, while others might express mixed or conflicting sentiment.

Researchers are aware, of course, of the existence of neutral documents. The rationale for ignoring them has been a reliance on two tacit assumptions:

- Solving the binary positive vs. negative problem automatically solves the three-category problem since neutral documents will simply lie near the boundary of the binary model
- There is less to learn from neutral documents than from documents with clearly defined sentiment

The purpose of this paper is to show that there is no basis for either of those myths and that neutrals can be exploited in interesting ways to great effect.

We consider two labeled corpora. The first consists of 1974 posts to chat groups devoted to popular U.S. television shows. The second consists of about 14,000 posts to shopping.com’s product evaluation pages. Both are equally distributed among positive, negative and neutral documents.

Is it in fact the case that neutral documents lie near the boundary of a learned model that distinguishes positive and negative examples? To test this, we trained a linear SVM on all positive and negative documents in the TV corpus. In Figure 1, we show the signed distance from the boundary of the positive and negative training examples, in ascending order from left to right. (This SVM correctly classifies 79.1% of the training examples.) In addition, we show the signed distance from the boundary of all neutral examples. There is no band near the boundary in which the preponderance of examples is neutral. We indicate the band around the boundary that is optimal in terms of overall classification accuracy (positive, negative, or neutral) when all examples in the band are classed as neutral. Even using this optimal band, we attain accuracy of only 54.8%. (Note that simply using the SVM boundary to distinguish positive from negative and not classifying any examples as neutral would yield accuracy of 52.7%.)

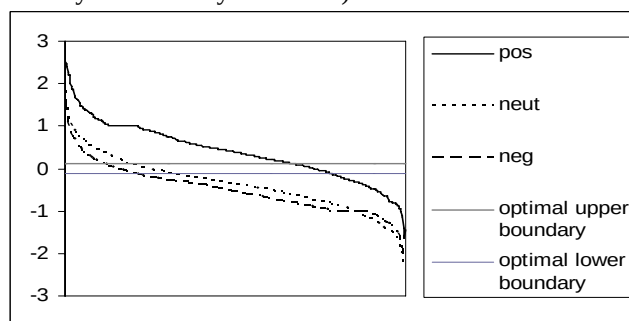


Figure 1 Distance from boundary in the TV shows corpus

Similar results are obtained for the second dataset.

What happens when we try to solve the three-class problem using positive, negative and neutral training documents? Five-fold cross-validation experiments using Weka’s implementation of multi-class SVM yields accuracy

of 56.5% for the TV corpus and 63.8% for the shopping.com corpus. Interestingly, we will see that these results are far inferior to results obtainable by making even stronger use of neutral documents.

Consider the algorithm used in this experiment for extending a binary algorithm to handle multiple classes, namely, pairwise coupling. In this approach, a model is learned for each pair of classes (positive-negative, positive-neutral, negative-neutral) and these models are then combined. Weka's implementation [Witten and Frank 2000] of the [Hastie and Tibshirani 1998] algorithm treats the three constituent pairwise problems identically. That is, no allowance is made for the particular relationships that positive, negative and neutral examples stand in to each other.

The main point of this paper is that it is crucial to take these special relationships into account. We begin by running the following experiment. For each of the pairs, negative-positive, negative-neutral, and positive-neutral, we ran five-fold cross-validation experiments. For each example, we recorded how it was classed in the holdout set in each of the three experiments.

Table 1 shows the actual class distribution of examples in the TV corpus assigned to each of the eight possible outcomes.

Pos Vs Neg	Pos Vs Neut	Neut Vs neg	original category		
			neg	neut	pos
Neg	Neut	Neg	354	52	
Neg	Neut	Neut	117	154	148
Neg	Pos	Neg		47	
Neg	Pos	Neut		9	108
Pos	Neut	Neg	145	69	
Pos	Neut	Neut	42	225	46
Pos	Pos	Neg		90	
Pos	Pos	Neut		12	356

Table 1: Class distribution of examples per pairwise outcomes in TV corpus

As can easily be computed from the table, the accuracies of the pairwise models in five-fold cross-validation trials on their respective category pairs are: positive-negative, 67.3%; positive-neutral, 73.7%; negative-neutral, 68.5%. We want to parlay these pairwise models into the best possible three-class model. To do this, let us define a *stack* [Wolpert 1992] as a mapping from each of the eight possible outcomes to some class. Let an *optimal stack* be the mapping from each of the eight possible outcomes to the majority class of the examples with that outcome. [Savicky and Furnkranz 2002] have considered when such optimal stacks (determined using holdout data) might permit optimal use of pairwise coupling.

For a given example, let's use the shorthand Class1 > Class2 to mean that the learned model of Class1 vs. Class2

classed the example as Class1. The optimal stack for this data can be neatly summarized as follows:

- If positive > neutral > negative then class=positive
- If negative > neutral > positive then class=negative
- Else class=neutral

This simple stack yields accuracy of 74.9% on the three-class problem, which is significantly better than multi-class SVM (and better than any of the constituent two-class problems).

What is most astonishing about this table is the following: When, according to our model for positive vs. neutral, a test example is classified as positive, it is not necessarily positive, but we *can assert with certainty that it is not negative* (despite not a single negative example being used in training.) Likewise, when, according to our model for negative vs. neutral, a test example is classified as negative, it is not necessarily negative, but we *can assert with certainty that it is not positive* (despite not a single positive example being used in training.)

An analogous (though not identical) principle holds in the shopping.com corpus.

These results strongly suggest that polarity problems be attacked by stacking results of pairwise coupling in non-standard ways, taking full advantage of neutral examples.

References

- [Dave, K., Lawrence, S., and Pennock, D. M., 2003] Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the Twelfth International World Wide Web Conference (WWW-2003)*
- [Hastie, T. and R. Tibshirani, 1998] Classification by pairwise coupling in M. I. Jordan, M. J. Kearns, and S. A. Solla (eds.) *Advances in Neural Information Processing Systems 10 (NIPS-97)*, pp. 507-513. MIT Press, 1998.
- [Pang, B., Lee, L. and Vaithyanathan, S., 2002] Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*
- [Savicky, P. And Fuernkranz, J., 2003] Combining Pairwise Classifiers with Stacking. in *Advances in Intelligent Data Analysis V.* (Ed.: Berthold M.R., Lenz H.J., Bradley E., Kruse R., Borgelt Ch.) - Berlin, Springer 2003, pp. 219-229.
- [Shanahan, J. G., Yan Q., Janyce W. (Eds.), 2005] *Computing Attitude and Affect in Text*, Springer, Dordrecht, The Netherlands, 2005
- [Witten I. H. and Frank E. 2000] *"Data Mining: Practical machine learning tools with Java implementations"* Morgan Kaufmann, San Francisco, 2000
- [Wolpert, D.H., 1992], Stacked Generalization, *Neural Networks*, Vol. 5, pp. 241-259, Pergamon Press

Transfer in Learning by Doing

Bill Krueger, Tim Oates, Tom Armstrong
University of Maryland Baltimore County
CSEE, 1000 Hilltop Circle
Baltimore, MD 21250
{wkrueg1,oates,arm1}@cs.umbc.edu

Paul Cohen, Carole Beal
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
{cohen,cbeal}@isi.edu

Abstract

We develop two related themes, *learning procedures* and *knowledge transfer*. This paper introduces two methods for learning procedures and one for transferring previously-learned knowledge to a slightly different task. We demonstrate by experiment that transfer accelerates learning.

1 Introduction

Procedures are interesting for many reasons: Infants exercise procedures (called “circular reactions” by Piaget) almost from birth. These gradually develop in complexity, adding new elements, and eventually incorporate objects in the infant’s environment. Piaget believed infants learn much about the world by executing procedures. Older students do, as well; indeed, the literature on expert/novice differences can be summarized in a single phrase: Novices know “how,” experts also know “why.” Novices can run statistical tests but not understand why they are appropriate, they can follow recipes but not understand the underlying chemistry and gastronomy.

There are advantages to learning fact-like knowledge alongside procedures. One can do useful work with procedures even if one doesn’t completely understand them. This means learning is grounded in activity and is gradual over the life of the agent. By exercising procedures the learner produces occasional failures as well as the context one needs to gradually learn both new procedures and non-procedural knowledge. The latter includes the facts and reasons we call “understanding,” or, less colloquially, the conditioning variables that affect the probabilities that procedures will succeed. In social environments, there often is a human to help the learner correct missteps before a procedure goes completely wrong. This minimizes the credit-assignment problem.

Our intention is to have a machine learn a sequence of increasingly-difficult games, starting with extremely easy ones. All the games are two-person card games. One player makes mistakes, the other is a competent player who offers corrections as necessary. For convenience we call these players the learner or child (*C*) and the adult (*A*). Correct play for the games can be modeled by finite state machines, so the

learner learns finite state machines and uses its current machine to generate its next move.

The experiments in this paper involve a sequence of three games. In the first game, the child and the adult each have two stacks of cards, one in which all of the cards are face-up and another in which all of the cards are face-down. Let’s denote these stacks by the owner (*A* for adult or *C* for child) and whether they are face-up (*U*) or face-down (*D*). Therefore, the four stacks are *AU*, *AD*, *CU*, *CD*. Each player must flip over the card on top of their face-down stack and move it to the top of their face-up stack. Each player does this as fast as they can, and can ignore the other player’s actions for the purposes of this game.

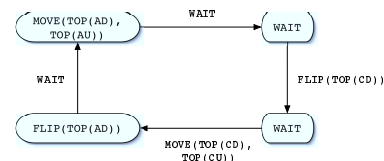


Figure 1: A state machine for Game 2 that yields perfect play.

In the second game, the players take turns flipping and moving cards, as shown in Figure 1. The two nodes on the left specify constraints on what the child observes, namely, the action that is currently being taken by the adult. To be in the lower-left state in the machine, the adult must currently be flipping over the top card on her face-down stack (*FLIP(TOP(AD))*). When this happens, the action executed by the child is *WAIT*. That is, when the adult does *FLIP(TOP(AD))*, the child does *WAIT* at the same time. This leads to a state in which the adult does *MOVE(TOP(AD), TOP(AU))*, which also results in the child waiting. Then, when the adult executes a *WAIT* action (the constraint in the upper-right state), the child flips the card on top of her face-down stack, and the adult continues to wait (the lower-right state) as the child does *MOVE(TOP(CD), TOP(CU))*. Some time later, the adult flips the card on top of her face-down stack, and the cycle repeats.

The third game is just like the second game, except when either player turns over a card and the color of that card matches that of the card atop the other player’s face-up stack, both players say “Squawk”.

2 Learning the State Machines

Our goal is to learn procedures, and, specifically, perfect-play state machines such as those we described in the previous section. A useful distinction can be drawn between learning a state machine and learning a state machine given a related state machine. We are most interested in the second case, which we present as a kind of *transfer learning*. Even so, we begin our discussion of learning with two methods for learning state machines *de novo*. These are Bayesian Model Merging (BMM) [Stolcke, 1994], a well-known method for learning HMMs; and State Splitting (SS), which also learns HMMs, though by splitting rather than merging states.

The training data in both approaches was a sequence of observations such as `MOVE(TOP(CD), TOP(CU))`, the symbol `SQUAWK`, and the special symbol `NO`. The BMM and SS algorithms rely on `NO` to indicate that the child has done something that is not allowed in perfect play. Incorrect play is generated by the learner’s incorrect machine in the active learning regime. Passive learning requires training data that includes mistakes and adult admonitions (i.e., `NO`) and corrections. For this purpose we build simulators of child and adult play.

BMM is a top-down approach in which an initial HMM is constructed with one state for every observation, and observations are greedily merged to maximize the posterior likelihood of the data using a description length prior on HMMs. It is incremental in the sense that new observations can be added during the merging process. SS is a bottom-up approach that starts with one state that matches all observations, and repeatedly splits states on observable features of the world in an effort to more accurately predict negative feedback from the teacher. Features are chosen that minimize entropy in the distribution of feedback (positive or negative) over all states.

Regardless of whether the HMM was learned via BMM or SS, it can be used for action selection (game play) and to parse new observations. We are interested in whether learning a series of related, yet increasingly difficult, games requires less effort in total than learning the most difficult game *de novo*. For example, is it easier to learn our three games in sequence, using the machine learned for game n to bias the learning of game $n + 1$, than to learn to play Squawk with no prior machine to serve as bias?

When using BMM to learn a new game biased by an existing HMM, new states and transitions are added for the new observations, but old and obsolete state transitions are not removed. Instead, as more and more data are collected for the new game, the probabilities of obsolete state transitions that are no longer traversed will tend to zero. This is not acceptable, as we want to quickly refine the model to work with the new game while harnessing as much information as possible from previous experience. To achieve this we preserve model structure but eliminate the bias of old probability parameters when moving from one game to the next. In BMM terminology, this means that we reset Viterbi counts to be uniform and small when starting to learn a new game.

3 Experiments

Figure 2 shows two machines that were intermediate steps on the path to learning the turn taking game. In the top machine, there is a single state from which the child can `WAIT`, `MOVE(CD CU)`, or `FLIP(CD)`. Edges are labeled with two counts - the number of times the transition was taken and the number of times that transition resulted in a `NO`. This machine does not yield perfect play, so the state will be split. The observable feature that most reduces entropy in the distribution of negative feedback is the feedback received for the move just made.

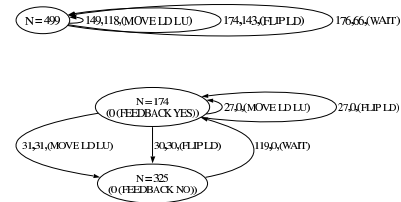


Figure 2: Machines produced by SS when learning the turn taking game.

The bottom machine in Figure 2 shows the result of splitting on this feature and pruning away any transitions that always cause a `NO`. This machine correctly captures the fact that if the adult corrects the child, the only legal action is to `WAIT` while the adult repairs the incorrect action, which will result in a `YES`. Residual feedback non-determinism exists in the top state due to errors in action ordering (e.g., flipping twice in a row). Therefore, that state will be split next.

In a second set of experiments, the child chose actions randomly from a set of known actions until a model was developed (using BMM) that could guide action selection. Initially, therefore, the child would frequently choose an incorrect action. Learning early on with no model is tedious and time-consuming as it requires heavy exploration. The second game, the simple turn-taking game, was still easy to learn. Only 10 simulation time steps were required to learn the optimal model for that game when starting from scratch. Without using this model to bias the model learned for the third game, 53 time steps were required to achieve the optimal model for the more complex game. However, when the game 2 model was used to bias the learning of game 3, only an additional 17 time steps were needed to learn game 3, making a total of 27 time steps. That is, game 3 took in total half as long to learn when game 2 was learned as an intermediate step in the learning process.

Acknowledgments

This work was supported by DARPA (contract number: 53-4540-0016; project name: Learning by Doing).

References

- [Stolcke, 1994] Andreas Stolcke. *Bayesian Learning of Probabilistic Language Models*. PhD thesis, University of California, Berkeley, 1994.

Supervised Local Tangent Space Alignment for Classification

Hongyu Li¹, Wenbin Chen², I-Fan Shen¹

¹Department of Computer Science and Engineering ²Department of Mathematics
Fudan University, Shanghai, China
{hongyuli,wbchen,yfshen}@fudan.edu.cn

Abstract

Supervised local tangent space alignment (SLTSA) is an extension of local tangent space alignment (LTSA) to supervised feature extraction. Two algorithmic improvements are made upon LTSA for classification. First a simple technique is proposed to map new data to the embedded low-dimensional space and make LTSA suitable in a changing, dynamic environment. Then SLTSA is introduced to deal with data sets containing multiple classes with class membership information.

1 Introduction

In many pattern recognition problems, original data taken with various capturing devices are usually of high dimensionality. Dimension reduction has emerged with an aim to obtain compact representations of the original data while reducing unimportant factors. It can make the resulting classification more efficient as fewer variables are being considered which reduces the complexity of the resulting classifier. LTSA proposed in [Zhang and Zha, 2004] is an unsupervised method for nonlinear dimension reduction. It does not make use of the class membership of each point to be projected and lacks generalization to new data. Here we extend LTSA to a supervised version and make it suitable in a changing environment.

2 Local Tangent Space Alignment

LTSA maps a data set $X = [x_1, \dots, x_N]$, $x_i \in R^m$ globally to a data set $Y = [y_1, \dots, y_N]$, $y_i \in R^d$ with $d < m$. The brief description of LTSA is presented as follows:

1. Finding k nearest neighbors x_{ij} of x_i , $j = 1, \dots, k$. Set $X_i = [x_{ij}, \dots, x_{ik}]$.
2. Extracting local information by calculating the d largest eigenvectors $g_1, \dots, g_d$ of the correlation matrix $(X_i - \bar{x}_i e^T)^T (X_i - \bar{x}_i e^T)$, and setting $G_i = [e/\sqrt{k}, g_1, \dots, g_d]$. Here $\bar{x}_i = \frac{1}{k} \sum_j x_{ij}$, e is a k -dimensional column vector of all ones.
3. Constructing the alignment matrix B (please refer to [Zhang and Zha, 2004] for details) by locally summing with initial $B = 0$ as follows:

$$B(I_i, I_i) \leftarrow B(I_i, I_i) + I - G_i G_i^T, i = 1, \dots, N.$$

The neighborhood set I_i represents the set of indices for the k nearest-neighbors of x_i . I is an $N \times N$ identity matrix.

4. Computing the $d + 1$ smallest eigenvectors of B and setting the global coordinates $Y = [u_2, \dots, u_{d+1}]^T$ corresponding to the 2nd to $d + 1$ st smallest eigenvalues.

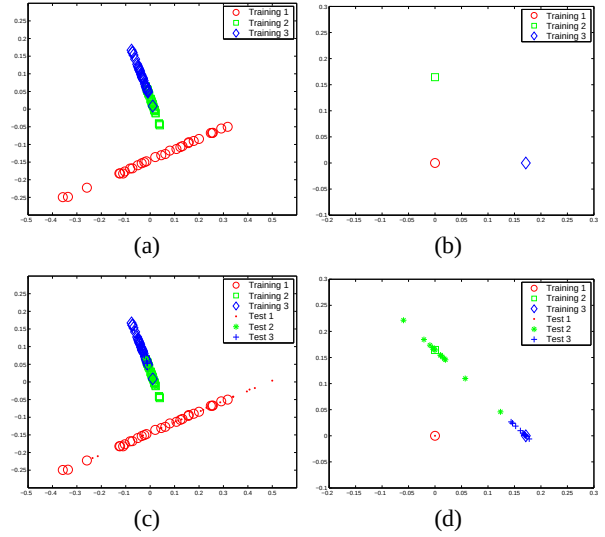


Figure 1: (a) and (b): Mapping training samples with LTSA and SLTSA respectively. (c) and (d): Mapping unknown test samples to the spaces discovered by LTSA and SLTSA.

3 Supervised LTSA

Our attempt is to adapt LTSA to a situation when the data come incrementally point by point. We assume that the dimensionality of the embedded space does not grow after projecting a new point to it, i.e., d remains constant.

The proposed technique is based on the fact that new points are assumed to come from those parts of the high-dimensional space that have already been explicitly mapped by LTSA. It implies that when a new point arrives, a task of interpolation should be solved. Let X as an input. For a new point x_{N+1} , its closest neighbor x_j is looked for from X . If y_j is the

projection of x_j to the embedded space, there must exist a point y_{N+1} around y_j corresponding to x_{N+1} .

LTSA belongs to unsupervised methods; it does not make use of the class membership of each point. *Supervised* LTSA (SLTSA) is proposed for classification purposes. The term implies that membership information is employed to form the neighborhood of each point. That is, nearest neighbors of a given point x_i are chosen only from representatives of the same class as that of x_i . This can be achieved by artificially increasing the shift distances between samples belonging to different classes, but leaving them unchanged if samples are from the same class.

In short, SLTSA is designed specially for dealing with data sets containing multiple classes. The results obtained with the unsupervised and supervised LTSA are expected to be different as is shown in Fig.1. The **iris** data set [Blake and Merz, 1998] includes 150 4-D data belonging to 3 different classes. Here first 100 data points are selected as training samples and mapped from the 4-D input space to a 2-D feature space using LTSA (Fig.1(b)) and SLTSA (Fig.1(a)) respectively. In Fig.1(b), three class centers on the feature space are completely separated. However in Fig.1(a) two classes (set 2 and 3) overlap such that their boundary cannot be accurately determined. Fig.1(c; d) respectively show the mapping of unknown test samples to the two feature spaces discovered by LTSA and SLTSA. In Fig.1(d), test samples in different classes can be well distributed around class centers. Thus the classification of test samples can be easily implemented on such space.

4 Results

To examine its performance, SLTSA has been applied to a number of data sets varying in number of samples, dimensions and classes.

The **binarydigits** set discussed here consists of 20×16 -pixel binary images of preprocessed handwritten digits. In our experiment only three digits: 0, 1 and 2 are dealt with and some of the data are shown in Fig.2(a). 90 of the 117 binary images are used as training samples and others as test samples. It is clear that the 320-dimensional binarydigits data contain many insignificant features, so removing them can make classification more efficient. The results after dimension reduction from 320 to 2 with LTSA and SLTSA are respectively displayed in Fig.2(b) and 2(c) where two coordinate axes represent the two most important features of the binarydigits data. The figures show that the feature space obtained with SLTSA provides better classification information than LTSA. Actually no cases of misclassification in the test samples occurred if using the SLTSA method, but LTSA will result in the error rate of 18.52% in the best case.

Our experimental results confirm that SLTSA generally leads to better classification performance than LTSA when combined with simple classifiers such as *k-nn* and *lda*. Beside this, the performance of SLTSA is comparable with supervised LLE [Kouropteva *et al.*, 2002] and finer than with PCA (please refer to the supplementary document). Maybe the reason is that those points on a high-dimensional input space are more likely to lie close on nonlinear rather than

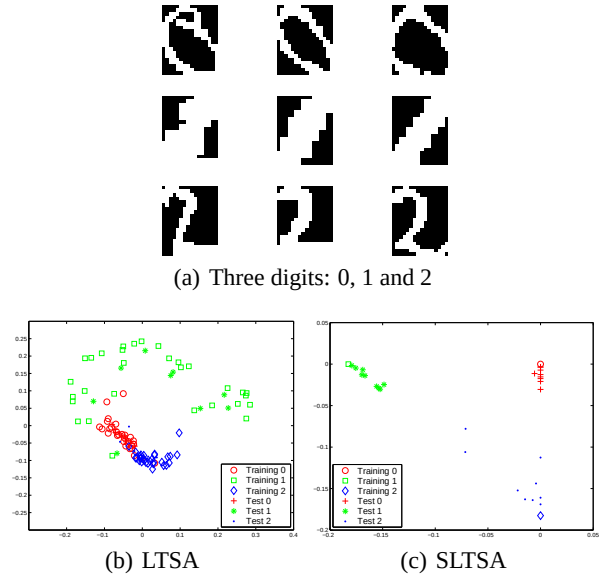


Figure 2: Recognition of three digits: 0, 1 and 2. (a) Some of these three digits are shown. They are represented by 20×16 binary images which can be considered as points on a 320-dimensional space. (b) Mapping these binary digits including training and test samples to a 2-dimensional feature space with LTSA, where these three clusters overlap and are unseparated. (c) The feature space obtained with SLTSA, which provides better clustering information.

linear manifolds. Thus, such linear methods as PCA cannot perform well.

5 Conclusions

The supervised version of LTSA was proposed here. It takes into account class membership during selecting neighbors. Another enhancement to LTSA is to generalize it to new data and make it suitable in a changing, dynamic environment.

We compare SLTSA with LTSA, PCA and supervised LLE on a number of data sets, in order to gain insight into what methods are suitable for data classification. The SLTSA method has been shown to yield very promising classification results in our experiments when combined with some simple classifiers.

References

- [Blake and Merz, 1998] C.L. Blake and C.J. Merz. Uci repository of machine learning databases, 1998.
- [Kouropteva *et al.*, 2002] O. Kouropteva, O. Okun, A. Hadid, M. Soriano, S. Marcos, and M. Pietikainen. Beyond locally linear embedding algorithm. Technical Report MVG-01-2002, Machine Vision Group, University of Oulu, 2002.
- [Zhang and Zha, 2004] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26(1):313–338, 2004.

Active Cost-Sensitive Learning

Dragos D. Margineantu

The Boeing Company

Mathematics & Computing Technology, Adaptive Systems

P.O.Box 3707, M/S 7L-66

Seattle, WA 98124-2207, USA

dragos.d.margineantu@boeing.com

Abstract

For many classification tasks a large number of instances available for training are unlabeled and the cost associated with the labeling process varies over the input space. Meanwhile, virtually all these problems require classifiers that minimize a non-uniform loss function associated with the classification decisions (rather than the accuracy or number of errors). For example, to train pattern classification models for a network intrusion detection task, experts need to analyze network events and assign them labels. This can be a very costly procedure if the instances to be labeled are selected at random. In the meantime, the loss associated with mislabeling an intrusion is much higher than the loss associated with the opposite error (i.e., labeling a legal event as being an intrusion).

As a result, to address these types of tasks, practitioners need tools that minimize the total cost computed as a sum of the cost of labeling and the loss associated with the decisions.

This paper describes an approach for addressing this problem.

1 Introduction

A number of applications require learning algorithms that are capable (1) to learn from both labeled and unlabeled examples, (2) to minimize the cost non-uniform associated with the labeling efforts, and (3) to minimize the misclassification loss.

The active learning framework [Cohn *et al.*, 1994] relies on algorithms that select unlabeled instances and provide them to the expert for labeling and learn a classifier based on the labeled data. In most of the cases the data is labeled incrementally and the classification model tries to minimize the misclassification error rate. To our knowledge, virtually all research efforts in active learning have assumed both a uniform loss function (for which the goal was to reduce the number of misclassifications) and a uniform labeling cost function (for which the goal was to reduce the number of instances that have to be labeled to achieve a certain accuracy).

Meanwhile, research in cost-sensitive learning [Elkan, 2001] has focused mostly on the problem of minimizing the

expected loss of the learned models, and has assumed that all (training) instances and their labels were readily available prior to the training phase.

Our approach in addressing the active cost-sensitive learning problem is based on learning algorithms that construct models for conditional density (or class probability) estimation $P(y|x)$, rather than class label predictors. The probability estimates provide an easy means for factoring in the misclassification losses in the classification decision making step. To address the labeling cost problem, we employ the same density estimation techniques over the available unlabeled data. The two methods are then combined into an algorithm that minimizes the combined cost.

Next section presents the general framework for our proposed techniques and discusses the challenges in implementing and applying these algorithms as well as some preliminary experimental results.

2 Active Learning for Minimizing the Combined Costs for Labeling and Decisions

Most learning algorithms can be transformed into class probability estimators that compute estimates $P(y|x)$ for the probabilities that an instance x is in class y ($y \in \{1 \dots K\}$, where K is the number of classes). We employ these estimates in our algorithm both for minimizing the labeling costs and the misclassification decisions. We make the assumption that the loss function associated with the decisions is represented as a static K -by- K loss matrix L available at learning time. The contents of $L(i, j)$ specify the cost incurred when an example is predicted to be in class i when in fact it belongs to class j .

The pseudo code for our approach, ACTIVE-CSL, is presented in Table 1.

The algorithm trains a base learner to compute the class probability estimates over the unlabeled data (line 2). Then, both the sampling step (line 3) and the decision making step (the hypothesis h) are based on those estimates. The unlabeled instance for which the next label is requested from the expert is selected in line 3. The selection rule chooses the instance that, if labeled provides the most expected gain in terms of the total cost (labeling + decision loss) on the labeled instances (or on a hold-out labeled validation set). It is important to note that iff C and L functions have a shape such that instances that are more expensive to label reduce the

Table 1: Pseudo code for the ACTIVE-CSL active cost-sensitive learning algorithm.

Input: a set S_l , of m labeled examples:
 $S_l = \langle (x_i, y_i), i = 1, 2, \dots, m \rangle$,
a set S_u of v unlabeled examples:
 $S_u = \langle (\xi_i), i = 1, 2, \dots, v \rangle$,
 L (a loss matrix), C (a labeling cost function)
a stopping criterion

```

[1] repeat
[2]   for each  $j$  learn  $P(y|\xi_j)$  using  $S_l$  as training data;
[3]    $l = \operatorname{argmin}_{\xi \in S_u} (C(\xi) +$ 
      
$$+ \sum_{k=1}^K P(k|\xi) \sum_i \sum_{j=1}^K P_{(\xi,k)}(j|x_i) L(y_i, j));$$

[4]    $\psi_l = \text{requested label for } \xi_l$ ;
[5]   remove  $\xi_l$  from  $S_u$ , add  $(\xi_l, \psi_l)$  to  $S_l$ ;
[6] while stopping criterion not met

Output:  $h(\mathbf{x}) = \operatorname{argmin}_{y \in Y} \sum_{j=1}^K P(j|\mathbf{x}) L(y, j)$ 
// the optimal classification with respect to  $L$  and  $P$ 

```

misclassification cost more than instances that are less expensive to label - the selection rule will trade the loss associated with misclassifications against examples to be labeled. The procedure selecting for the next instance to be labeled, as employed by ACTIVE-CSL, and based on computing the maximum expected gain upon labeling (the second sum in line 3), is somewhat similar in nature to the querying rule of uncertainty sampling [Lewis and Catlett, 1994].

One important detail that needs to be addressed when applying this algorithm in practice is the choice for the algorithm used for estimating the probabilities (line 3 in Table 1), given that ACTIVE-CSL and its predictions rely on these estimates. In general, the quality of learned density estimates is dependent on the amount and the distribution of the labeled data that is available and on the hypothesis constructed by the base learning algorithm. Several research studies have addressed the problem of learning good probability estimates and calibrating classification scores and ranks into accurate probabilities [Zadrozny and Elkan, 2001].

We have implemented and tested ACTIVE-CSL by using bagged probability estimation trees [Provost and Domingos, 2003] as class probability estimators (in line 2 of the code). We have compared two procedures for computing probability estimates out of the bagged trees: (a) by averaging the probabilities computed by the individual trees and (b) by estimating the confidence for the individual probabilities based on the distribution of the estimates of the individual trees (as suggested by [Margineantu, 2002]). For assessing the quality of the misclassification decisions we have employed active learning curves (showing the gain in terms of cost compared to a random selection of instances for labeling) and BDELTA-COST [Margineantu and Dietterich, 2000] - a paired test for cost-sensitive classification decisions. The loss matrices for

our experiments were generated based on some generic loss models, and the labeling cost function mapped instances that were closer to the decision boundary to higher labeling costs (the actual function we employed was a bell-shaped function with a maximum on the decision boundary).

We tested the two implementations of ACTIVE-CSL on five data sets from the UC Irvine Repository [Blake and Merz, 1998] (Breast cancer Wisconsin, Horse colic, King-rook vs. king-pawn, Liver disease, and Sonar) and on the binary version of the KDD Cup 1998 donations data.

The preliminary results show a minor advantage of the implementation employing confidence-based estimates over the implementation using averaged probability estimates. We have also run tests by employing random forests with $\log m$ attributes tested in a node [Breiman, 2001] as the class probability estimator and the results show no significant difference between the classification decisions based on random forests and the decisions based on bagged probability estimation trees.

References

- [Blake and Merz, 1998] C. L. Blake and C. J. Merz. UCI Repository of ML databases, 1998. [www.ics.uci.edu/~mllearn/MLRepository.html].
- [Breiman, 2001] L. Breiman. Random forests. Technical report, Dept. of Statistics, University of California, Berkeley, 2001.
- [Cohn et al., 1994] D. A. Cohn, L. Atlas, and R. Ladner. Improved generalization with active learning. *Machine Learning*, 15:201–221, 1994.
- [Elkan, 2001] C. Elkan. The foundations of cost-sensitive learning. In *Proc. of the Seventeenth Intrnl. Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 2001.
- [Greiner et al., 2002] R. Greiner, A. J. Grove, and D. Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139:2, pages 137–174, 2002.
- [Hettich and Bay, 1999] S. Hettich and S. D. Bay. The UCI KDD archive, 1999. [http://kdd.ics.uci.edu/].
- [Lewis and Catlett, 1994] D. Lewis and J. Catlett. Heterogeneous uncertainty sampling. In *Proc. Eleventh Intrnl. Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.
- [Margineantu and Dietterich, 2000] D. D. Margineantu and T. G. Dietterich. Bootstrap methods for the cost-sensitive evaluation of classifiers. In *Proc. of the Seventeenth Intrnl. Conference on Machine Learning*, pages 583–590. Morgan Kaufmann, 2000.
- [Margineantu, 2002] D. D. Margineantu. Class probability estimation and cost-sensitive classification decisions. In *Proc. ECML-2002, 13th European Conference on Machine Learning, Proceedings*, pages 270–281. LNAI 2430, Springer Verlag, 2002.
- [Provost and Domingos, 2003] F. Provost and P. Domingos. Tree induction for probability-based rankings. *Machine Learning*, 52:3, 2003.
- [Saar-Tschansky and Provost, 2001] M. Saar-Tschansky and F. Provost. Active learning for class probability estimation and ranking. In *Proc. Seventeenth Intrnl. Joint Conf. on Artificial Intelligence*, pages 911–917. AAAI Press/MIT Press, 2001.
- [Zadrozny and Elkan, 2001] B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proc. of the Eighteenth Intrnl. Conference on Machine Learning*, pages 609–616. Morgan Kaufmann, 2001.

Automatic learning of domain model for personalized hypermedia applications

Hermine Njike, Thierry Artières, Patrick Gallinari, Julien Blanchard, Guillaume Letellier

LIP6, Université Paris 6
8 rue du capitaine Scott, 75015, Paris, France
{Firstname.Lastname}@lip6.fr

Abstract

This paper deals with the automatic building of personalized hypermedia. We build upon ideas developed for educational hypermedia; the definition of a domain model and the use of overlay user models. Since much work has been done on learning user models and adapting hypermedia based on such models, we tackle the core problem: the automatic definition of a domain model for a static hypermedia.

1 Introduction

In adaptive hypermedia one aims at developing user centric help strategies. One has to characterize the user in order to infer relevant help action for him all along his situation. A domain model may be used that characterizes the whole knowledge accessible in the hypermedia; it is used to infer information in the user model [De Bra et al, 2003]. Based on a user model, an adaptation model personalizes the hypermedia in order to offer to the user the most interesting and relevant information [Brusilovsky 2001]. An interesting task in adaptive hypermedia concerns educational hypermedia [Henze et al, 1999]. Although building such systems is still difficult, the task is well identified; in such systems domain models are often manually designed and defined as a graph of the concepts being discussed in the hypermedia [Herder et al, 2002]. Popular user models are *overlay* user models; these share the same representation as domain models and are used to represent a user knowledge / interest in a concept space. These models are vectors of attributes (measures of interest or knowledge), one for each concept in the domain model. These are updated from user navigation logs according to the domain model with standard techniques such as Bayesian Networks. However, the definition of domain models, hence user models, is done manually. We are interested in learning automatically such models, allowing the automatic building of personalized hypermedia from its content. This would allow for instance the automatic building of an adaptive website from any document collection. To do this, one can take advantage of works done in the educational hypermedia field concerning the learning of overlay user models and the personalization of hypermedia based on overlay user models. In this context, the core problem for the automatic conception of a

personalized hypermedia lies in the automatic learning of a relevant domain model. We focus on this task here. This is not an easy task since this model involves high level concepts that cannot be easily inferred automatically.

2 Discovering concepts from a corpus

Some approaches were developed in the information retrieval field for generating documents hierarchies according to specialization/ generalization relations between concepts but these fail to infer semantic relations between concepts and are thus useless regarding our goal. Recently hierarchies have been proposed [Krishna et al, 2001, Sanderson et al, 1999] which are automatically built from corpora. They are term hierarchies built from generalization / specialization relations automatically discovered between terms in a corpus. Once this hierarchy is built, documents may be “projected” on it, thus producing a document hierarchy. We propose to extend these approaches to the construction of concept hierarchies where concepts are discovered from the corpus and are identified by sets of keywords and not only by single terms. Such a richer representation allows discovering relations that single term concept cannot. Our method takes as input a corpus, i.e. a collection of documents (e.g. pages of a website) that are preprocessed as usual in information retrieval tasks; non informative words are removed, all remaining words are lemmatized. The main steps of the procedure are as follows. Documents are first segmented into homogeneous paragraphs. The segmentation task consists in identifying, in each document, homogeneous text regions [Hearst, 1997]. Next, we cluster all these parts of documents in order to determine groups of paragraphs related to a similar topic. Each cluster is then considered to be a set of part of texts dealing with a concept of the collection. Each concept is represented as the set of most significant words (e.g. with highest *tfidf* values) in the texts of the cluster. Finally, we discover generalization links between concepts using a subsumption measure that we have extended to concepts. At last, applying transitivity leads to a concept hierarchy.

3 Application to user modeling

We applied our method to the discovery of a domain model, i.e. a concept hierarchy, on a collection of documents which is a part of the www.looksmart.com site hierarchies. The

main interest of this corpus is that we can compare, after learning, the discovered hierarchy and the manually designed one. The corpus consists in about 100 documents and 7k terms about artificial intelligence. It has been manually organized in hierarchies of themes. We ran the method on the flat corpus, without any use of the hierarchical information. Compared to the initial Looksmart hierarchy with five categories, our discovered hierarchy is larger and deeper. Most of the original categories are refined. For example, many sub-categories do emerge from the original “Knowledge Representation” category: ontologies, building ontologies, KDD (papers about data representation for KDD)... It is clear that such a hierarchy could not have been obtained using single keyword concepts. An interesting feature of this discovered hierarchical organization is that it allows using efficient visualization tools such as Treemaps [Schneiderman, 1992]. Treemaps allow displaying a tree-like structure in a 2D grid where each node is represented by a rectangle whose size or color determined by a node specific value. Fig. 1 shows a Treemap representing the discovered Looksmart domain model. Each concept is shown as a rectangle with different colour; the hierarchy is shown through inclusion of rectangles. Smaller rectangles represent documents. Set of keywords associated to concepts only are shown for readability reasons.

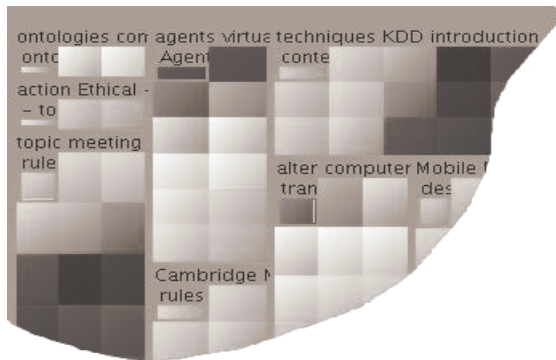


Fig. 1. Part of the treemap of the concept hierarchy discovered on the Looksmart corpus. Each rectangle stands for a concept.

We realized a second experiment by running the method on the pages of the website of a French museum. Fig. 2 shows the resulting domain model with french keywords. As an illustration, we have shown a navigation path of a particular user on this treemap, where the colour of a concept is a function of its thematic similarity with the three last pages visited by the user. As may be seen, concepts close to these pages lie close to the current concept. Other information could be visualized. Hence, one can browse and investigate a user model by assigning for instance *knowledge* or *interest* information about concepts to the size or colour of rectangles.

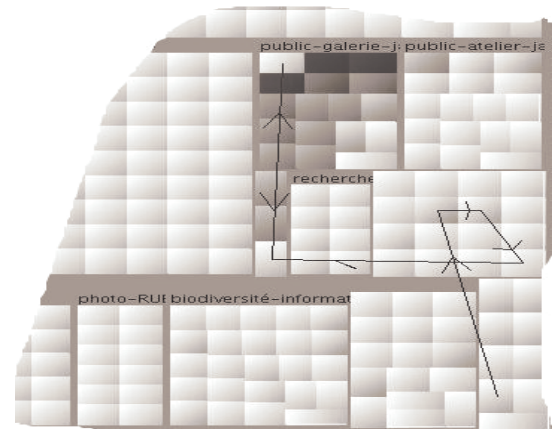


Fig. 2. Part of the treemap for the website of a French museum where a navigation path has been drawn.

4 Conclusion

We described an approach to automatically learn a domain model from a corpus of hypermedia documents. This model consists of concepts organized into a specialization / generalization Based on such a domain model, one can define user models as overlay models, and use existing techniques for learning and personalization. Our procedure may be thought as a core step for developing automatically an adaptive hypermedia from any document collection.

Acknowledgments

The authors would like to thank J.D. Fekete from Université Paris Sud (France) for helpful discussion about treemaps.

References

- [Brusilovsky, 2001] Brusilovsky P., Adaptive Hypermedia, User Modeling and User-Adapted Interaction, 2001.
- [De Bra et al., 2003] De Bra P., Aerts A., Berden B., De Lange B., Rousseau B., Aha! The adaptive hypermedia architecture, HT'03, United Kingdom.
- [Hearst, 1997] Hearst M., 1997, TextTitling : Segmenting Text into multi-paragraph Subtopic Passages. Computational Linguistics. pp. 33-64.
- [Henze et al., 1999] Henze N., Nedjl W., Student modeling in an active learning environment using bayesian networks, User Modeling 1999.
- [Herder et al., 2002] Herder E., Van Dijk B., Personalized adaptation to device characteristics, International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, 2002.
- [Krishna et al., 2001] Krishna K., Krishnapuram R., A Clustering Algorithm for Asymmetrically Related Data with Applications to Text Mining. International Conference on Information and Knowledge Management, 2001, Atlanta, Georgia, USA. pp.571-573.
- [Sanderson et al., 1999] Sanderson M., Croft B., 1999, Deriving concept hierarchies from text. SIGIR Conference '99. pp.206-213.
- [Schneiderman, 1992] Schneiderman B., Tree visualization with tree-maps: 2-d space-filling approach, ACM Transactions on Graphics, Vol. 11, No. 1, January 1992.

Kernels on Prolog Ground Terms

Andrea Passerini and Paolo Frasconi

Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze

<http://www.dsi.unifi.it/neural/>

Abstract

We describe a family of kernels over untyped and typed Prolog ground terms and show that they can be applied for learning in structured domains, presenting experimental results in a QSPR task.

1 Introduction

Starting from the seminal work of Haussler on convolution kernels [Haussler, 1999], several researchers have proposed kernels on discrete data structures such as sequences, trees and graphs (see [Gärtner, 2003] for a review). Recently, kernels over complex individuals have been defined using higher order logic abstractions [Gärtner *et al.*, 2004]. The family of kernels developed in this paper originates from a specialization to first-order logic of such kernels, as it is designed to work on individuals represented as Prolog ground terms. There are several reasons for such a specialization. *Simplicity*: Prolog representations provide a simpler representational framework. *Sufficiency*: useful higher order structures such as sets can be simulated in Prolog. Types can also be introduced. In practice, Prolog based representations are sufficiently expressive for many application domains. *Legacy*: many inductive logic programming systems and knowledge bases are actually based on first order logic and Prolog is a well supported language. *Extendibility*: first-order logic programs have been extended to deal with uncertainty through the use of probability resulting in models such as stochastic and Bayesian logic programs (see [De Raedt and Kersting, 2003] for a review); these extensions can be very interesting in the context of machine learning.

We develop variants of the kernel on typed and untyped terms. The present formulation allows us to obtain a relatively simple proof of positive definiteness based on showing that kernels over terms are a special case of convolution kernels [Haussler, 1999]. We finally present experimental evidence of the usefulness of these kernels by learning the boiling point of alkanes from their chemical structure.

2 Kernels on Prolog ground terms

We begin with kernels on untyped terms. For a given program P we denote by $\mathcal{B}$ the Herbrand universe of P (i.e. the set of all ground terms that can be formed from symbols in P) by $\mathcal{C} \subset \mathcal{B}$ the set of all constants, and by $\mathcal{F}$ the ranked set of functors.

Definition 1 (Kernels on untyped terms). *The kernel between two terms t and s is a function $K : \mathcal{B} \times \mathcal{B} \mapsto \mathbb{R}$ defined inductively as follows:*

- if $s \in \mathcal{C}$ and $t \in \mathcal{C}$ then $K(s, t) = \kappa(s, t)$ where $\kappa : \mathcal{C} \times \mathcal{C} \mapsto \mathbb{R}$ is a valid kernel on constants;
- else if s and t are compound terms and have different arities or functors, i.e. $s = f(s_1, \dots, s_n)$ and $t = g(t_1, \dots, t_m)$, then

$$K(s, t) = \iota(f, g) \quad (1)$$

- where $\iota : \mathcal{F} \times \mathcal{F} \mapsto \mathbb{R}$ is a valid kernel on functors;
- else if s and t are compound terms and have the same arity and functor, i.e. $s = f(s_1, \dots, s_n)$ and $t = f(t_1, \dots, t_n)$, then

$$K(s, t) = \iota(f, f) + \sum_{i=1}^n K(s_i, t_i) \quad (2)$$

- in all other cases $K(s, t) = 0$.

We call κ and ι *atomic* kernels as they operate on non-structured symbols. A special but useful case is the atomic match kernel δ defined as $\delta(x, z) = 1$ if $x = z$ and $\delta(x, z) = 0$ if $x \neq z$.

A finer level of granularity in the definition of ground term kernels can be gained from the use of typed terms. This extra flexibility may be necessary to specify different kernel functions associated with constants of different type (e.g. numerical vs. categorical). It may also be necessary in order to specify different kernels associated to different arguments of compound terms. Our approach for introducing types is similar to that proposed in [Lakshman and Reddy, 1991]. We denote by $\mathcal{T}$ the ranked set of type constructors. The type signature of a function of arity n has the form $\tau_1 \times \dots \times \tau_n \mapsto \tau'$ where $n \geq 0$ is the number of arguments, $\tau_1, \dots, \tau_k \in \mathcal{T}$ their types, and $\tau' \in \mathcal{T}$ the type of the result. Functions of arity 0 have signature $\perp \mapsto \tau'$ and can be therefore interpreted as constants of type τ' . The type signature of a predicate of arity n has the form $\tau_1 \times \dots \times \tau_n \mapsto \Omega$ where $\Omega \in \mathcal{T}$ is the type of booleans. We write $t : \tau$ to assert that t is a term of type τ . We denote by $\mathcal{B}$ the set of all typed ground terms, by $\mathcal{C} \subset \mathcal{B}$ the set of all typed constants, and by $\mathcal{F}$ the set of typed functors. Finally we introduce a (possibly empty) set of *distinguished* type signatures $\mathcal{D} \subset \mathcal{T}$ that can be useful to specify ad-hoc kernel functions on certain compound terms.

Definition 2 (Kernels on typed terms). *The kernel between two typed terms t and s is defined inductively as follows:*

- if $s \in \mathcal{C}$, $t \in \mathcal{C}$, $s : \tau$, $t : \tau$ then $K(s, t) = \kappa_\tau(s, t)$ where $\kappa_\tau : \mathcal{C} \times \mathcal{C} \mapsto \mathbb{R}$ is a valid kernel on constants of type τ ;

A Learning Scheme for Generating Expressive Music Performances of Jazz Standards

Rafael Ramirez and Amaury Hazan

Music Technology Group
Pompeu Fabra University
Ocata 1, 08003 Barcelona, Spain
{rafael,ahazan}@iua.upf.es

Abstract

We describe our approach for generating expressive music performances of monophonic Jazz melodies. It consists of three components: (a) a melodic transcription component which extracts a set of acoustic features from monophonic recordings, (b) a machine learning component which induces an expressive transformation model from the set of extracted acoustic features, and (c) a melody synthesis component which generates expressive monophonic output (MIDI or audio) from inexpressive melody descriptions using the induced expressive transformation model. In this paper we concentrate on the machine learning component, in particular, on the learning scheme we use for generating expressive audio from a score.

1 Introduction

Expressive performance is an important issue in music which has been studied from different perspectives [Gabrielsson, 1999]. The main approaches to empirically study expressive performance have been based on statistical analysis (e.g. [Repp, 1992]), mathematical modelling (e.g. [Todd, 1992]), and analysis-by-synthesis (e.g. [Friberg, 1995]). In all these approaches, it is a person who is responsible for devising a theory or mathematical model which captures different aspects of musical expressive performance. Recently, there has been work on applying machine learning techniques to the study of expressive performance. Widmer [Widmer, 2002] has focused on the task of discovering general rules of expressive classical piano and recognizing famous pianists from their playing style. Lopez de Mantaras et al. [Lopez de Mantaras, 2002] reported on SaxEx, a case-based reasoning system capable of inferring a set of expressive transformations and applying them to a solo performance in Jazz. In this paper we describe an approach to investigate musical expressive performance based on inductive machine learning. In particular, we are interested in monophonic Jazz melodies performed by a saxophonist. Our work differentiates from that of Widmer in that, being focused on saxophone Jazz performances, we are interested in intra-note variations (e.g. vibrato) absent in piano, as well as melody alterations (e.g. onset deviations, ornamentations) which are normally considered performance

errors in classical music. The work of Lopez de Mantaras et al. is similar to ours but they are unable to explain their predictions. The deviations and changes we consider are on note duration, note onset, note energy, and intra-note features (e.g. attack, vibrato). The study of these variations is the basis of an inductive content-based transformation tool for generating expressive performances of musical pieces. The tool can be divided into three components: a melodic transcription component, a machine learning component, and a melody synthesis component. In the following, we briefly describe each of these components.

2 Melodic description

Sound analysis and synthesis techniques based on spectral models are used for extracting high-level symbolic features from the recordings. The sound spectral model analysis techniques are based on decomposing the original signal into sinusoids plus a spectral residual. From the sinusoids of a monophonic signal it is possible to extract information on note pitch, onset, duration, attack and energy, among other high-level information. This information can be modified and the result added back to the spectral representation without loss of quality. We use the software SMSTools which is an ideal tool for preprocessing the signal and providing a high-level description of the audio recordings, as well as for generating an expressive audio according to the transformations obtained by machine learning methods.

The low-level descriptors used to characterize the melodic features of our recordings are instantaneous energy and fundamental frequency. The procedure for computing the descriptors is first to divide the audio signal into analysis frames and compute a set of low-level descriptors for each analysis frame. Then, a note segmentation is performed using low-level descriptor values. Once the note boundaries are known, the note descriptors are computed from the low-level and the fundamental frequency values (see [Gomez et al., 2003] for details about the algorithm).

3 Expressive performance knowledge induction

Data set. The training data used in our experimental investigations are monophonic recordings of three Jazz standards

(*Body and Soul, Once I loved and Like Someone in Love*) performed by a professional musician (a saxophone player) at 11 different tempos around the nominal tempo. The resulting data set is composed of 1936 performed notes.

Descriptors. In this paper, we are concerned with note-level (in particular note duration, note onset and note energy) and intra-note-level (in particular intra-note pitch and amplitude shape) expressive transformations. Each note in the training data is annotated with its corresponding deviation and a number of attributes representing both properties of the note itself and some aspects of the local context in which the note appears. Information about intrinsic properties of the note include note duration, note metrical position, and note envelope information, while information about its context include the note Narmour group(s) [Narmour, 1990], duration of previous and following notes, and extension and direction of the intervals between the note and the previous and following notes.

Machine learning techniques. In order to induce predictive models for duration ratio, onset deviation and energy, variation, we have applied machine learning techniques such as regression trees, model trees and support vector machines (for a complete comparison of the accuracy of these techniques, see [Ramirez et al., 2005]). Among these techniques, model trees is the most accurate method, and thus, we have based the machine learning component of our tool on this method. We have also induced rule-based models [Ramirez et al., 2004] to *explain* the predictions made by our tool. In order to induce a predictive model for intra-note features we have devised a learning scheme roughly described as follows:

1. apply k-means clustering to all the notes in the data set. We decided to set the number of clusters to five. This decision was taken after analyzing a large number of notes in our data set and considering that there were basically five qualitatively different types of note shapes. We characterize each note in the data set by its attack, sustain and release.
2. apply a classification algorithm (i.e. classification trees) to predict the cluster to which the note belongs. In order to train our classifier we used the descriptors described above.
3. given a note and its cluster, apply a nearest neighbor algorithm to determine the most similar note in the cluster. We use the pitch and duration of a note as the distance measure, i.e. given a note, we look in the predicted cluster for the closest note in duration and in pitch. We are particularly interested in duration and pitch because we want to minimize the loss in sound quality when transforming the selected note to a note with the required pitch and the computed duration.

Once we obtain all the notes in a score by applying the learning scheme described above, we proceed to 'glue' the obtained notes together. Finally, we apply an algorithm to obtain smooth note transitions. A sample of a melody produced by our learning scheme can be found at www.iaa.upf.es/~rramirez/promusic/demo.wav.

4 Melody synthesis

The melody synthesis component transforms an inexpressive melody input into an expressive melody following the induced models. Given a melody score (i.e. an inexpressive description of a melody), our tool can either generate an expressive MIDI performance, or generate an expressive audio performance. In the second case, in addition to using the duration, onset and energy models for computing expressive deviations of these parameters, we apply the intra-note model to obtain the set of notes to be used to construct the audio expressive performance. In order to build the final audio performance we transform each of the obtained notes according to the computed duration, onset and energy deviations, and concatenate the transformed notes using an algorithm that optimizes the transitions between notes.

Acknowledgments

This work is supported by the Spanish TIC project ProMusic (TIC 2003-07776-C02-01). We would like to thank Emilia Gomez, Esteban Maestre and Maarten Grachten for processing the data.

References

- [Friberg, 1995] Friberg, A., A Quantitative Rule System for Musical Performance. PhD Thesis, KTH, Sweden, 1995.
- [Gabrielsson, 1999] Gabrielsson, A., The performance of Music. In D.Deutsch (Ed.), The Psychology of Music (2nd ed.) Academic Press, 1999.
- [Gomez et al., 2003] Gomez, E., Grachten, M., Amatriain, X., Arcos, J., "Melodic characterization of monophonic recordings for expressive tempo transformations", Proceedings of Stockholm Music Acoustics Conference, Stockholm, August 2003.
- [Lopez de Mantaras, 2002] Lopez de Mantaras, R., and Arcos, J. 2002. Ai and music from composition to expressive performance. AI Magazine 23(3).
- [Narmour, 1990] Narmour, E., The Analysis and Cognition of Basic Melodic Structures: The Implication Realization Model. University of Chicago Press, 1990.
- [Ramirez et al., 2004] Ramirez, R., Hazan, A., Gomez, E., Maestre, E., Understanding expressive transformations in saxophone jazz standards using inductive machine learning. Proceedings of Sound and Music Conference (SMC), Paris, October 2004.
- [Ramirez et al., 2005] Ramirez, R., Hazan, A., Modeling Expressive Music Performance in Jazz, Clearwater Florida, May 2005.
- [Rapp, 1992] Repp, B.H., Diversity and Commonality in Music Performance: an Analysis of Timing Microstructure in Schumann's 'Traumerei'. Journal of the Acoustical Society of America 104.
- [Todd, 1992] Todd, N., The Dynamics of Dynamics: a Model of Musical Expression. Journal of the Acoustical Society of America 91, 1992.
- [Widmer, 2002] Widmer, G., In Search of the Horowitz Factor: Interim Report on a Musical Discovery Project. Invited paper. In Proceedings of the 5th International Conference on Discovery Science, Lbeck, Springer-Verlag.

Incorporating a folding rule into inductive logic programming

David A. Rosenblueth

Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas

Universidad Nacional Autónoma de México

Apdo. 20-726, 01000 México D. F.

Mexico

drosenbl@servidor.unam.mx

Abstract

Many inductive logic programming systems have operators reorganizing the program so far inferred, such as the intra-construction operator of CIGOL. At the same time, there is a similar reorganizing operator, called the “folding rule,” developed in program transformation. We argue that there are advantages in using an extended folding rule as a reorganizing operator for inductive-inference systems. Such an extended folding rule allows an inductive-inference system not only to recognize already-learned concepts, but also to increase the efficiency of execution of inferred programs.

1 Introduction

At first glance, it might seem that inductive logic programming (ILP) and logic program transformation (LPT) have opposite objectives. While ILP is interested in generalizing from examples, LPT is careful to preserve correctness. Nevertheless, there are ILP operators, like the intra-construction operator of CIGOL, that do not generalize, and instead reorganize the program so far inferred. This suggests looking at logic program transformation rules to obtain better nongeneralizing operators for ILP.

One LPT rule, called “folding,” [Pettorossi and Proietti, 1998] is reminiscent of intra-construction, with some important differences. For example whereas intra-construction produces the definition of a new predicate (i.e. “concept”), folding receives as input such a definition, so that there is a separate, dedicated rule for creating a definition. The reason there is such a separate rule in LPT is that the derivation of efficiently executable programs is closely related to being able to fold using an already-existing definition.

Researchers have developed several reorganizing operators for ILP. Such operators can be found, for instance in the systems devised by Muggleton, Banerji, Ling, and Rouveirol and Puget. We have selected Muggleton’s intra-construction as a representative of these operators for performing a comparison with folding. The main ideas of our comparison, however, should carry over to other similar operators.

Throughout, A and B denote atoms, and $\tilde{Q}$ and $\tilde{R}$ denote tuples of atoms (we will assume definite logic programs). Also, $\text{msg}(e_1, e_2)$ denotes the most specific generalization of

expressions e_1 and e_2 , and “ $\simeq$ ” denotes equality up to variable renaming.

When used to fold two clauses, the *folding rule* infers (F) from (D_1) , (E_1) , (D_2) , and (E_2) :

$$\frac{\begin{array}{ll} B_1 \leftarrow \tilde{Q}_1 & (D_1) \\ A \leftarrow \tilde{Q}_1 \theta_1, \tilde{R} & (E_1) \end{array} \quad \begin{array}{ll} B_2 \leftarrow \tilde{Q}_2 & (D_2) \\ A \leftarrow \tilde{Q}_2 \theta_2, \tilde{R} & (E_2) \end{array}}{A \leftarrow B, \tilde{R} \quad (F)}$$

where $B = B_1 \theta_1 = B_2 \theta_2$. In addition, there are some syntactic restrictions for soundness, for which we refer the reader to [Pettorossi and Proietti, 1998]. An application of folding replaces (E_1) and (E_2) by (F).

Observe that this rule has the limitation of requiring both clauses to be folded (E_1) and (E_2) , to have the same head A and the same part of the body $\tilde{R}$ which is not an instance of the bodies of the definition (D_1) and (D_2) .

On the other hand, folding has the advantage of allowing the clauses (D_1) and (D_2) to be drawn from a previous program [Pettorossi *et al.*, 1996].

To be able to perform a comparison with *intra-construction*, we will assume that intra-construction consists of two parts: first the “invention” of a predicate, and then the clause-reorganization proper. The second part would then be:

$$\frac{\begin{array}{ll} B_1 \leftarrow & (D_1) \\ A_1 \leftarrow \tilde{R}_1 & (E_1) \end{array} \quad \begin{array}{ll} B_2 \leftarrow & (D_2) \\ A_2 \leftarrow \tilde{R}_2 & (E_2) \end{array}}{A \leftarrow B, \tilde{R} \quad (F)}$$

where

$$\begin{aligned} (A, \tilde{R}) &= \text{msg}((A_1, \tilde{R}_1), (A_2, \tilde{R}_2)) \\ B &= p(X_1, \dots, X_n) \\ B_i &= p(t_1^i, \dots, t_n^i) \end{aligned}$$

$$(A, \tilde{R}) \{X_1/t_1^i, \dots, X_n/t_n^i\} \simeq (A_i, \tilde{R}_i)$$

An application of the second part of intra-construction also replaces (E_1) and (E_2) by (F). Compared with folding, not only must the definition clauses be unit clauses, but their argument places must be terms that undo the generalization when resolving (F) with (D_1) and (D_2) . If we are introducing a new definition, such constraints might not be relevant. If, however, we wish to recognize an already-inferred concept or we are interested in increasing the efficiency of execution of our inferred program, it might be important to be able to relax such limitations.

2 A common extension of folding and intra-construction

As an illustration showing the need for increasing the efficiency of an inferred program, consider the *string-matching problem*. This problem consists in finding all occurrences (or only one occurrence, if any) of a string called the *pattern* within another string called the *text*. Suppose that our pattern is the string *aab*. If we give the following two examples, both of which are a text having an occurrence of this pattern:

$$\begin{aligned} s([a, a, b]) &\leftarrow \\ s([a, a, a, b]) &\leftarrow \end{aligned} \quad (1)$$

we can apply absorption, deriving:

$$s([a|X]) \leftarrow s(X) \quad (2)$$

A third example:

$$s([b, a, a, b]) \leftarrow$$

allows us to apply absorption again, inferring:

$$s([b|X]) \leftarrow s(X) \quad (3)$$

Clauses (1), (2), and (3) can be regarded as representing a finite-state automaton accepting the language $(a + b)^*aab$. However, these clauses are a nondeterministic program, requiring a quadratic time in the worst case to find an occurrence of the pattern. Note that this program cannot be converted to a deterministic version with intra-construction. A reason is that we would need to be able to apply intra-construction without inventing a new predicate at every application. (Such already-existing predicates represent edges going backwards in the corresponding automaton.) Allowing the use of already-existing definitions is accompanied by a possible violation of the conditions for intra-construction (i.e. the B_i 's may not have the required form).

This limitation suggests using the ordinary folding [Pettorossi and Proietti, 1998] rule as part of the operator repertoire of an ILP system. In fact, [Pettorossi *et al.*, 1996] show how to determinize a program similar to the one we just inferred with absorption (given as specification by the user), where the folding rule plays a central role.

The ordinary folding rule [Pettorossi and Proietti, 1998], however, is not adequate for inductive logic programming: We observed that this rule requires the set of atoms that do not match the bodies of the definition (i.e. A and $\tilde{R}$) be the same in each clause to be folded (i.e. (E_1) and (E_2)).

Now we give another illustration. Suppose an inductive-inference system has inferred the following description of an arch, where the example $arch([], beam, [])$ is not given until later:

$$\begin{aligned} arch([X|Y], beam, [X|Y]) &\leftarrow brick_or_block(X), \\ &\quad column(Y) \\ column([]) &\leftarrow \\ column([X|Y]) &\leftarrow brick_or_block(X), column(Y) \end{aligned} \quad (4)$$

If we now add the example:

$$arch([], beam, []) \leftarrow \quad (5)$$

both folding and intra-construction are unable to derive the desired concept from (4) and (5):

$$arch((X, beam, X)) \leftarrow column(X) \quad (6)$$

Being able to derive the concise version of *arch* (6) might be important if we have already the same concept under a different name:

$$gateway((X, beam, X)) \leftarrow column(X)$$

We have recently extended the ordinary folding rule so as to overcome these limitations of folding [Rosenblueth, 2005]. Our extension can be viewed as a common generalization of ordinary folding and intra-construction:

$$\begin{array}{c} B_1 \leftarrow \tilde{Q}_1 \quad (D_1) \qquad B_2 \leftarrow \tilde{Q}_2 \quad (D_2) \\ A_1 \leftarrow \tilde{Q}_1\theta_1, \tilde{R}_1 \quad (E_1) \qquad A_2 \leftarrow \tilde{Q}_2\theta_2, \tilde{R}_2 \quad (E_2) \\ \hline A_1 \leftarrow B_1\theta_1, \tilde{R}_1 \quad (F_1) \qquad A_2 \leftarrow B_2\theta_2, \tilde{R}_2 \quad (F_2) \\ \hline A \leftarrow B, \tilde{R} \quad (F) \end{array}$$

where

$$\begin{aligned} (A, \tilde{R}) &= msg((A_1, \tilde{R}_1), (A_2, \tilde{R}_2)) \\ B &= msg(B_1\theta_1\tau_1, B_2\theta_2\tau_2) \end{aligned}$$

Here, θ_i only acts on the variables occurring in $\tilde{Q}_i$, and τ_i only acts on the variables occurring in B_i and not occurring in $\tilde{Q}_i$ (i.e. the *head-only* variables in (D_i)). In addition, we preserve the syntactic conditions on θ_i developed for ordinary (multiple-clause) folding [Pettorossi and Proietti, 1998].

The main point of our folding rule is the computation of the substitutions τ_i in such a way that soundness is preserved. We distinguish two kinds of head-only variable instantiation: (a) *coverage*, taking a function symbol or variable from the substitutions ρ_i , where $(A_i, \tilde{R}_i) = msg((A_1, \tilde{R}_1), (A_2, \tilde{R}_2))\rho_i$ ($i = 1, 2$), and (b) *matching*, taking the function symbol from the corresponding position in the clause of the other single-clause folding.

The contributions of this poster over [Rosenblueth, 2005] are (1) the comparison of folding with intra-construction and (2) the development of inductively inferred example programs. We are currently investigating the possibility of embedding our folding rule in CIGOL.

References

- [Pettorossi and Proietti, 1998] Alberto Pettorossi and Maurizio Proietti. Transformation of logic programs. In Dov M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 5, pages 697–787. Oxford University Press, 1998.
- [Pettorossi *et al.*, 1996] Alberto Pettorossi, Maurizio Proietti, and Sophie Renault. Enhancing partial deduction via unfold/fold rules. In *Proc. 6th Int. Workshop on Logic Program Synthesis and Transformation*, pages 146–168, Stockholm, Sweden, 1996. Springer-Verlag. Lecture Notes in Computer Science No. 1207.
- [Rosenblueth, 2005] David A. Rosenblueth. A multiple-clause folding rule using instantiation and generalization. *submitted to Fundamenta Informaticae (Special Issue on Program Transformation)*, 2005.

Development of new techniques to improve Web search*

David Sánchez, Antonio Moreno

Department of Computer Science and Mathematics
University Rovira i Virgili (URV)

Avda. Països Catalans, 26. 43007 Tarragona (Spain)
{david.sanchez, antonio.moreno}@urv.net

Abstract

Web search engines are a great help for accessing web sites but they present several problems regarding semantic ambiguity. In order to solve them, we propose new methods for polysemy disambiguation of web resources and discovery of lexicalizations and synonyms of search queries.

1 Introduction

Search engines like Google have become an indispensable tool. However, their ranking algorithms are based only on a keyword's presence and have some serious handicaps that are reflected on the final results. On the one hand, if several ways of expressing the same search query exist (e.g. *synonyms*, *lexicalizations*, or even *morphological forms*), a considerable amount of relevant resources will be omitted. On the other hand, if the user selects a polysemic word as a query, the set of returned results will contain sites that are using that concept in different contexts (e.g. *Virus*).

In order to face these two problems, in this paper we present *new, automatic and autonomous methodologies for semantic disambiguation and classification of web resources and for discovery of lexicalizations and synonyms for a specific domain*. These techniques use a previously obtained taxonomy of terms and web resources through the method described in [Sanchez and Moreno, 2004] to perform the appropriate contextualization (see an example for the *Sensor* domain in figure 1).

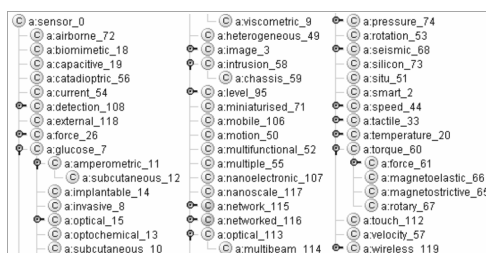


Figure 1. Example of an initial taxonomy of *Sensors*.

* The support of "Departament d'Universitats, Recerca i Societat de la Informació" of Catalonia is acknowledged.

2 Discovery of lexicalizations and synonyms

There are several methodologies that try to find lexicalizations for a keyword like *Latent Semantic Analysis* [Deerwester *et al*, 1990]. It considers that words that co-occur in the same documents are semantically related. However, it tends to return domain related words but, sometimes, not truly "equivalent" ones [Bhat *et al*, 2004]. Other techniques [Valarakos *et al*, 2004] identify lexicalizations with the assumption that they use a common set of 'core' characters. They can detect alternative spellings (e.g. *Pentium III*, *Pent. 3*), but not synonyms (e.g. *sensor* and *transducer*).

We have developed a novel methodology for discovering lexicalizations and synonyms using an initial taxonomy and a web search engine. Our approach is based on considering the longest branches (e.g. *subcutaneous amperometric glucose sensor*) of the taxonomy as a contextualization constraint and using them as the search query omitting the initial keyword (e.g. "*subcutaneous amperometric glucose*") for obtaining new webs. In some cases, those documents will contain equivalent words for the main one just behind the searched query that can be considered candidates for lexicalizations or synonyms (e.g. *subcutaneous amperometric glucose transducer*). For each candidate (see Table 1 with candidates for the *Sensor* domain), a relevance measure (1) is computed, obtaining a ranked candidate list. Those that overpass a minimum threshold computed in function on the number of multiwords considered will be selected.

$$relev = \#Dif_Web_Appear * (10^{(\#Multi_words_terms - 1)}) \quad (1)$$

Table 1. Lexicalizations and synonyms candidates for the *Sensor* domain (24 multiwords). Elements in **bold** are selected results.

Concept and derivates	Hits	Webs	Multi	Relev.
sensor, sensors	437	294	24	2E25
transducer, transducers	28	19	3	1900
system	4	4	2	40
probe	2	2	2	20
transmitter	11	11	1	11
measurement	10	7	1	7

3 Polysemy detection and semantic clustering

In this case our goal is to be able to group different concepts (and their associated web sites) of the taxonomy associated to a search query in sets according to the concrete sense in which they are used. So, the user will obtain disambiguated sets of relevant web sites for the desired domain.

For that purpose, many techniques [Ide and Véronis, 1998] use an external dictionary to select the most appropriate meaning in each case. However, performing this classification without previous semantic knowledge is not a trivial process [Sanderson and Croft, 1999]. In this sense, we can take profit from the context where each concept has been extracted, concretely, the web documents that contain it. We can assume that each website included in the taxonomy is using the initial keyword (e.g. *virus*) in a concrete sense, so concepts that are selected from the analysis of that single document (e.g. *linux*, *email*) belong to the domain associated to the same keyword's meaning. Applying this idea over a significant amount of web sites we can find, as shown in figure 2 for the *Virus* example, quite consistent semantic relations between concepts.

The algorithm performs a clusterization process with the list of terms of a specific level of the taxonomy. For each pair, a similitude measure (2) is computed based on the amount of coincidences between the concept's URL sets. The resulting similarity matrix allows us to detect the most similar concepts and join them. A new matrix is computed for the resulting set of classes. This process is repeated until no more concepts can be joined (there are no coincidences between URLs), building a dendrogram (see figure 2).

$$sim(A, B) = \text{Max} \left(\frac{\#Coin(URL(A), URL(B))}{\#URL(A)}, \frac{\#Coin(URL(A), URL(B))}{\#URL(B)} \right) \quad (2)$$

The result is a partition (with 2 elements for the *virus* domain as shown in figure 2) of classes that groups the concepts and web resources that belong to a specific meaning. The number of final classes is automatically discovered by the clustering algorithm thanks to the constrained joining criteria. Note that this methodology can be applied to a set of terms at any level of the taxonomy.

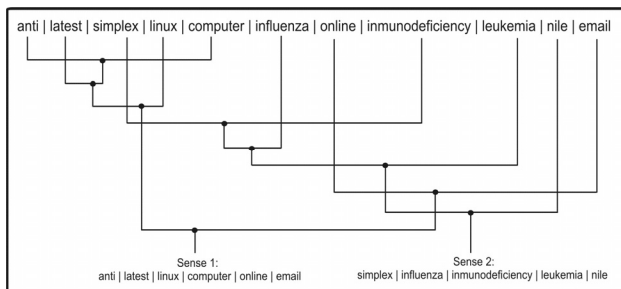


Figure 2. Dendrogram representing semantic associations between the classes found for the keyword “virus”. Two final clusters are obtained: Sense 1 groups the classes associated to “computer program” and Sense 2 the ones related to “biological agent”.

4 Conclusions

Taking into consideration the amount of resources available easily on the Web, we believe that methodologies that ease the search of information should be developed. Standard search engines are widely used for this task but they present serious limitations due to their pattern-search algorithms because they lack any kind of semantic content.

On the one hand, *polysemy* becomes a serious problem when the retrieved resources obtained from the search engine are only based on the keyword's presence; in this case, we propose an automatic approach for *polysemy disambiguation of concepts and web resources*. On the other hand, in order to extend the search widely and retrieve the largest amount of resources that are *semantically* relevant for the query specified, an *algorithm for discovering alternative keywords (lexicalizations and synonyms)* for the same domain is also proposed. This is very useful for domains with a little amount of available web resources.

Moreover, it is important to note that the proposed methodologies perform in an *automatic* and *autonomous* way, allowing to maintain the results updated easily by performing executions as frequently as desired. This is a very useful feature when dealing with a highly changing environment like the Web and distinguishes our approach from other similar ones in which user's interaction or a thesaurus like WordNet are required [Lamparter *et al*, 2004].

References

- [Bhat *et al*, 2004] V. Bhat, T. Oates, V. Shanbhag and C. Nicholas: Finding aliases on the web using latent semantic analysis. *Data Knowledge & Engineering* 49, 2004.
- [Deerwester *et al*, 1990] S. Deerwester, S. Dumais, T. Landauer, G. Furnas and R. Harshman: Indexing by latent semantic analysis, *Journal of the American Society of Information Science* 41 (6) pp. 391-407. 1990.
- [Ide and Véronis, 1998] N. Ide and J. Véronis: Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics* 24. 1998.
- [Lamparter *et al*, 2004] S. Lamparter, M. Ehrig and C. Tempich: Knowledge Extraction from Classification Schemas. ODBASE 2004, LNCS 3290, 2004.
- [Sanchez and Moreno, 2004] D. Sanchez and A. Moreno: Automatic Generation of Taxonomies from the WWW. In: *Proceedings of the 5th International Conference on Practical Aspects of Knowledge Management*. LNAI, vol. 3336, Vienna, 2004.
- [Sanderson and Croft, 1999] M. Sanderson, B. Croft: Deriving concept hierarchies from text. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1999, Berkeley, USA.
- [Valarakos *et al*, 2004] A.G. Valarakos, G. Paliouras, V. Karkaletsis and G. Vouros: Enhancing Ontological Knowledge Through Ontology Population and Enrichment, EKAU 2004, LNAI 3257, 2004.

Intimate Learning: A Novel Approach for Combining Labelled and Unlabelled Data

Zhongmin Shi and Anoop Sarkar

School of Computing Science, Simon Fraser University, Canada
{zshi1,anoop}@cs.sfu.ca

Abstract

This paper introduces a new bootstrapping method closely related to co-training and scoped-learning. The method is tested on a Web information extraction task of learning course names from web pages in which we use very few labelled items as seed data (10 web pages) and combine with an unlabelled set (174 web pages). The overall performance improved the precision/recall from 3.11%/0.31% for a baseline EM-based method to 44.7%/44.1% for intimate learning.

1 Intimate learning

The expensive nature of labelling data for machine learning methods and the lack of success in using purely unsupervised methods have motivated the study of learning methods that combine labelled and unlabelled data. Successful methods in this area include bootstrapping methods like co-training and scoped-learning. In this paper we introduce a novel method called *intimate learning* for bootstrapping that is related to these methods. The task is to learn the target function $h_t(X) \in Y$ where X is the set of all feature sets and Y is the set of class labels. The input to our learning algorithm is a set of n labeled examples of the form $(\mathbf{x}_i, y_i)$. $\mathbf{x}_i \in X$ has p_i features $x_{i1}, x_{i2}, \dots, x_{ip_i}$ associated with the i th example. $y_i \in Y$ is the label of the i th example. In *intimate learning*, we find another class label $y'_i \in Y'$ which is relevant to y_i and we assume that a new function $h'(X) \in Y'$ requires fewer labelled items to learn. We say that Y' is relevant to Y if accurately finding $y'_i \in Y'$ can help in identifying $y_i \in Y$ in example i . We then create the new target function $h(X, h'(X)) \in Y$ that performs the same classification as $h_t(X)$ does. We call Y' the **intimate class**, $h'(X)$ the **intimate function**. Intimate learning is related to the co-training algorithm [Blum and Mitchell, 1998], in which for training examples $(\mathbf{x}_i, y_i)$, $\mathbf{x}_i$ is decomposed into a pair $(\mathbf{x}_{i1}, \mathbf{x}_{i2})$ corresponding to two different “views”, and the target function $h(X) = h_1(X_1) = h_2(X_2)$ predicting a single label class. While in our model, X has only one “view” but labeled into two classes (target and intimate classes), i.e., $h_t(X) \neq h'(X)$ ¹. Other related work is scoped-learning [Blei *et al.*, 2002], which uses a classifier trained on global features from the entire training data and classifiers trained on

scope-limited features which are more specific to local subsets of the data.

2 The Information Extraction task

We apply intimate learning to the problem of identifying course names from those pages identified as course web pages in WebKB which consists of web pages collected from four universities.² For our experiment we used: 10 web pages as seeds, 174 web pages as unlabeled training data and 40 web pages as test data. All web pages are tokenized using space and punctuation symbols. We used an EM-based decision list learning algorithm [Collins and Singer, 1999] as our baseline method for combining labeled and unlabeled data (more details in §2.1). The course name feature model is initialized using the seed data and then trained using EM on the training data. Our observation on the training data indicated that the course number, which does not form part of our target label is likely to co-occur with a course name. Course names have similar characteristics to other named entities like names of people or organizations. Compared with course names, course numbers have far more regular forms, which usually consist of the department abbreviation and the number of the course, e.g., *CMPT 825*. As a result identifying a course number is easier and we take this class to be our intimate class Y' . Our target class Y determines whether a string is a course name. In this paper, we have Y and Y' with one element each (*+course-name* and *+course-number*), rather than a 2-class classification task (*+course-name*, *-course-name*). See §2.1 for details on how test examples are handled. We explain the details of our algorithm using Figure 1 (for now, ignore the dashed arrow). The algorithm is a two-stage process. The left half of Figure 1 illustrates the first stage, in which the intimate class is learned by a classification algorithm that is identical to the EM-based baseline system (see §2.2). The second stage in the right half implements almost the same EM-based algorithm, but with course numbers added from the first stage as an extra feature towards learning of the course name (the darker line with an arrow).

2.1 EM Learning for course number identification

The features used for identifying a course number are:

- The HTML format of the course number, which is the pair of HTML tags before and after the course number

¹Intimate learning is *not* the same as feature selection: Y' is not directly observed. Furthermore Y' doesn't determine Y ; X combined with Y' can improve prediction of Y .

²Other applications of intimate learning include finding product names in web pages, which appear together with prices, quantities, locations which are generally much easier to identify.

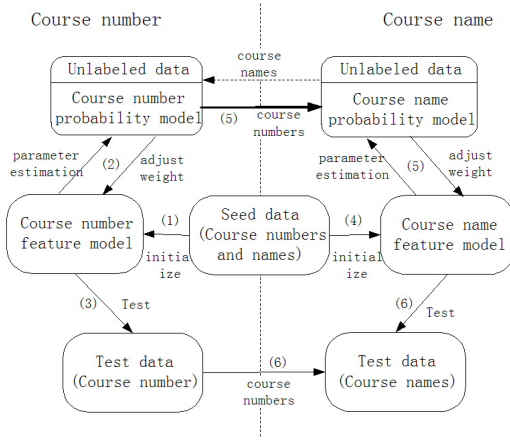


Figure 1: Course name identification system chart

respectively.

- Department name S and length $|S|$.
- Course id: integer I and its digit length $|I|$.
- The separator symbol sequence between S and I .

Features are collected for each string s_j from the seed data. We estimate the conditional probability $F_{num} = P(y' | x_{i,j})$ of seeing the label y' given the feature $x_{i,j}$. F_{num} defines a decision list of rules $x_{i,j} \rightarrow y'$ ranked by confidence score $P(y' | x_{i,j})$. In training, we use EM to (re)estimate the course number probability model $P_{num}(y' | s)$, where s is the input string. In the E-step, P_{num} is re-estimated based on F_{num} . We assume all features do not equally contribute to course number identification and assign different weight c_i to x_i , and each parameter of P_{num} is computed by:

$$P(y' | s_j) = \frac{\sum_i c_i \cdot P(y' | x_{i,j})}{\sum_i c_i} \quad (1)$$

In the M-step, F_{num} is adjusted with respect to P_{num} :

$$P(y' | x_{i,j}) = \frac{\text{Count}(x_{i,j}) \cdot P(y' | x_{i,j}) + P(y' | s_j)}{\text{Count}(x_{i,j}) + 1} \quad (2)$$

Since y' is a single label +course-number, for test examples we pick those to be labelled as a course number by using 2-means clustering to separate those examples that have high confidence scores (high $P(y' | x_{i,j})$ values) from those examples that have low confidence scores. This method avoids hand-picking or using a held-out set to pick a confidence threshold.

2.2 Course name identification

The learning procedure of the course name identification is the same as that for the course number, except that the course number identified using the model defined in §2.1 becomes an important feature in predicting whether an example is labelled as a course name. For each candidate course name, the features used are:

- *Intimate class*: course number preceding candidate
- The pair of HTML tags before and after course name.

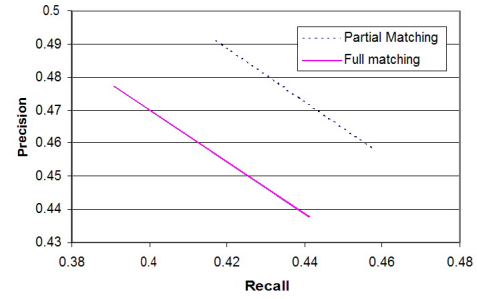


Figure 2: Experimental results of course name identification. The precision and recall curves for partial and full matchings.

- Each word in the course name and the number of words.
- Each separator symbol between the course number and course name and total number of such symbols.

The course name feature model F_{nam} is initialized from the seed data and trained by the EM algorithm defined in §2.1. Applying the trained feature model to the test data generates all course name candidates with their probabilities. We again apply the 2-means clustering algorithm defined in §2.1. Note that the chosen candidates are individual words instead of the full string as a course name. We simply group contiguous words as a single course name. The **baseline system** is identical to the course name feature model F_{nam} except that it does not use the identified course number as an input feature.

3 Experiments and discussion

Two metrics are applied to the performance evaluation of course name identification: *partial matching*, in which the course name is correctly recognized if any of its words is predicted, and *full matching*, in which the course name is correctly predicted only if all words of the course name are predicted and in the correct order. The precision/recall of the baseline system is **3.11%/0.31%** for *full matching*. Figure 2 illustrates the performance of our implementation on course name identification, for full and partial matchings. Our experiments show that the intimate learning algorithm exhibits significant gains in performance over the baseline system obtaining **44.7%/44.1%** for *full matching* and 45.8%/46.5% for *partial matching*. Since the course number and course name are two related classes, in addition to intimate learning, a co-training-based extension can also be applied to training one class by the other, and vice versa as shown by the dashed arrow in Figure 1. For details and full set of references, please refer to <http://natlang.cs.sfu.ca/researchProject.php?s=299>.

References

- [Blei et al., 2002] D. Blei, D. Bagnell, and A. McCallum. Learning with scope, with application to information extraction and classification. In *Proc. UAI*, 2002.
- [Blum and Mitchell, 1998] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. COLT*, 1998.
- [Collins and Singer, 1999] M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proc. EMNLP*, 1999.

Collective Object Identification

Parag Singla Pedro Domingos

Department of Computer Science and Engineering

University of Washington

Seattle, WA 98195-2350, U.S.A.

{parag, pedrod}@cs.washington.edu

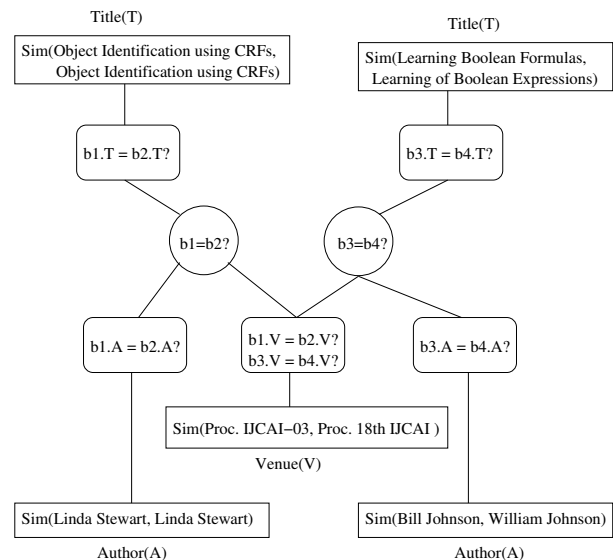
In many domains, the objects of interest are not uniquely identified, and the problem arises of determining which observations correspond to the same object. For example, in information extraction and NLP we need to determine which noun phrases refer to the same entity. When merging multiple databases, a problem of keen interest to many large scientific projects, businesses, and government agencies, we need to determine which records represent the same entity and should therefore be merged. This problem, first placed on a firm statistical footing by Fellegi and Sunter [1969], is known by the name of object identification, record linkage, de-duplication and others. Most approaches described to solve this problem are variants of the original Fellegi-Sunter model, in which object identification is viewed as a classification problem: given a vector of similarity scores between the attributes of two observations, classify it as “Match” or “Non-match.” A separate match decision is made for each candidate pair, followed by transitive closure to eliminate inconsistencies. Typically, a logistic regression model is used. We call this the standard model.

Making match decisions separately ignores that information gleaned from one match decision may be useful in others. For example, if we find that a paper appearing in *Proc. IJCAI-03* is the same as a paper appearing in *Proc. 18th IJCAI*, this implies that these two strings refer to the same venue, which in turn can help match other pairs of IJCAI papers. In this paper, we propose an approach which accomplishes this propagation of information. Our approach makes decisions collectively, performing simultaneous inference for all candidate match pairs, and allowing information to propagate from one candidate match to another via the attributes (or fields) they have in common. Our model is based on conditional random fields [Lafferty *et al.*, 2001]. We call our model the collective model. Figure 1(a) shows a four-record bibliography database and 1(b) shows the corresponding graphical representation for the candidate pairs (b_1, b_2) and (b_3, b_4) in the collective model. There are three types of nodes in the figure. *Record-match nodes* are Boolean-valued and they correspond to asking the question “Is record b_i the same as record b_j ?” *Field-match nodes* are also Boolean-valued and they correspond to asking the question “Do field values $b_i.F$ and $b_j.F$, for the field F, represent the same underlying property?” *Field-similarity nodes* are real-valued nodes taking values in the domain $[0, 1]$ and they encode how similar two field val-

Record	Title	Author	Venue
b1	Object Identification using CRFs	Linda Stewart	Proc. AAAI-05
b2	Object Identification using CRFs	Linda Stewart	Proc. 20th AAAI
b3	Learning Boolean Formulas	Bill Johnson	Proc. AAAI-05
b4	Learning of Boolean Expressions	William Johnson	Proc. 20th AAAI



(a) A bibliography database



(b) Collective model (fragment)

Figure 1: Example of collective object identification. For clarity, we have omitted the edges linking the record-match nodes to the corresponding field-similarity nodes.

ues are, according to a pre-defined similarity measure. The values of these nodes can be directly computed from data, and hence they are also called the *evidence nodes*. Intuitively, an edge between two nodes represents the fact their

values directly influence each other. Note how dependencies flow through the shared field-match node corresponding to the venue field. Inferring that b_1 and b_2 refer to the same underlying paper will lead to the inference that the corresponding venue strings “Proc. IJCAI-03” and “Proc. 18th IJCAI” refer to the same underlying venue, which in turn might provide sufficient evidence to merge b_3 and b_4 . In general, our model can capture complex interactions between candidate pair decisions, potentially leading to better object identification.

For random fields where maximum clique size is two and all non-evidence nodes are Boolean, the inference problem can be reduced to a graph min-cut problem, provided certain constraints on the parameters are satisfied [Greig *et al.*, 1989]. Our formulation of the problem satisfies these constraints. Since min-cut can be solved exactly in polynomial time, we have a polynomial-time exact inference algorithm for our model. We follow the standard approach of gradient descent to learn the parameters. Calculating the exact derivative is intractable as it involves an expectation over an exponential number of configurations. We use a voted perceptron algorithm [Collins, 2002], which approximates this expectation by the feature counts of the most likely configuration, which we find using our polynomial-time inference algorithm with the current parameters.

Combining models is often a simple way to improve accuracy. We combine the standard and collective models using logistic regression. For each record-match node in the training set, we form a data point with the outputs of the two models as predictors, and the true value of the node as the response variable. We then apply logistic regression to this dataset. Notice that this still yields a conditional random field.

We performed experiments on real and semi-artificial databases, comparing the performance of (a) the standard Fellegi-Sunter model using logistic regression, (b) the collective model, and (c) the combined model.

The first set of experiments was on Cora database, which is a collection of 1295 different citations to computer science research papers. We cleaned it up by correcting some labels and filling in missing values. This cleaned version contains references to 132 different research papers. We used the author, venue, and title fields. The second set of experiments was done on the BibServ.org database, which is the result of merging citation databases donated by its users, Citeseer, and DBLP. We experimented on the user-donated subset of BibServ, which contains 21,805 citations. Table 1 reports the results for these databases. The combined model gives the best performance on both Cora and BibServ, followed by the collective model. Transitive closure helps these on Cora but hurts all models on BibServ (where recall was close to 100% even without transitive closure). The best combined model outperforms the best standard model in F-measure by about 2% on Cora and 3% on BibServ.

To further observe the behavior of the algorithms, we generated variants of the Cora database by taking distinct field values from the original database and randomly combining them to generate distinct papers. Figures 2(a) and 2(b) compare the performance of the collective and standard models as number of clusters and level of distortion in the data are

Table 1: F-measures on Cora and BibServ before and after transitive closure.

Model	Cora		BibServ	
	Before	After	Before	After
Standard	86.9%	84.7%	82.7%	68.5%
Collective	87.4%	88.9%	82.8%	73.6%
Combined	85.8%	89.0%	85.6%	76.0%

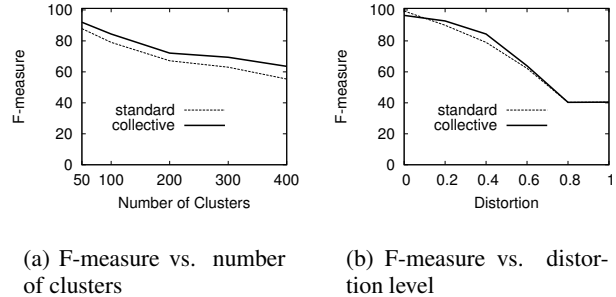


Figure 2: Experimental results on semi-artificial data.

varied, respectively.¹ The collective model clearly dominates the standard model over a broad range of number of clusters and level of distortion.

In summary, determining which observations correspond to the same object is a key problem in information integration, citation matching, natural language, vision, and other areas. We have developed a collective approach to this problem, where information propagates among related decisions via shared field values, and shown experimentally that it outperforms the standard one of making decisions independently.

Acknowledgments

This research was partly supported by ONR grant N00014-02-1-0408, by a gift from the Ford Motor Co., and by a Sloan Fellowship awarded to the second author.

References

- [Collins, 2002] M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. EMNLP-02*, pages 1–8, 2002.
- [Fellegi and Sunter, 1969] I. Fellegi and A. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.
- [Greig *et al.*, 1989] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51:271–279, 1989.
- [Lafferty *et al.*, 2001] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th ICML*, pages 282–289, 2001.

¹For clarity, we have not shown the curves for the combined model, which are similar to the collective one’s.

Question Classification by Structure Induction*

Menno van Zaanen and Luiz Augusto Pizzato and Diego Mollá

Division of Information and Communication Sciences (ICS)

Department of Computing

Macquarie University

2109 Sydney, NSW

Australia

{menno, pizzato, diego}@ics.mq.edu.au

Abstract

In this article we introduce a new approach (and several implementations) to the task of question classification. The approach extracts structural information using machine learning techniques and the patterns found are used to classify the questions. The approach fits in between the machine learning and handcrafting of regular expressions (as it was done in the past) and combines the best of both: classifiers can be generated automatically and the output can be investigated and manually optimised if needed.

1 Introduction

Before a question answering (QA) system can answer a question, it needs to have an idea what the question is about. One of the principal tasks of the question analysis stage of a QA system is the determination of the expected answer type (EAT). Finding the EAT of a question is called question classification (or EAT classification) [Hermjakob, 2001].

In this work we introduce a new approach to the problem of EAT classification, based on structural information that can be extracted from the questions. Re-occurring structures found in questions, such as “How far ...” may help finding the correct EAT (in this case “distance”) for a particular question. The approach described here automatically finds these structures during training and uses this information when classifying new questions.

Our approach combines the two main methods to question classification: machine learning and pattern matching. Using machine learning, patterns are extracted from the training data. These patterns serve as regular expressions during the classification task.

In the next two sections, we will describe two systems that fit into this approach. The first one uses a grammatical inference system and the second one is based on tries.

2 Alignment-Based Learning Classifier

The structure extraction phase of the first system is done by Alignment-Based Learning (ABL) [van Zaanen, 2002]. ABL

*This work is supported by the Australian Research Council, ARC Discovery Grant no. DP0450750.

“DESC”	(What)(is (caffeine))
“DESC”	(What)(is (Teflon))
“LOC”	(Where) is (Milan)
“LOC”	What (are the twin cities)

Table 1: Example sentences with ABL structure

is a generic unsupervised grammatical inference framework that learns structure from plain text sentences. The underlying idea of ABL is that constituents can be interchanged. To give an example, if we exchange the noun phrase *the man* in the sentence *He sees the man* with another noun phrase *a woman*, we get another valid sentence: *He sees a woman*. This process can be reversed (by aligning sentences) and possible constituents, called hypotheses can be found. This is called the alignment learning phase, one of the three phases of ABL, and the one that we use in this article. Table 1 shows an example of the structure learned from a toy corpus of 4 sentences.

In the training phase, regular expressions associated with the structures found are stored together with the EAT of the corresponding questions. Several EATs with their frequencies may be stored if a regular expression matches several questions.

We have experimented with two implementations. The first implementation, which we call *hypo*, uses the words in the hypotheses (i.e. the words between the brackets) to form regular expressions which are stored together with the EATs of the associated questions. The second implementation, called *unhypo*, uses the words that are left after removing each hypothesis. For example, the *hypo* version uses the first question of Table 1 to produce the patterns */What/*, */is caffeine/*, and */caffeine/*, whereas *unhypo* produces the patterns */is caffeine/*, */What/*, and */What is/*.

During the classification phase we have experimented with two further approaches that differ on the use of the frequency counts of the matched regular expressions. The first method (called *default*) increments the counts of the EATs of the question by the frequency that is stored with the EATs of the regular expression. The second method (called *prior*) will simply increment the counts by 1. When all regular expressions are tried, both methods select the EAT with the highest count. If there several expressions with the same count, the

default method makes a random choice, whereas the *prior* method selects the EAT with the highest overall frequency.

Finally, we have experimented with the impact of the parts of speech (POS) as a simple approach to disambiguate words. We used Brill’s tagger [Brill, 1992] and the resulting POS information is simply combined with the plain words. Adjacent words that have the same POS are combined into one token. For example the question *Who is Federico Fellini?* is, after POS tagging, divided into three tokens: (*Who*, WP), (*is*, VBZ) and (*Federico Fellini*, NNP).

3 Trie Classifier

The second system uses a trie structure. A trie $T(S)$ is a data structure defined by a recursive rule $T(S) = \{T(S/a_1), T(S/a_2), \dots, T(S/a_r)\}$. S is a set of sequences whose elements are taken from the alphabet A . S/a_n is the set of strings that contains all strings of S that start with a_n but stripped of that initial element [Clément *et al.*, 1998].

During the learning phase, all questions are inserted into a trie structure that contains, in addition to the token, the EAT and frequency information (the number of questions that use that path in the trie).

During the classification phase, the trie is traversed in the usual way. If the new question is a prefix of a training question the traversal is trivial, and the node at the end of the traversal path indicates the EAT of the question.

To find a path for unseen questions, non-matching nodes are skipped in a methodical way in what we call the look-ahead process. Let us say that question tokens of question $Q = q_1 q_2 \dots q_n$ match up to q_i : $T(S/q_1/q_2/\dots/q_i)$, and $T(S/q_1/q_2/\dots/q_i/q_{i+1})$ does not exist. The look-ahead process then builds a set of sub-tries of the form $\{T(S/q_1/q_2/\dots/q_i/\beta/q_{i+2}) | \beta \in A\}$. The sub-trie with the highest frequency associated is selected, and the process continues with q_{i+2} until all tokens are consumed.

There are two variations of the above process. In the *strict* method, β and q_{i+1} must have the same POS tag. If the resulting set of sub-tries is empty, the search process stops and the EAT is retrieved from the node q_i in the trie. In the *flex* method, if an empty set of sub-tries is retrieved, we consider β as a sequence of question words until we find a question word equal to q_{i+2} .

We also experimented with a set of variations that have no POS information. In this case, β and q_{i+1} must be exactly the same token. Again we allow for a *strict* and a *flex* version.

4 Results

To compare our systems with current machine-learning methods, we have used the same data as [Zhang and Sun Lee, 2003]. This is a collection of 5,452 training questions and 500 test questions. The data have 6 coarse-grained classes and 50 fine-grained classes.

Table 2 indicates the precision of the questions classified with the coarse-grained classes for all our systems and for a baseline that always selects the most frequent class according to the training data. This baseline is, of course, the same for the plain words and POS tagged data. All our implementations performed well above the baseline.

			words	POS
Baseline			0.188	0.188
ABL	hypo	default	0.516	0.682
		prior	0.554	0.624
	unhypo	default	0.652	0.638
		prior	0.580	0.594
Trie	strict		0.844	0.812
	flex		0.850	0.794

Table 2: Results on the coarse-grained data

The results show that the POS information helps the performance of the ABL method but not of the trie-based method. Overall, the trie-based approach outperforms the ABL implementations. We obtained similar results with the fine-grained data (not shown for reasons of space).

Our best results fall close to the best-performing system in the bag-of-words and bag-of- n -grams versions of [Zhang and Sun Lee, 2003]. Their results ranged from 75.6% (Nearest Neighbours on words) to 87.4% (SVM on n -grams). Given the simplicity of our methods and their potential for further improvement, this is encouraging.

5 Conclusion

The results on the annotated questions of the TREC10 data show that the approach is feasible and both systems generate acceptable results. We expect that future systems that fall in the structure induced question classification approach (for example, based on other grammatical inference systems) will result in even better performances.

The automatically learned regular expressions can be inspected and extended by humans. The systems therefore combine the advantages of machine-learning and pattern-matching methods.

Additionally, we think that the structure found by the systems can be used to find the focus of the question.

References

- [Brill, 1992] Eric Brill. A simple rule-based part-of-speech tagger. In *Proc. ANLP*, pages 152–155, Trento, Italy, 1992.
- [Clément *et al.*, 1998] J. Clément, P. Flajolet, and B. Vallée. The analysis of hybrid trie structures. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 531–539, Philadelphia:PA, USA, 1998. SIAM Press.
- [Hermjakob, 2001] Ulf Hermjakob. Parsing and question classification for question answering. In *Proc. ACL/EACL Workshop on Open-Domain Question Answering*, 2001.
- [van Zaanen, 2002] Menno van Zaanen. *Bootstrapping Structure into Language: Alignment-Based Learning*. PhD thesis, University of Leeds, Leeds, UK, January 2002.
- [Zhang and Sun Lee, 2003] Dell Zhang and Wee Sun Lee. Question classification using support vector machines. In Charles Clarke, Gordon Cormack, Jamie Callan, David Hawking, and Alan Smeaton, editors, *Proc. SIGIR*, pages 26–32, New York:NY, US, 2003. ACM Press.

An Inductive Database for Mining Temporal Patterns in Event Sequences

Alexandre Vautier, Marie-Odile Cordier and René Quiniou

Irisa - DREAM Project Campus de Beaulieu 35042 RENNES Cedex, France

{Alexandre.Vautier,Marie-Odile.Cordier,Rene.Quiniou}@irisa.fr

1 Introduction

Data mining aims at discovering previously unknown and potentially useful information from large collections of data. The discovered knowledge, in the form of *patterns*, is extracted without any extra information about data. A pattern is a representation of a concept (rules describing properties of data, clusters in databases, etc.) that generalizes some data in a database.

Imielinski and Mannila [Imielinski and Mannila, 1996] designed a data mining formalization framework named inductive database (IDB). The main goal of IDBs is to manage knowledge discovery applications just as database management systems successfully manage business applications. IDBs contain patterns in addition to data. The data mining process is modeled as an interactive process in which users can query data as well as patterns. IDBs represent also a good way to take advantage of the efficiency of database algorithms by using jointly data mining and database algorithms.

Some IDBs implementations [Lee and De Raedt, 2003] use sequences of symbols as data and complex sequences of symbols as patterns (sequences with gaps). In many problems one often needs to discover patterns which are more sophisticated than a simple sequence of symbols. To this end, we propose to extend IDB patterns by adding numerical constraints. In this proposal, the data are event sequences and the patterns are chronicles. A chronicle is composed of events and numerical temporal constraints, in the form of numerical intervals, on delays between their occurrences. Figure 1 illustrates a chronicle $\mathcal{C}_1$ and an event sequence l_1 .

A chronicle is recognized in an event sequence as follows: a set o_i of events of the sequence l_1 instantiates the chronicle $\mathcal{C}_1$ if every event of $\mathcal{C}_1$ is instantiated by an event of o_i respecting type and temporal constraints. A set o_i is called a *chronicle instance* of $\mathcal{C}_1$. For example, the event sequence l_1 has six instances of the chronicle $\mathcal{C}_1$: $|\mathcal{I}_{\mathcal{C}_1}(l_1)| = 6$.

2 Time in data mining

Several solutions have been proposed to tackle the problem of time integration in data mining. Some pattern extraction techniques use symbol ordering as a model of time. For instance, Winepi and Minepi [Mannila *et al.*, 1997] extract patterns represented by serial or parallel sequences of symbols. SeqLog [Lee and De Raedt, 2003] extracts ordered sequences

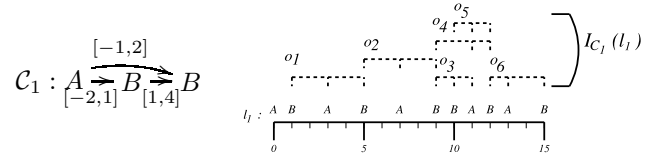


Figure 1: A chronicle $\mathcal{C}_1$ of size 3: intervals labeling edges represent bounds on the distance between event times as $-2 \leq t_B - t_A \leq 1$. Right, 6 instances of $\mathcal{C}_1$ in a sequence l_1 .

of symbols with gaps. Parallel and serial notions can also be expressed in chronicles. In addition, they can state numerical constraints on delay between events.

Time can be also represented by a special numerical attribute. For instance, Yoshida [Yoshida *et al.*, 2000] extracts frequent itemsets containing a temporal feature. Their algorithm extracts “delta-patterns” that are ordered lists of itemsets with time intervals between two successive itemsets (as for instance: $\{b\} \xrightarrow{[17,19]} \{a,c\} \xrightarrow{[5,12]} \{b,c\}$) which specify the numerical temporal constraints between the itemsets. However, interval bounds of delta-patterns are always positive whereas interval bounds of chronicles can be negative.

Our goal is to discover chronicles that are frequent in some event sequences and infrequent in other ones. For instance, let l_2 and l_3 be the event sequences of figure 2. A complex query such that $freq(\mathcal{C}, l_2) \geq 3 \wedge freq(\mathcal{C}, l_3) \leq 1$ asks for the set of chronicles that are frequent in l_2 and infrequent in l_3 according respectively to thresholds 3 and 1. Chronicles $\mathcal{C}_S$ and $\mathcal{C}_G$ (fig. 2) satisfy this query because both cover three instances in l_2 and only one in l_3 .

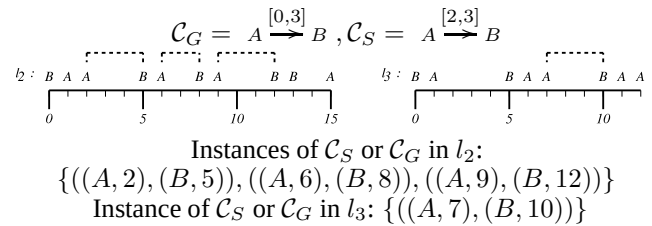


Figure 2: Chronicles search under frequency constraints

Furthermore, we introduce a generality relation between chronicles: a chronicle $\mathcal{C}$ is *more general* than a chronicle $\mathcal{C}'$ (noted $\mathcal{C} \sqsubseteq \mathcal{C}'$) if there is a match between $\mathcal{C}$ and $\mathcal{C}'$ such that every event of $\mathcal{C}$ is matched to an event of $\mathcal{C}'$ and the temporal constraints of $\mathcal{C}$ contains the corresponding constraint in $\mathcal{C}'$. Thus, we can note that $\mathcal{C}_G \sqsubseteq \mathcal{C}_S$ (fig. 2) and that there is no solution chronicle more specific than $\mathcal{C}_S$ or more general than $\mathcal{C}_G$.

3 Frequency of chronicles in a sequence

The most used interest measure for patterns mining is based on frequency. The use of a generality relation makes chronicle search based on frequency easier, but only on the condition that constraints on frequency satisfy monotonicity or anti-monotonicity properties. Let $\mathcal{C}$ be more general than $\mathcal{C}'$. The threshold constraints on frequency satisfy one of these properties if the frequency of $\mathcal{C}$ is equal or greater than the frequency of $\mathcal{C}'$ in an event sequence. To ensure these properties for some frequency measure, we introduce a recognition criterion Q that specifies how instances of some chronicle are selected in a sequence and consequently how the frequency of this chronicle is computed.

Some recognition criteria have been defined in the literature. The minimal occurrences criterion [Mannila *et al.*, 1997] selects all the shortest instances of an episode (a partially ordered collection of events occurring together). The earliest distinct instances criterion, $Q_{e\&d}$ [Dousson and Duong, 1999], selects all the instances such that two instances of a same chronicle have no common events and occur as early as possible in the event sequence according to a total order on instances. Let $freq^{Q_{e\&d}}(\mathcal{C}_1, l_1)$ be the frequency of $\mathcal{C}_1$ computed on the event sequence l_1 (fig. 1). The recognition criterion $Q_{e\&d}$ recognizes the set of instances $E = \{o_1, o_3, o_6\}$ and the value of $freq^{Q_{e\&d}}(\mathcal{C}_1, l_1) = |E| = 3$.

4 Queries on event sequences

Let P and N be two sets of event sequences and T be a set of frequency thresholds. Every event sequence l has a threshold T_l . If l is an element of P (resp. N) then T_l is a minimum (resp. maximum) frequency threshold. Our goal is to extract some phenomenon in the form of chronicles. These are frequent in at least one event sequence of P and infrequent in every event sequence of N . A query has the general form:

$$Qu(P, N, T) = (\exists l \in P, freq^Q(\mathcal{C}, l) \geq T_l) \wedge (\forall l \in N, freq^Q(\mathcal{C}, l) < T_l) \quad (1)$$

The computation of frequent and maximally specific chronicles, $Fmc^Q(l, T_l)$, in an event sequence l according to a threshold T_l is performed by the levelwise algorithm of FACE [Dousson and Duong, 1999] that we have adapted to the task. The method consists in computing all the $Fmcs$ from the different event sequences separately (with FACE), memorizing them, and merging them to get the solutions of the query: this method is efficient if we consider a set of queries that use the same sets T and $(P \cup N)$. We use Mitchell's algorithm to merge these $Fmcs$. This algorithm is extensively used to compute the version spaces of symbol sequences [De Raedt and

Kramer, 2001]. It computes the bounds of the version space from a conjunction of constraints. Query (1) is rewritten in such a way as to use Mitchell's algorithm:

$$Qu(P, N, T) = \bigvee_{\mathcal{B} \in Fmc^Q(P, T)} \left(\mathcal{C} \sqsubseteq \mathcal{B} \bigwedge_{\overline{\mathcal{B}} \in Fmc^Q(N, T)} \mathcal{C} \not\sqsubseteq \overline{\mathcal{B}} \right) \quad (2)$$

where $Fmc^Q(E, T) = \bigcup_{l \in E} Fmc^Q(l, T_l)$.

For each chronicle from $Fmc^Q(P, T)$, the version space of a conjunction of constraints is computed by Mitchell's algorithm. The union of these version spaces gives the solutions set to query (2). One of the main interests of this method is that it needs to compute only once the $Fmcs$ in each sequence.

5 Conclusion

We have presented an original method that extracts temporal information in the form of patterns named chronicles. These patterns use numerical temporal constraints on events. Furthermore, they provide a way to express sequentiality and parallelism between events and they generalize temporal patterns used by Mannila and De Raedt [Mannila *et al.*, 1997; Lee and De Raedt, 2003], among others. Chronicles are extracted by querying an IDB that contains event sequences and chronicles. The user sets the minimum or maximum frequency thresholds of searched chronicles in event sequences.

Our contribution introduces the notion of recognition criterion that generalizes the specification of pattern frequency computation on temporal data. Furthermore, we use a generality relation on chronicles that enables us to adapt version space algorithms to manage numerical temporal constraints. These algorithms require the prior computation of frequent and maximal chronicles for each event sequence used in the query. This computation is performed by a data mining tool that we have adapted to the task. Our approach can compute the complete and correct set of solutions.

References

- [De Raedt and Kramer, 2001] L. De Raedt and S. Kramer. The levelwise version space algorithm and its application to molecular fragment finding. In *Proc. of IJCAI*, 2001.
- [Dousson and Duong, 1999] C. Dousson and T. Vu Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proc. of IJCAI 1999*, pages 620–626, 1999.
- [Imielinski and Mannila, 1996] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39:58–64, 1996.
- [Lee and De Raedt, 2003] S. D. Lee and L. De Raedt. *Database Support for Data Mining Applications*, volume 2682 of *LNCS*, chapter Constraint Based Mining of First Order Sequences in SeqLog. Springer-Verlag, 2003.
- [Mannila *et al.*, 1997] H. Mannila, H. Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and KD*, 1(3), 1997.
- [Yoshida *et al.*, 2000] M. Yoshida, T. Iizuka, H. Shiohara, and M. Ishiguro. Mining sequential patterns including time intervals. In *Data Mining and KD*, 2000.

Discovering Time Differential Law Equations Containing Hidden State Variables and Chaotic Dynamics*

Takashi Washio, Fuminori Adachi and Hiroshi Motoda
Institute of Scientific and Industrial Research, Osaka University,
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan.
washio@ar.sanken.osaka-u.ac.jp

Abstract

This paper proposes a novel approach to discover simultaneous time differential law equations having high plausibility to represent first principles underlying objective processes. The approach has the power to identify law equations containing hidden state variables and/or representing chaotic dynamics without using any detailed domain knowledge.

1 Introduction

A set of well known pioneering approaches of scientific law equation discovery is called BACON family [Langley *et al.*, 1987]. They try to figure out a static equation on multiple quantities over a wide state range under a given laboratory experiment. Some approaches introduced unit dimension constraints and “scale-type constraints” to limit the search space to mathematically admissible equations reflecting the first principles [Falkenhainer and Michalski, 1986], [Washio and Motoda, 1997]. Especially, the scale-type constraints have wider applicability since it does not require any unit information of quantities. Subsequently, LAGRANGE addressed the discovery of “simultaneous time differential law equations” reflecting the dynamics of objective processes under “passive observations” where none of quantities are experimentally controllable [Dzeroski and Todorovski, 1995]. Its extended version called LAGRAMGE introduced domain knowledge of the objective process to limit the search space within plausible law equations [Todorovski and Dzeroski, 1997]. Extended IPM having similar functions with LAGRAMGE further identified plausible law equations containing “hidden state variables” when the variables are known in the domain knowledge [Langley *et al.*, 2003]. PRET identified “chaotic dynamics” under similar conditions with these approaches where rich domain knowledge is available [Bradley *et al.*, 1998]. However, scientists and engineers can develop good models of the objective dynamics without using the discovery approaches in many practical cases when the detailed domain knowledge is available. Accordingly, the main applications of the discovery approaches are to identify simultaneous time

differential equations reflecting the first principles under passive observation and “little domain knowledge.”

In this paper, we propose a novel approach called SCALE-TRACK (SCALE-types and state TRACKing based discovery system) to discover a model of an objective process under the following requirements.

- (1) The model is simultaneous time differential equations representing the dynamics of an objective process.
- (2) The model is not an approximation but a plausible candidate to represent the underlying first principles.
- (3) The model is discovered from passively observed data without using domain knowledge specific to the objective process.
- (4) The model can include hidden state variables.
- (5) The model can represent chaotic dynamics.

2 Outline

2.1 Basic Problem Setting

We adopt the following “state space model” of objective dynamics and measurements without loss of generality.

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t)) + \mathbf{v}(t) \quad (\mathbf{v}(t) \sim N(0, \Sigma_v)), \text{ and} \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{w}(t) \quad (\mathbf{w}(t) \sim N(0, \Sigma_w)),\end{aligned}$$

where the first equation is called a “state equation” and the second a “measurement equation.” $\mathbf{x}$ is called a state vector, $\mathbf{f}(\mathbf{x})$ a system function, $\mathbf{v}$ a process noise vector, $\mathbf{y}$ a measurement vector, $\mathbf{C}$ a measurement matrix, $\mathbf{w}$ a measurement noise and t a time index. $\mathbf{f}(\mathbf{x})$ is not limited to linear formulae in general. $\mathbf{C}$ is represented by a linear transformation matrix, since the measurement facilities are artificial and linear in most cases. If $\mathbf{C}$ is column full rank, the values of all state variables are estimated by solving the measurement equation with $\mathbf{x}$. Otherwise, some state variables are not estimated within the measurement equation, and these variables are called “hidden state variables.” In the scientific law equation discovery, $\mathbf{f}(\mathbf{x})$ is initially unknown, and even $\mathbf{x}$ is not known correctly. Only a state subvector $\mathbf{x}'(\subseteq \mathbf{x})$ and a submatrix $\mathbf{C}'(\subseteq \mathbf{C})$ are initially known. To derive $\mathbf{C}$ from $\mathbf{C}'$, SCALETRACK must identify the dimension of $\mathbf{x}$ at first. Then, it searches plausible candidates of $\mathbf{f}(\mathbf{x})$ from the measurement time series data.

*The support of JSPS Grant-in-aid for Scientific Research, No.16300045 is acknowledged.

2.2 Entire Approach

The entire approach of SCALETRACK is outlined in Figure 1. Given a set of measurement time series data, the dimension of x is identified through a statistical analysis called “correlation dimension analysis” [Berge *et al.*, 1984]. For each element of y , its time trajectory is mapped to a phase space constructed by time lagged values of the element, and the degree of freedom, *i.e.*, the dimension of x , embedded in the time trajectory is estimated by computing the sparseness of the trajectory in the space.

Once the dimension is known, all possible combinations of scale-types of the elements in x are enumerated based on scale-type constraints, the known measurement submatrix C' and the scale-types of the elements in y . The representative scale-types of quantities are ratio scale and interval scale. The examples of the ratio scale quantities are physical mass and absolute temperature where each has an absolute origin, while the examples of the interval scale quantities are temperature in Celsius and sound pitch where their origins are not absolute and arbitrary changed by human’s definitions. Due to these natures, the quantitative relations among the quantities are strongly constrained [Luce, 1959], and these constraints are used to determine the scale types of the elements in x from y . After every combination of the scale types in x is derived, the candidate formulae of a state equation are generated for each combination based on “Extended Product Theorem” [Washio and Motoda, 1997] limiting the admissible formulae of the equation based on the scale-type constraints.

Subsequently, through a set of state tracking simulations called “SIS/RMC filter” on the given measurement time series data, the parameter values and the states in every candidate state equation are estimated [Doucet *et al.*, 2000]. This state tracking has many advantages comparing with the other nonlinear state tracking approaches such as the conventional Extended Kalman Filter [Haykin, 2001] and the qualitative reasoning based PRET [Bradley *et al.*, 1998]. The former using the linearization of the state equations does not work well when the equations include some singular points and/or some state regions having strong sensitivity to the tracking error. The latter faces a combinatorial explosion of qualitative states when the dimension and/or the complexity of the state space structure are high. In contrast, SIS/RMC filter does not require any approximation to be spoiled by the singularity and the strong nonlinearity, and does not face the combinatorial explosion of the states to be considered, because it tracks the state probability distributions by using its direct and sequential Monte Carlo integration within Bayesian framework. In our approach, the estimated parameter values are rounded off to integers when the values are close enough to the integers within the expected estimation errors, since the parameters tend to be integers in many physical processes. Finally, some state equations providing highly accurate tracking in terms of “Mean Square Error (MSE)” are selected as the plausible candidates of first principle based and dynamic state space models of the objective process.

2.3 Implementation

The evaluation of candidate state equations by the SIS/RMC filter is the most time consuming step. Any search can not

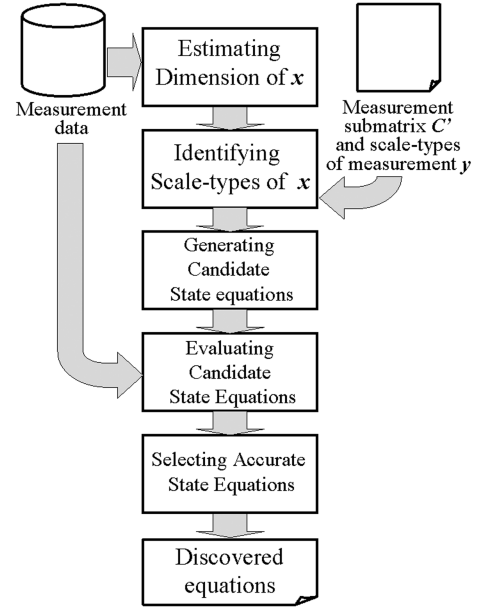


Figure 1: Outline of Entire Approach.

Table 1: Basic Performance.

case	ν ($\sigma_w = 1.0\%$)	ct (h)	σ_w (%)				
			0.1	0.5	1.0	2.0	5.0
RR	2.21	1.5	+	±	±	±	—
RRH	2.21	5.5	±	±	—	—	—
RI	2.19	4.0	+	±	±	±	—
RIH	2.19	5.5	+	±	—	—	—

be skipped, since the search space is nonmonotonic. We experienced that one run of stand alone SCALETRACK to discover a simple state equation took more than a month even if we used an efficient algorithm. Accordingly, the current SCALETRACK introduced a simple grid computing framework using a PC cluster consisting of a control server and 10 personal computers where each has an Athlon XP 1900+ (1.6 GHz) CPU and 2GB RAM. The server computes the first three steps and then allocates the task to evaluate 10% of candidate state equations to each computer. Because this task is mutually independent, and occupies the most of computation in SCALETRACK, this implementation accelerates the run speed almost 10 times.

3 Result

3.1 Basic Performance Evaluation

The evaluation is made in terms of scale-types of state variables, hidden state variables and measurement noise levels by using the following two dimensional artificial formulae.

$$\left. \begin{aligned} \dot{x}_1(t) &= x_1(t)x_2(t) \\ \dot{x}_2(t) &= -0.5x_2(t) \end{aligned} \right\} RR,$$

where $y_1 = x_1$ and $y_2 = x_2$ are ratio scale.

$$\left. \begin{aligned} \dot{x}_1(t) &= 0.4x_1(t)(x_2(t) + 0.2) \\ \dot{x}_2(t) &= -0.1(x_2(t) + 0.6) \end{aligned} \right\} RI,$$

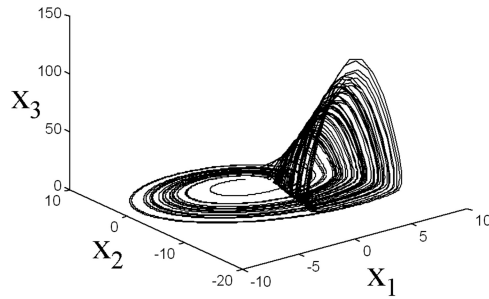


Figure 2: An Attractor of Altered Rossler Chaos.

where $y_1 = x_1$ is ratio scale and $y_2 = x_2$ interval scale. Table 1 shows the evaluation result. All state variables are observed in RR and RI. Whereas, the measurement variable y_2 is not available in RRH and RIH respectively, and hence a hidden state variable exists in these cases. The correlation dimension analysis properly estimated the dimension ν of state vectors as nearly 2 in each case. The computation times ct required for RRH, RI and RIH were longer than that of RR, because the variety of admissible formulae containing interval scale variables is larger than that of ratio scale variables. The result in that the formula having the correct shape is top ranked by the accuracy is marked by +. If the correct formula is derived within the top five solutions, it is marked by $\pm$, otherwise it is marked by $-$. The table shows that almost $\sigma_w = 2.0\%$ relative noise is acceptable for no hidden state cases, while noise less than 1.0% is required for hidden state cases. Since $0.5-2.0\%$ noise is the most widely seen in many applications, the performance of SCALETRACK is practical for no hidden state cases and some hidden state cases.

3.2 Discovery of Chaos

The state equation to be discovered is the following Altered Rossler Chaos equation.

$$\begin{aligned}\dot{x}_1 &= -x_2 - x_3, & \dot{x}_2 &= x_1 + 0.36x_2, \text{ and} \\ \dot{x}_3 &= 0.01(x_1 - 4.5)(x_1 + 1000x_3 - 4.5).\end{aligned}$$

This has an attractor in a (x_1, x_2, x_3) -phase space as depicted in Figure 2. All state variables are interval scale, and can be measured through the corresponding interval scale measurement variables respectively. $\nu = 3.33$ was obtained in the correlation dimension analysis, and hence the state equations consisting of three state variables were searched. The required computation time was 15.0 hours, and the following most accurate state equation was resulted. This formula has an identical shape with the original except some discrepancies of coefficients. This indicates the high ability of SCALETRACK to discover the Chaotic dynamics reflecting the underlying first principles.

$$\begin{aligned}\dot{x}_1 &= -x_2 - x_3, & \dot{x}_2 &= x_1 + 0.33x_2, \text{ and} \\ \dot{x}_3 &= 0.064(x_1 - 6.34)(x_1 + 1002x_3 - 4.75).\end{aligned}$$

4 Conclusion

SCALETRACK achieved three advantages which have not been addressed in any past work of mathematics, physics and

engineering not limited to scientific discovery. The first is the discovery of simultaneous time differential equations having plausibility to represent first principles. The second is the discovery of hidden state variables. The third is the discovery of chaotic dynamics. These discoveries are done without using detailed domain knowledge. These advantages are essentially important in many scientific and engineering fields due to the wide existence of such dynamics in nature.

References

- [Berge *et al.*, 1984] Pierre Berge, Yves Pomeau, and Christian Vidal. *Order in Chaos - For understanding turbulent flow*. Hermann, Paris, France, 1984.
- [Bradley *et al.*, 1998] Elizabeth A. Bradley, Agnes A. O'Gallagher, and Janet E. Rogers. Global solutions for nonlinear systems using qualitative reasoning. *Annals of Mathematics and Artificial Intelligence*, 23:211–228, 1998.
- [Doucet *et al.*, 2000] Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing*, 10:197–208, 2000.
- [Dzeroski and Todorovski, 1995] Saso Dzeroski and Ljupco Todorovski. Discovering dynamics: from inductive logic programing to machine discovery. *Journal of Intelligent Information Systems*, 4:89–108, 1995.
- [Falkenhainer and Michalski, 1986] Brian C. Falkenhainer and Ryszard S. Michalski. Integrating quantitative and qualitative discovery: The abacus system. *Machine Learning*, 1:367–401, 1986.
- [Haykin, 2001] Simon S. Haykin. *Kalman Filtering and Neural Networks*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2001.
- [Langley *et al.*, 1987] Pat W. Langley, Herbert A. Simon, Gary L. Bradshaw, and Jan M. Zytkow. *Scientific Discovery; Computational Explorations of the Creative Process*. MIT Press, Cambridge, Massachusetts, 1987.
- [Langley *et al.*, 2003] Pat Langley, Dileep George, Stephen Bay, and Kazumi Saito. Robust induction of process models from time-series data. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 432–439, Menlo Park, California, August 2003. The AAAI Press.
- [Luce, 1959] Duncan R. Luce. On the possible psychological laws. *Psychological Review*, 66(2):81–95, 1959.
- [Todorovski and Dzeroski, 1997] Ljupco Todorovski and Saso Dzeroski. Declarative bias in equation discovery. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 376–384, San Mateo, California, July 1997. Morgan Kaufmann.
- [Washio and Motoda, 1997] Takashi Washio and Hiroshi Motoda. Discovering admissible models of complex systems based on scale-types and identity constraints. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 810–817, Nagoya, Japan, August 1997.

Learning Global Models Based on Distributed Data Abstractions

Xiaofeng Zhang and William K. Cheung

Department of Computer Science

Hong Kong Baptist University

Kowloon Tong, Hong Kong

{xfzhang,william}@comp.hkbu.edu.hk

Abstract

Due to the increasing demand of massive and distributed data analysis, achieving highly accurate global data analysis results with local data privacy preserved becomes an increasingly important research issue. In this paper, we propose to adopt a model-based method (Gaussian mixture model) for local data abstraction and aggregate the local model parameters for learning global models. To support global model learning based on solely local GMM parameters instead of virtual data generated from the aggregated local model, a novel EM-like algorithm is derived. Experiments have been performed using synthetic datasets and the proposed method was demonstrated to be able to achieve the global model accuracy comparable to that of using the data regeneration approach at a much lower computational cost.

1 Introduction

Most machine learning and data mining algorithms work with a rather basic assumption that all the data can first be pooled together in a centralized data repository. Recently, there exist a growing number of cases that the data have to be physically distributed due to some practical constraints, such as privacy concerns. The need to tackle them calls for the recent research interest in distributed data mining.

A common methodology for distributed data mining is to first perform local data analysis and then combine the results to form a global one [Kargupta *et al.*, 2000]. One major limitation is that local analysis may result in loss of information for global pattern discovery. In [Zhang *et al.*, 2004], a model parameter exchange approach was proposed to improve the global model accuracy, which however is not a generic methodology which can easily be generalized to different types of global models. An alternative is to adopt a flexible enough model for local data abstraction and have the subsequent analysis based on the local model parameters, which is what we adopt in this paper.

As inspired by [Merugu and Ghosh, 2003], Gaussian mixture model (GMM) [Bishop, 1995] is chosen for the local data abstraction. Via different parameter settings, GMM can interpolate a set of data with more detailed information as well

as a rough ensemble illustration. Thus, by sharing only local model parameters for global analysis, the degree of data privacy preservation and the bandwidth requirement in a distributed environment can readily be controlled. In this paper, an EM-like algorithm is proposed for learning a global GMM directly from the aggregated version of the local abstractions.

2 Problem Formulation

2.1 Local Data Abstraction

Let $t_i \in \mathbb{R}^d$ denote an observed data instance, θ_l denote the parameters of the l^{th} local model (GMM) as the abstraction of the l^{th} source. The probability density function of the l^{th} local model $p_{local}(t_i|\theta_l)$ with K_l components is given as,

$$p_{local}(t_i|\theta_l) = \sum_{j=1}^{K_l} \alpha_{jl} p_j(t_i|\theta_{lj}) \quad \sum_{j=1}^{K_l} \alpha_{jl} = 1$$
$$p_j(t_i|\theta_{lj}) = (2\pi)^{-\frac{d}{2}} |\Sigma_{lj}|^{-\frac{1}{2}} \exp\{-\frac{1}{2}(t_i - \mu_{lj})^T \Sigma_{lj}^{-1} (t_i - \mu_{lj})\}.$$

L sets of local GMMs' parameters $\{\theta_1, \theta_2, \dots, \theta_L\}$ obtained by performing local model-based density estimation (e.g., the EM algorithm) are sent to a global server to learn a global data model.

2.2 Learning Global Model Parameters

We start the derivation of our learning algorithm by considering first the conventional setting of the Expectation Maximization (EM) algorithm. Let R_{ik} denote the estimated indicator for the k^{th} component of the global model generating the i^{th} data instance.

Assume R_{ik} is now a new indicator for the l^{th} local component¹ with its underlying data to be generated by the k^{th} global component and $\bar{R}_{ik}$ denotes the average estimate of the probability that data instances from the l^{th} local component are generated by the k^{th} global component. By approximating R_{ik} as

$$R_{ik} \approx \bar{R}_{ik} = \frac{\sum_{i \in l^{th} source} R_{ik}}{N_l} \quad (1)$$

where N_l denotes the number of data from the l^{th} source, the

¹Note that we abuse the index “ l ” to refer to one local component of the aggregated local model.

new M-steps become

$$\begin{aligned}\mu_k &= \frac{\sum_{l=1}^L R_{lk} \mu_l}{\sum_{l=1}^L R_{lk}} \quad \alpha_k = \frac{1}{L} \sum_{l=1}^L R_{lk} \\ \Sigma_k &= \frac{\sum_{l=1}^L R_{lk} (\Sigma_l + \mu_l \mu_l^T)}{\sum_{l=1}^L R_{lk}} - \mu_k \mu_k^T.\end{aligned}\quad (2)$$

KL-divergence is adopted to compute R_{lk} (E-step), resulting in

$$R_{lk} = \frac{\exp\{-D(p_{global}(t|\phi_k)||p_{local}(t|\theta_l))\}}{\sum_{i=1}^M \exp\{-D(p_{global}(t|\phi_i)||p_{local}(t|\theta_l))\}} \quad (3)$$

where $D(P||Q)$ denotes the KL-divergence between the two probabilistic models P and Q , which can be derived as

$$\ln \frac{|\Sigma_l|^{\frac{1}{2}}}{|\Sigma_k|^{\frac{1}{2}}} + \frac{1}{2}(\text{trace}(\Sigma_l^{-1} \Sigma_k) + (\mu_k - \mu_l)^T \Sigma_l^{-1} (\mu_l - \mu_k) - 1).$$

3 Experiments

Experiments have been performed using synthetic data sets for performance evaluation. Four data sets (as ground truth) were generated using known mixture models with 4, 5, 6 and 7 components, respectively. The data instances were then randomly partitioned as four sources and local GMMs were learned via the conventional EM algorithm for each source. Two global models were trained, one using the proposed method and one using the conventional data regeneration approach. As shown in the Table 1, the proposed method can achieve a speedup factor ranging from 1.5 to 257 with highly comparable model accuracy.

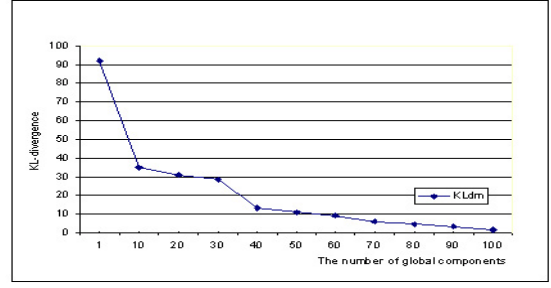
Table 1: Comparing the global models learned using the proposed method and the data regeneration method (g x denotes the true data model with x components).

		time/speedup	log-likelihood/diff.	mean/diff.	KL
g4	2:	0.09s/1.5	-1470.2/+2	3.7/-0.03	0.43
	3:	0.22s/1.8	-1169.4/-70.9	1.0/-0.03	0.12
g5	2:	1.33s/33	-1850.6/-17	3.2/-0.02	0.21
	3:	8.11s/162	-1898.5/-67	4.1/+0.04	1.31
	4:	10.3s/257	-1788.4/+1	3.3/+0.02	3.66

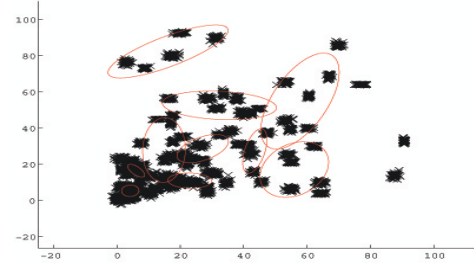
The proposed approach has also been applied to a more sizeable dataset generated using a GMM with 100 components. Global models with different number of components were learned and their corresponding KL-divergence values when compared with the true model are shown in Figure 1 (a). The divergence value decreased rapidly when the number of components increased from 1 to 10 and then started to saturate. One could determine the optimal number of global components by setting a threshold on the KL-divergence value as cut-off. Figure 1 (b) shows the result with 10 global components identified.

4 Conclusion and Future Plan

In this paper, we proposed a novel EM-like algorithm for learning a global model directly from the aggregated local model parameters with promising experimental results on synthetic datasets. Currently, we are extending the obtained results to learn more sophisticated global models. Furthermore, we are investigating how active negotiation between



(a) The KL-divergence of the model learned using the proposed method with different numbers of components of the global model.



(b) GMM with 10 components learned based on 100 local Gaussian components for the aggregated local model.

Figure 1: Experimental results with a sizeable dataset.

the global mediator and the local data sources for compromising individual local data sources' privacy levels can be adopted to improve the global model accuracy.

Acknowledgement

This work has been partially supported by RGC Central Allocation Group Research Grant (HKBU 2/03/C).

References

- [Bishop, 1995] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [Kargupta et al., 2000] H. Kargupta, B. Park, D. Hersherberger, and E. Johnson. Collective Data Mining: A New Perspective Towards Distributed Data Mining. In *Advances in Distributed and Parallel Knowledge Discovery*. MIT/AAAI Press, 2000.
- [Merugu and Ghosh, 2003] S. Merugu and J. Ghosh. Privacy-preserving Distributed Clustering using Generative Models. In *Proceedings of IEEE International Conference on Data Mining*, 2003.
- [Zhang et al., 2004] Xiaofeng Zhang, C.M. Lam, and William K. Cheung. Mining local data sources for learning global cluster models via local model exchange. *The IEEE Intelligent Informatics Bulletin*, 4(2), 2004.

POSTER PAPERS

MULTI-AGENT SYSTEMS

Language Learning in Multi-Agent Systems

Martin Allen

Computer Science Department
University of Massachusetts
Amherst, MA 01003, USA
mwallen@cs.umass.edu

Claudia V. Goldman

Caesarea Rothschild Institute
University of Haifa
Mount Carmel, Haifa 31905, Israel
clag@cri.haifa.ac.il

Shlomo Zilberstein

Computer Science Department
University of Massachusetts
Amherst, MA 01003, USA
shlomo@cs.umass.edu

Abstract

We present the problem of learning to communicate in decentralized and stochastic environments, analyzing it formally in a decision-theoretic context and illustrating the concept experimentally. Our approach allows agents to converge upon coordinated communication and action over time.

1 Introduction

Learning to communicate in multi-agent systems is an emerging challenge AI research. Autonomous systems, developed separately, interact more and more often in contexts like distributed computing, information gathering over the internet, and wide-spread networks of machines using distinct protocols. As a result, we foresee the need for autonomous systems that can learn to communicate with one another in order to achieve cooperative goals. We make some first steps towards solving the attendant problems.

Coordination among agents acting in the same environment while sharing resources has been studied extensively, particularly by the multi-agent systems community. While such coordination may involve communication, typically there is no deliberation about the value of communication, resulting in systems with no communication or ones allowing free communication of well-understood messages. In contrast, we study decentralized systems that require agents to adapt their communication language when new situations arise or when mis-coordination occurs.

2 The Decentralized Learning Framework

We study the problem in the context of *decentralized Markov Decision Processes* [Bernstein et al., 2002] with communication (Dec-MDP-Com). Such a process is a multi-agent extension of a common MDP in which each agent α_i observes only its own local portion of the state-space, and can attempt to communicate with others using the set of messages Σ_i . Decentralization makes Dec-MDPs, with communication or not, significantly harder to solve than regular MDPs; for their complexity properties, see [Goldman and Zilberstein, 2004].

If agents in a system share the same language, optimal linguistic action is a matter of deciding *what* and *when* to communicate, given its cost relative to the projected benefit of

sharing information. However, where agents utilize different sets of messages, and do not fully understand one another, message-passing alone is not enough. Rather, agents need to learn how to *respond* to the messages that are passed between them—in a sense, learning what those messages *mean*.

Definition 1 (Translation). Let Σ and Σ' be sets of messages. A *translation*, τ , between Σ and Σ' is a probability function over message-pairs: for any messages σ , σ' , $\tau(\sigma, \sigma')$ is the probability that σ and σ' have the same meaning. $\tau_{\Sigma, \Sigma'}^+$ is the set of all translations between Σ and Σ' .

Agents may need to consider multiple possible translations between messages; that is, agents possess beliefs as to which translation is correct given their present situation.

Definition 2 (Belief-state). Let agents α_1 , α_2 use sets of messages Σ_1 , Σ_2 . A *belief-state* for α_i is a probability-function β_i over translation-set $\tau_{\Sigma_i, \Sigma_j}^+$ ($i \neq j$). For translation $\tau \in \tau_{\Sigma_i, \Sigma_j}^+$, $\beta_i(\tau)$ is the probability that τ is correct.

Updating beliefs about translations is thus an important part of the overall process of learning to communicate. Agents act based upon local observations, messages received, and current beliefs about how to translate those messages. Their actions lead to new observations, causing them to update beliefs and translations. The procedure governing these updates comprises the agent's *language-model*, a function from actions, messages, and observations to distributions over translations. Such models may be highly complex, or difficult to compute, especially where languages are complicated, or the environment is only partially observable. Here we concentrate upon special—but interesting—cases for which generating these probabilities is much more straightforward.

3 Formal Properties of the Problem

Our main formal results isolate conditions under which Dec-MDP-Coms reduce to simpler problems, and present a protocol for learning to communicate in such reduced problems.

Reduction to MMDPs

Boutilier [1999] defines *multiagent MDPs* (MMDPs), consisting of a set of agents operating in a fully- and commonly-observed environment; transitions between states in that environment arise from *joint actions* of all agents, and a common reward is shared by the system as a whole. While we can

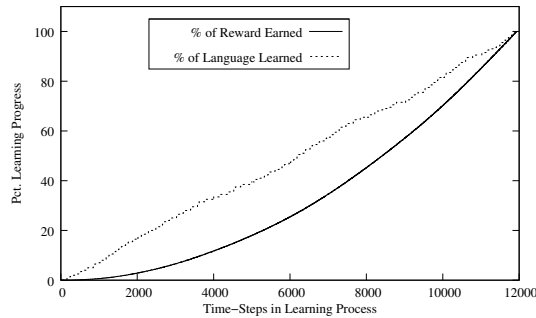


Figure 1: Reward accumulated as language is learned.

calculate an optimal joint policy for such a process offline, this is not the same thing as *implementing* it. Unless agents can coordinate their actions, there is no guarantee of a jointly optimal policy, since communication is not allowed, or is unreliable. Boutilier thus defines *coordination problems*, which arise when agents may each take an individual action that is potentially optimal, but which combine in sub-optimal fashion. We show that certain, putatively more complex, Dec-MDP-Coms in fact reduce to MMDPs for which such problems do not arise. This is notable, as Dec-MDPs are generally intractable, while MMDPs can be solved efficiently.

Definition 3 (Fully-describable). A Dec-MDP-Com is *fully-describable* if and only if each agent α_i possesses a language Σ_i that is sufficient to communicate both: (a) any observation it makes, and (b) any action it takes.

Definition 4 (Freely-describable). A Dec-MDP-Com is *freely-describable* if and only if for any agent α_i and message $\sigma \in \Sigma_i$, the cost of communicating that message is 0.

Claim 1. A Dec-MDP-Com is equivalent to an MMDP without coordination problems if (a) it is both fully- and freely-describable; and (b) agents share a common language. $\square$

Suitability and Convergence

For any freely- and fully-describable Dec-MDP-Com, agents can calculate an optimal joint policy, under the working assumption that all agents share a common language and that all relevant information is shared. Where agents must in fact learn to communicate, however, implementation of such policies requires cooperation from the environment, so that agents can update translations appropriately over time. The full definition of a *suitable* Dec-MDP-Com cannot be included here; we simply note that in such problems, the probability that each agent assigns to the actual prior observations and actions of others *following* some state-transition is strictly greater than that of the observations and actions considered most likely *before* that transition (unless those entries were actually correct). Suitable Dec-MDP-Coms provide enough information to ensure that others' actual actions and observations are more likely than mistaken ones.

We extend work of Goldman *et al.* [2004] (where agents communicate states but not actions), to give an *elementary action protocol*. Using such a protocol for action and belief-update, agents move towards optimality, based upon the observed consequences of action in a suitable problem-domain.

Claim 2. Given an infinite time-horizon, agents acting according to the elementary action protocol in a suitable Dec-MDP-Com will eventually converge upon a joint policy that is optimal for the states they encounter from then on. $\square$

4 Empirical Results and Conclusions

To explore the viability of our approach, we implemented our language-learning protocol for a reasonably complex (but still suitable) Dec-MDP-Com, involving two agents in joint control of a set of pumps and flow-valves in a factory setting.

Our results show the elementary protocol converging on optimal policies in a wide range of problem-instances. Figure 1 gives an example, for a problem-instance featuring 100 vocabulary-items for each agent, showing the percentage of total accumulated reward, and total shared vocabulary, at each time-step in the process of learning and acting. As can be seen, the learning process (top, dotted line) proceeds quite steadily. Reward-accumulation, on the other hand, grows with time before finally stabilizing. Initially, language learning outpaces reward gain given that knowledge, as agents still find many of the other's actions and observations hard to determine. As time goes on, the rate of accumulated reward narrows this gap considerably; agents now know much of what they need to communicate, and spend more time accumulating reward in familiar circumstances, without necessarily learning anything new about the language of others.

These experimental results conform with intuition, showing that while a small amount of language learning does little to help agents in choosing their actions, they are capable of very nearly optimal action even in the presence of an understanding that is still less than perfect. This opens the door for further study into approximation in these contexts. We continue to investigate and compare other approaches to the problem, including analysis of the differences between possible optimal offline techniques and online learning methods.

Acknowledgments

This work was supported in part by the National Science Foundation under grant IIS-0219606 and by the Air Force Office of Scientific Research under grant F49620-03-1-0090.

References

- [Bernstein *et al.*, 2002] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27:819–840, 2002.
- [Boutilier, 1999] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *Procs. of IJCAI-99*, pages 478–485, Stockholm, Sweden, 1999.
- [Goldman and Zilberstein, 2004] C. V. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *Journal of Artificial Intelligence Research*, 22:143–174, 2004.
- [Goldman *et al.*, 2004] C. V. Goldman, M. Allen, and S. Zilberstein. Decentralized language learning through acting. In *Procs. of AAMAS-04*, pages 1006–1013, New York City, NY, 2004.

On the Evolution of Memory Size in the Minority Game (extended abstract)

Ricardo M. Araújo and Luis C. Lamb

Institute of Informatics, Federal University of Rio Grande do Sul

91501-970, Porto Alegre, Brazil

email: rmaraujo@inf.ufrgs.br, LuisLamb@acm.org

1 Introduction

This work presents a study on the effects of different memory sizes in the Minority Game (MG) market model [Zhang, 1998]. We analyse the effects on an agent's performance when this agent is endowed with a different memory size with respect to other agents in the game. Our aim is to identify in which situations a large or small memory might be advantageous in the game. From the obtained results we argue that there exist convergence to and evolutionary stability around certain memory sizes and we consider an evolutionary setup which confirms our hypothesis.

The MG is defined as an odd number of agents (N) which must choose at each turn of the game whether they will be in one of two possible groups. After all agents have made their choices, agents in the minority group are rewarded. In order to make a decision, agents use an inductive learning algorithm in order to try to exploit patterns from previous outcomes. The number of past turns that agents consider to decide defines their *memory size* (M).

One of the main properties in a MG is its *efficiency*, which is evaluated by the statistical variance (σ^2) of the number of agents in a group [Moro, 2004]. Efficiency is known to be a function of a control parameter $\alpha = \frac{2^M}{N}$ [Manuca *et al.*, 2000]. For N fixed, M becomes the main control parameter of the game. Figure 1 shows σ^2 for a range of memory sizes. Three regions are observed: (i) for small M variance is very high, above the expected for the random case game (which we will call inefficient region); (ii) for high values of M , variance is exactly the one expected for the random case game (random case region); (iii) for intermediate M the system presents small variances (efficient region).

2 Memory Size and Performance

Firstly, we tackle the *design* aspects of agents by asking *what is the best memory size for an agent immersed in a MG where agents have a different memory size*. We do so by modifying the typical MG setup, defining two basic elements: (i) the *environment*, a classical MG with an even number (N) of agents; (ii) a *target-agent* endowed with a possibly different memory size (m) from that in the environment. We have simulated different target-agents within environments in all 3 efficiency regions, taking N as a large value, so that the target-agent plays no role in the determination of the game's

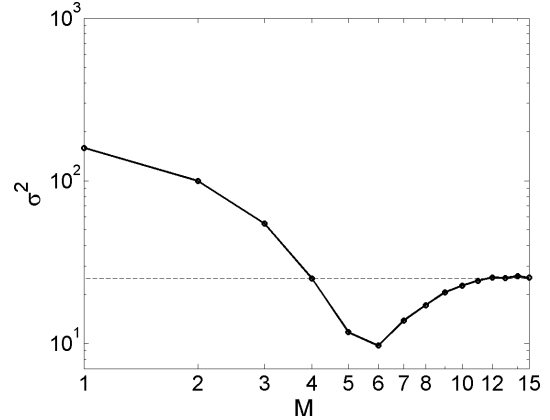


Figure 1: Variance versus memory size

efficiency. We are interested in the target-agent's success rate (number of correct decisions over total number of rounds).

Figure 2 shows the success rate for different values of m for the environment in the inefficient region ($M = 3$). It is shown that the target-agent with larger memories performs better than average. This agrees with the common belief that having access to more information is beneficial. Figure 3 shows that this is not always the case. For $M = 6$ the environment is now in the efficient region and increases in the target-agent's memory size does not translate into better performance. Actually, having access to more information is shown to be *harmful*.

Finally, we have simulated our target-agent in the random case region. We have observed that the target-agent showed no gains or losses of performance throughout all values of m . This is mainly due to the fact that if agents are acting as if they were choosing randomly, as is the case in this region, then there will be no pattern in the outcome time series to be exploited.

Figure 4 depicts the gain ($\frac{f_{target}}{f_{avg}}$) of a target-agent with one more bit of memory than the environment for a range of M (i.e. $m = M + 1$ for all M). We notice that gains are reduced as the system approaches the efficient region and indeed become smaller than unit for every M in the efficient region, reapproaching unit when closer to the random case region. It is only in the inefficient region that gains become larger than unit and agents with more memory are able to

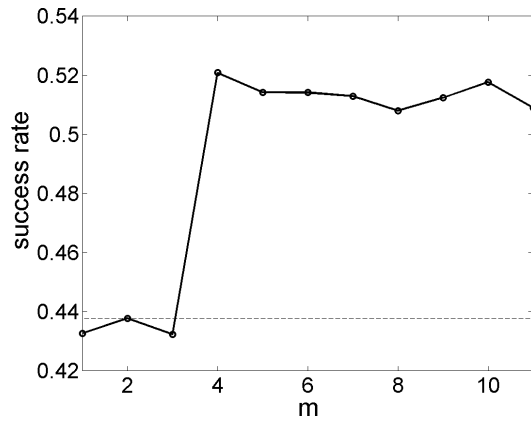


Figure 2: Target-agent's success rate in the inefficient region. Dashed line is the average success rate of the environment.

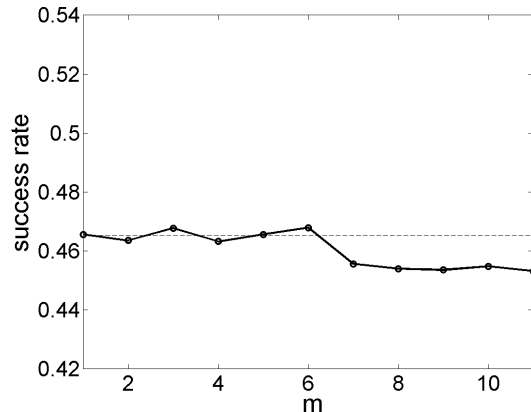


Figure 3: Target-agent's success rate in the efficient region.

exploit the environment. Interestingly, the same experiment repeated having $m = M - 1$ shows that *smaller* memories makes no difference to the target-agent's performance, it becomes the same as the environment's average.

3 Evolution of memory size

Now we apply our results in an evolutionary setup, in a MG where mutant agents - endowed with slightly different memory sizes - may appear and replace badly performing agents. Let the system be initially in an inefficient phase. Our experiments have shown that an agent with a larger memory would be able to exploit the environment in this region while smaller memories provide no benefits. Thus, agents with larger memories would be able to survive and average memory size would increase until the point where further increases lead to no gains. This happens when the system enters the efficient region. M_0 denotes the memory size in which the system first becomes efficient. The system cannot move beyond M_0 , as larger memories would cause harm to the mutant's performance. In addition, the system cannot move towards lower memory sizes: whenever a sufficient number of agents convert to smaller memories, the system would again be in an inefficient region and the push towards larger memories would reappear.

A similar reasoning might be applied to a MG starting at any point M in the efficient region, above or equal to M_0 .

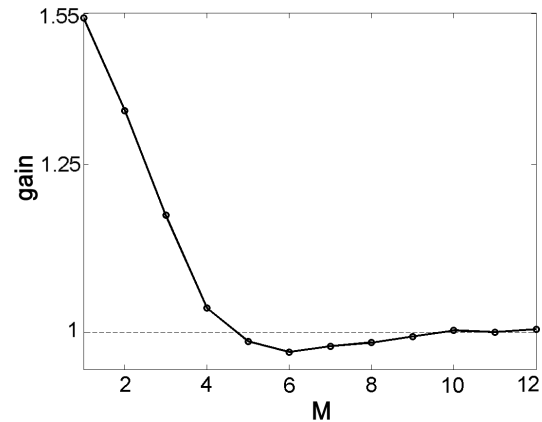


Figure 4: Gain of a target-agent with $m=M+1$.

A mutant agent with larger memory would be penalized, but one with smaller memory would perform just as well and we could expect that memory sizes would end up being distributed throughout values below M . However, this would result in a reduction of the average memory size, causing the system to move towards the inefficient region and making larger memories harmful, thus creating a new upper limit in memory size. The process continues until M_0 is reached.

We may say that M_0 is not only *evolutionary stable*, as no mutant agent can do better than the agents already in the environment at this point, but it is also an *attractor* of memory sizes in the efficient and inefficient regions. By starting in any point in these regions, the system is expected to converge to M_0 . This reasoning was indeed confirmed by simulations. However we did not have a convergence towards $M_0 = 4$ as it would be expected by analysing the traditional MG (see Fig. 1). We have observed an initial "arms race" of memory sizes up to $M = 4$ and then a fall towards a final equilibrium at $M = 1$. This may be explained if we consider that two adaptations happen, adaptation of memory size and adaptation of strategies, for when an agent modifies her memory size she also modifies her strategies. Since the strategies space is much larger than the memory sizes space, adaptation of the latter is faster. Thus, at the beginning of the simulation we indeed have $M_0 = 4$ and, as strategies adapt, a different dynamics emerge [Araújo and Lamb, 2004], with a different value of M_0 .

Acknowledgments: This work was partly supported by CAPES, CNPq and FAPERGS, Brazil.

References

- [Araújo and Lamb, 2004] R. M. Araújo and L. C. Lamb. Towards understanding the role of learning models in the dynamics of the minority game. In *Proc. of 16th IEEE ICTAI 2004*, pages 727–731, 2004.
- [Manuca et al., 2000] R. Manuca, Y. Li, R. Riolo, and R. Savit. The structure of adaptive competition in minority games. *Physica A*, 282:559, 2000.
- [Moro, 2004] E. Moro. The minority game: an introductory guide. In E. Korutcheva and R. Cuerno, editors, *Advances in Condensed Matter and Statistical Physics*. Nova Science Publishers, 2004.
- [Zhang, 1998] Y.-C. Zhang. Modeling market mechanism with evolutionary games. *Europhysics News*, March/April 1998.

Achieving Allocatively-Efficient and Strongly Budget-Balanced Mechanisms in the Network Flow Domain for Bounded-Rational Agents

Yoram Bachrach and Jeffrey S. Rosenschein

{yori, jeff}@cs.huji.ac.il

School of Engineering and Computer Science

Hebrew University

Jerusalem, Israel

Abstract

Vickrey-Clarke-Groves (VCG) mechanisms are a framework for finding a solution to a distributed optimization problem in systems of self-interested agents. VCG mechanisms have received wide attention in the AI community because they are efficient and strategy-proof; a special case of the Groves family of mechanisms, VCG mechanisms are the *only* direct-revelation mechanisms that are allocatively efficient and strategy-proof. Unfortunately, they are only weakly budget-balanced.

We consider self-interested agents in a network flow domain, and show that in this domain, it is possible to design a mechanism that is both allocatively-efficient and almost completely budget-balanced. This is done by choosing a mechanism that is not *strategy-proof* but rather *strategy-resistant*. Instead of using the VCG mechanism, we propose a mechanism in which finding a beneficial manipulation is an NP-complete problem, and the payments from the agents to the mechanism may be minimized as much as desired.

1 Mechanisms for Network Flow

Mechanisms face the problem of finding a system-wide solution to an optimization problem based on private information given by self-interested agents. A well-known solution to such problems in the case of quasi-linear preferences is that of Groves mechanisms. A special case of the Groves family of mechanisms are VCG mechanisms, which are budget-balanced, allocatively-efficient, and strategy-proof.

A significant disadvantage of VCG mechanisms is that they are only *weakly* budget-balanced. We would in principle prefer a *strongly* budget-balanced mechanism, where the total sum of payments to the mechanism is zero: $\sum_i t_i(\theta) = 0$. Impossibility results ([Green and Laffont, 1977] and [Hurwicz, 1975]) show that in a quasi-linear environment, it is impossible to achieve a mechanism that is strategy-proof, allocatively-efficient, and strongly budget-balanced.

We address this by slightly relaxing the strategy-proof requirement, replacing it with *strategy-resistance*. A mechanism is strategy-proof if the dominant strategy of each agent is to reveal its true type to the mechanism. Strategy-resistance

only requires that even if an agent is given the reported types of the other agents, it has a computationally intractable problem to solve if it wishes to find a beneficial manipulation (i.e., report a false type to the mechanism to gain higher utility).

In this work, we consider the domain of network flows. In that domain, the edges of a network flow belong to several self-interested agents. Each agent reports its edges to the mechanism. The mechanism is then required to choose a flow from the source vertex to the target vertex. Agents gain utility from flow units on their edges. Agents can lie, and declare a subset of the edges they really own. This manipulation may change the flow that the mechanism chooses, and agents may try to take advantage of this. However, we show that it is possible to use the fact that agents have computational limitations and are not unboundedly rational, so as to construct mechanisms with beneficial properties.

1.1 Self-Interested Network Flows

We explore the problem of designing a mechanism for bounded-rational agents in a distributed flow problem. We demonstrate that for this domain, we can find a mechanism that is allocatively-efficient, ϵ -budget-balanced, and strategy-resistant. This means that if we assume the agents are bounded-rational and would not try to manipulate the mechanism if such manipulation is an NP-complete problem, they would all truthfully report their preferences. Once the mechanism gets their true preferences, it chooses the outcome that maximizes total utility of the agents. To achieve this truthfulness, the mechanism requires payments; however, the total sum of these payments can be minimized as much as required.

We now present the self-interested layered-graph network flow problem. Consider a flow network on a layered graph. We have a graph $G = \langle V, E \rangle$, with source vertex s and target vertex t . The vertices of the graph are partitioned into $n + 1$ layers, $L_0 = \{s\}, L_1, \dots, L_n = \{t\}$. The edges only run between consecutive layers. We have a capacity function $c : E \rightarrow \mathbb{R}$ which is the maximal flow allowed on the edges. We also have a set I of agents. Each agent controls a subset $E_i \subset E$ of the graph's edges. No two agents control the same edge: $\forall i \neq j, E_i \cap E_j = \emptyset$.

The mechanism chooses a valid flow from s to t . A valid flow is a function $f : E \rightarrow \mathbb{R}$ such that $\forall (u,v) \in E, f(u,v) \leq c(u,v)$, $\forall (u,v) \in E, f(u,v) = -f(v,u)$, and $\forall u \in V - \{s,t\}, \sum_{v \in V} f(u,v) = 0$.

Each agent values the flow according to the total flow going through its edges. Let f be the valid flow chosen by the mechanism, and E_f the set of edges in f through which there is a positive flow: $E_f = \{e \in E \mid f(e) > 0\}$. We denote the set of A_i 's edges used in the flow f by: $E_{f,i} = E_f \cap E_i$. The agent's valuation of the flow is $v_i(f) = \sum_{e \in E_{f,i}} f(e)$.

When the mechanism is given the agents' true types, $\theta = E_1, E_2, \dots, E_I$, we want it to choose the flow that maximizes the total utility of the agents. The mechanism would be allocatively-efficient if it chooses $f^*(\theta) = \arg \max_f \sum_i \sum_{e \in E_{f,i}} f(e)$. In a *layered* graph the flow that maximizes the sum of agents' utilities is the maximal flow (proof of this is given in the full paper). Thus, if each agent truthfully declares its subset of edges, the mechanism can easily compute $f^*(\theta)$ by running a maximal flow algorithm, such as the Edmonds-Karp algorithm.

As shown in the full version of this paper, a naive mechanism for the self-interested flow problem, with no payments to the mechanism, is not strategy-proof. An agent may declare a subset of the edges it controls, to change the flow that the mechanism chooses to a flow that the agent values more.

1.2 Self-Interested Network Flow Mechanism

We assume quasi-linear utility. Each agent pays the mechanism a payment p_i , and its utility is $u_i(f) = v_i(f) - p_i$. We now show that by using a straightforward payment rule, we make finding a beneficial manipulation NP-hard. The payment rule we use is simple: each agent pays the mechanism a constant of c for each edge it declares it owns. Let $E'_i \subset E_i$ be the subset of edges an agent declares it owns. Then $p_i(E'_i) = c|E'_i|$. We also give a minor correction to make sure the mechanism is individual-rational. The payment p_i gives the agent a utility of 0 when the agent's valuation of the chosen flow is less than the payment to the mechanism. Assume that A_i knows E'_j for all $j \neq i$. It can easily calculate the utility it would get by truthfully declaring all its edges. How hard is it for i to find a subset of edges it could declare to the mechanism so as to gain a higher utility? We define the problem of *finding* a useful manipulation in the self-interested network flow domain as follows.

FLOW-EDGE-SUBSET: We are given a layered graph flow network, with the capacity function $c : E \rightarrow \mathbb{R}$, E_{-i} the declared edges of the other agents, and E_i , the set of our edges. We are also given the constant c of the payment, and we know that if we declare that we have k edges, our payment to the mechanism would be $p_i = ck$. We assume the mechanism prefers a maximal flow that maximizes *our* utility of all the possible maximal flows. We are also given a constant k , the target utility for A_i . We are asked if there is a subset of A_i 's edges $E'_i \subset E_i$, such that the maximal flow chosen by the mechanism, $f^*(E_1, \dots, E_{i-1}, E'_i, E_{i+1}, \dots, E_I)$ gives A_i a utility of at least k : $u_i(f^*, p_i) = v_i(f^*) - p_i = \sum_{e \in E_{f^*,i}} f(e) - c|E'_i| \geq k$.

A reduction, given in the full version of this paper, indicates that FLOW-EDGE-SUBSET is an NP-complete problem. The reduction given is from a general VERTEX-COVER instance to a FLOW-EDGE-SUBSET problem.

Although we do not include the full details of the reduc-

tion here, we do give some intuition about how it was constructed. Given the original VERTEX-COVER instance, we build a layer graph, where one of the agents, A_1 , controls all the edges between two consecutive layers. Each edge A_1 owns corresponds to a vertex in the original VERTEX-COVER graph. The network flow graph is constructed so that in order for A_1 to maximize its valuation, it must declare a subset of edges that is a *vertex cover* of G . However, in order to minimize its payment to the mechanism, A_1 must choose as few edges as possible. If the payment constant c is small enough, A_1 's top priority is to make its chosen vertices a vertex-cover, and only then choosing as few edges as possible. This makes its preferred choice the minimal vertex cover of the original graph.

2 Conclusions and Future Directions

We have suggested a mechanism for the distributed network flow problem with self-interested agents. With a proper choice of the payment constant c , finding a beneficial manipulation is an NP-complete problem. Since finding a beneficial manipulation is intractable, we encourage agents to truthfully report their preferences. In this case, the mechanism would choose the result maximizing the sum of agents' utilities, and we have an allocatively-efficient mechanism.

Given some $\epsilon > 0$, we can make the mechanism ϵ -budget-balanced, by choosing a small enough payment constant $c < \frac{\epsilon}{n(|E|+|V|)}$, where n is the number of agents. This way, all of the agents together pay less than ϵ . The mechanism we have described is also individual-rational. Therefore, in the domain of network flows, this achieves an individual-rational, allocatively-efficient, ϵ -budget-balanced, and strategy-resistant mechanism. The standard VCG solution in this domain would be only weakly budget-balanced, but strategy-proof.

Impossibility results ([Green and Laffont, 1977] and [Hurwicz, 1975]) indicate that no direct-revelation mechanism can achieve strong budget-balance without sacrificing either allocative-efficiency or strategy-proofness. We believe that in many cases, trading strategy-proofness for strategy-resistance is a fair price to pay for achieving strong budget-balance.

Our strategy-resistance rested on the worst case assumptions of NP-hardness, which only indicates that a certain problem has *some* hard instances. A stronger notion of strategy-resistance could also require the manipulation problem to have no approximation methods, that it be in some harder complexity class, or that it be hard to find *any* (not just the optimal) beneficial manipulation. This work constitutes a first step in establishing the notion of strategy-resistance.

References

- [Green and Laffont, 1977] Jerry Green and Jean-Jacques Laffont. Characterization of satisfactory mechanisms for the revelation of preferences for public goods. *Econometrica*, 45(2):427–38, March 1977.
- [Hurwicz, 1975] Leonid Hurwicz. On the existence of allocation systems whose manipulative Nash equilibria are pareto-optimal. In *3rd World Congress of the Econometric Society (Unpublished)*, 1975.

The Evolution of Artificial Social Systems

Guido Boella

Dipartimento di Informatica
Università di Torino
Italy
guido@di.unito.it

Leendert van der Torre

CWI Amsterdam
and Delft University of Technology
The Netherlands
torre@cwi.nl

1 Introduction

The basic idea of the artificial social systems approach of Shoham and Tennenholtz [1995; 1997] is to add a mechanism, called a social law, that will minimize the need for both centralized control and on-line resolution of conflicts. A social law is defined as a set of *restrictions* on the agents' activities which allow them enough freedom on the one hand, but at the same time constrain them so that they will not interfere with each other. Several variants have been introduced to reason about the design and emergence of social laws. However, existing models of artificial social systems cannot be used for the evolution of such systems, because these models do not contain an explicit representation of the social laws in force. In this paper we use enforceable social laws [Boella and van der Torre, 2005] to address the question how artificial social systems can be extended to reason about the evolution of artificial social systems.

2 Artificial social systems and social laws

Shoham and Tennenholtz [1995] introduce social laws in a setting without utilities. They define also *rational* social laws [Shoham and Tennenholtz, 1997] as social laws that improve a social game variable. A game or multi-agent encounter is a set of agents with for each agent a set of strategies and a utility function defined on each possible combination of strategies. We extend artificial social systems with a control system, called a normative system, to model enforceable social laws. Following Boella and Lesmo [2002], the normative system is represented by a socially constructed agent called the normative agent or agent 0. In [Boella and van der Torre, 2005], the normative system is represented by the set of control strategies of agent 0, but not by a utility function.

Definition 1 A normative game (or a normative multi-agent encounter) is a tuple $\langle N, R, S, T, U_1, U_2 \rangle$, where $N = \{0, 1, 2\}$ is a set of agents, R , S and T are the sets of strategies available to agents 0, 1 and 2 respectively, and $U_1 : R \times S \times T \rightarrow \mathbb{R}$ and $U_2 : R \times S \times T \rightarrow \mathbb{R}$ are real-valued utility functions for agents 1 and 2, respectively.

We use here as game variable the maximin value, following Tennenholtz [2000]. This represents safety level decisions, see Tennenholtz' paper for a motivation.

Definition 2 Let R , S and T be the sets of strategies available to agent 0, 1 and 2, respectively, and let U_i

be the utility function of agent i . Define $U_1(R, s, T) = \min_{r \in R, t \in T} U_1(r, s, t)$ for $s \in S$, and $U_2(R, S, t) = \min_{r \in R, s \in S} U_2(r, s, t)$ for $t \in T$. The maximin value for agent 1 (respectively 2) is defined by $\max_{s \in S} U_1(R, s, T)$ (respectively $\max_{t \in T} U_2(R, S, t)$). A strategy of agent i leading to the corresponding maximin value is called a maximin strategy for agent i .

A social law is useful with respect to an efficiency parameter q if each agent can choose a strategy that guarantees it a payoff of at least q .

Definition 3 Given a normative game $g = \langle N, R, S, T, U_1, U_2 \rangle$ and an efficiency parameter q , we define a social law to be a restriction of S to $\bar{S} \subseteq S$, and of T to $\bar{T} \subseteq T$. The social law is useful if the following holds: there exists $s \in \bar{S}$ such that $U_1(R, s, \bar{T}) \geq q$, and there exists $t \in \bar{T}$ such that $U_2(R, \bar{S}, t) \geq q$.

A social law is quasi-stable if an agent does not profit from violating the law, as long as the other agent conforms to the social law (i.e., selects strategies allowed by the law).

Definition 4 Given a normative game $g = \langle N, R, S, T, U_1, U_2 \rangle$, and an efficiency parameter q , a quasi-stable social law is a useful social law (with respect to q) which restricts S to $\bar{S}$ and T to $\bar{T}$, and satisfies the following: there is no $s' \in S \setminus \bar{S}$ which satisfies $U_1(R, s', \bar{T}) > \max_{s \in \bar{S}} U_1(R, s, \bar{T})$, and there is no $t' \in T \setminus \bar{T}$ which satisfies $U_2(R, \bar{S}, t') > \max_{t \in \bar{T}} U_2(R, \bar{S}, t)$.

When the set of strategies R of agent 0 is a singleton, then our definitions reduce to those of Tennenholtz [2000]. With the extension of agent 0 representing the control system we define enforceable social laws as quasi-stable social laws in normative games where the strategies of agent 0 may have been restricted [Boella and van der Torre, 2005].

Definition 5 Given a normative game $g = \langle N, R, S, T, U_1, U_2 \rangle$, and an efficiency parameter q , a social law (i.e., a restriction of S to $\bar{S} \subseteq S$, and of T to $\bar{T} \subseteq T$) is enforceable if there is a restriction of R to $\bar{R} \subseteq R$ such that $\bar{S}, \bar{T}$ is quasi-stable in the normative game $g = \langle N, \bar{R}, S, T, U_1, U_2 \rangle$.

Computational problems can be defined to find enforceable social laws (with respect to an efficiency parameter).

3 Representing social laws

We extend normative games with a utility function of agent 0, to represent the norms which are enforced. Since agent 0 is a socially constructed agent, in the sense of Searle [1995], its utility function can be updated. In particular, the enforcement of a social law by $\bar{R} \subseteq R$ is represented by giving $\bar{R}$ strategies a high utility, and $R \setminus \bar{R}$ strategies a low utility. Moreover, we go beyond the framework of enforceable social laws by varying the utility of agent 0 depending on the strategies played by the other agents, and by considering incremental updates of the utility function to represent the evolution of artificial social systems. Formally, we extend a normative game with a utility function $U_0 : R \times S \times T \Rightarrow \mathbb{R}$, we define $U_0(r, S, T) = \min_{s \in S, t \in T} U_0(r, s, t)$ for $r \in R$, and we define useful and quasi-stable social laws in the obvious way. Enforced social laws are defined as follows.

Definition 6 Given a normative game $g = \langle N, R, S, T, U_1, U_2 \rangle$, and an efficiency parameter q , a social law (i.e., a restriction of S to $\bar{S} \subseteq S$, and of T to $\bar{T} \subseteq T$) is enforced if there is a unique restriction of R to $\bar{R} \subseteq R$ such that $\bar{R}, \bar{S}, \bar{T}$ is quasi-stable.

3.1 Identification of enforced social laws

The game in Table 1 illustrates that the computational problem to find quasi-stable laws corresponds in extended normative games to the identification of enforced social laws. The table should be read as follows. Strategies are represented by literals, i.e., atomic propositions or their negations. Each table represents the sub-game given a strategy of agent 0, represented by $\neg n$ and n , respectively. Agent 1 is playing columns and agent 2 is playing rows. The values in the tables represent the utilities of agent 0 (in italics), 1 and 2.

$\neg n$	p	$\neg p$	n	p	$\neg p$
q	3,3,3	0,4,1	q	3,3,3	1,2,1
$\neg q$	0,1,4	1,2,2	$\neg q$	1,1,2	0,2,2

Table 1: What is the enforced social law?

Agent 0 (the normative system) can play strategy $\neg n$ or n , agent 1 can play strategy p or $\neg p$, agent 2 can play strategy q or $\neg q$. When the normative system plays $\neg n$, the sub-game of agent 1 and 2 is a classical prisoner's dilemma. Intuitively, the strategy $\neg n$ corresponds to the state before the social law is introduced, and n corresponds to the introduction of a control system that sanctions an agent for deviating from p, q . For example, the utility of agent 1 in $\neg p, q, n$ (2) is lower than its utility in $\neg p, q, \neg n$ (4) due to this sanction.

When the normative system plays n , the agents are always worse off compared to the normative agent playing $\neg n$, all else being equal. Nevertheless, due to the dynamics of the game, the overall outcome is better for both agents. For example, in the sub-game defined by strategy $\neg n$, the only Nash equilibrium is 2, 2. Now suppose we set the efficiency parameter to 3, which means that all agents will be better off. If the normative system plays n , then the sub-game has a Nash equilibrium which is the (Pareto optimal) 3, 3. This explains why the agents accept the possibility to be sanctioned.

3.2 Iterated design of enforced social laws

The social law design problem is, given a normative game, to define a new utility function for the normative system. The principle that we like to maintain as much as possible from the existing social laws can be represented by the use of the principle of minimal change. Table 2 represents the evolution of an artificial social system by an incremental increase of the utility of agent 0 to the efficiency parameter of the new social law.

$\neg n_1, \neg n_2$	p_1	p_2	$\neg p_1, \neg p_2$
q_1	0,3,3	0,4,1	0,6,0
q_2	0,1,4	0,2,2	0,0,0
$\neg q_1, \neg q_2$	0,0,6	0,0,0	0,0,0
n_1	p_1	p_2	$\neg p_1, \neg p_2$
q_1	1,3,3	1,4,1	1,0,0
q_2	1,1,4	1,2,2	1,0,0
$\neg q_1, \neg q_2$	1,0,0	1,0,0	1,0,0
n_2	p_1	p_2	$\neg p_1, \neg p_2$
q_1	3,3,3	3,1,1	3,0,0
q_2	3,1,1	3,0,0	3,0,0
$\neg q_1, \neg q_2$	3,0,0	3,0,0	3,0,0

Table 2: Iterated design

The first table represents that the normative system does not impose a control system, the second table represents that there is a sanction for playing $\neg p_1, \neg p_2$ or $\neg q_1, \neg q_2$, and the third table represents that there is an additional sanction for playing something else than p_1 and q_1 . The first social law is $\bar{S} = \{p_1, p_2\}, \bar{T} = \{q_1, q_2\}$ based on control system $\bar{R} = \{n_1, n_2\}$, and the second social law is $\bar{S} = \{p_1\}, \bar{T} = \{q_1\}$ based on control system $\bar{R} = \{n_2\}$.

References

- [Boella and Lesmo, 2002] G. Boella and L. Lesmo. A game theoretic approach to norms. *Cognitive Science Quarterly*, pages 492–512, 2002.
- [Boella and van der Torre, 2005] G. Boella and L. van der Torre. Enforceable social laws. In *Procs. of AAMAS'05*. ACM Press, 2005.
- [Searle, 1995] J.R. Searle. *The Construction of Social Reality*. The Free Press, New York, 1995.
- [Shoham and Tennenholtz, 1995] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: offline design. *Artificial Intelligence*, 73 (1-2):231 – 252, 1995.
- [Shoham and Tennenholtz, 1997] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94 (1-2):139 – 166, 1997.
- [Tennenholtz, 2000] M. Tennenholtz. On stable social laws and qualitative equilibria. *Artificial Intelligence*, 102 (1):1–20, 2000.

Fast convergence to satisfying distributions

Teddy Candale and Sandip Sen

Mathematical and Computer Sciences Department

University of Tulsa

{teddy-candale,sandip}@utulsa.edu

Abstract

We investigate an environment where self-interested agents have to find high-quality service resources. Agents have common knowledge about resources which are able to provide these services. The performance of resources is measured by the satisfaction obtained by agents using them. The performance of a resource depends on its intrinsic capability and its current load. We use a satisfying rather than an optimizing framework, where agents are content to receive service quality above a threshold. We introduce a formal framework to characterize the convergence of agents to a state where each agent is satisfied with the performance of the service it is currently using. We analyzed the convergence behavior of such a system and identified a mechanism to speed up convergence.

1 Introduction

Agents searching for high-quality services can use either their own interaction experience or referrals from peer agents. We assume that agents are interested in finding a quality of service which exceeds an acceptable performance threshold. The performance of a resource depends on its intrinsic characteristics and is inversely correlated to the current workload it is handling. Individual agent satisfaction depends both on the resource selected and choices made by the other agents. Two agents may have different satisfactions for the same performance level of a resource. Efficient decentralized protocols for finding satisfying resources are necessary. Reaching a suitable global state where all agents are satisfied with the resources they are using can be viewed as a coordination problem. Resource choice decisions that appear to be most appropriate for an individual may prove to be damaging for a part or the whole agent group. Selecting the locally optimal action can then increase the number of conflicts of interests where resources are shared. Besides, knowledge about the global state is likely to be inaccurate in dynamic, partially known, open environments.

Coordination in such environments is a challenging problem which, however, needs reasonably good solutions. There are two primary motivations for the current work: (a) characterize the properties of agent satisfaction functions and re-

source selection strategies that guarantees convergence to stable system states, and (b) providing guidelines for designing such systems with fast convergence to stable states. Our proposed, distributed coordination scheme ensures convergence to a satisfactory system state. We experimentally highlight the effect of different protocol and system parameters that influence the coordination process and expedites convergence to equilibrium states

2 Framework

Let $\mathcal{E} = \langle \mathcal{A}, \mathcal{R}, perf, L, S, \Gamma \rangle$ where: (i) $\mathcal{A} = \{a_k\}_{k=1..K}$ is the set of agents, (ii) $\mathcal{R} = \{r_n\}_{n=1..N}$ is the set of resources, (iii) $f : \mathcal{R} \times \mathbb{R} \rightarrow [0, 1]$, intrinsic performance function of a provider, (iv) $L = \mathcal{A} \rightarrow \mathbb{R}_+$, daily load assigned to agents, (v) $S : \mathcal{A} \times [0, 1] \rightarrow [0, 1]$, satisfaction function for agents, (vi) $\Gamma = \{\gamma_1, \dots, \gamma_K\}$, set of satisfaction thresholds of agents. If a set $\mathcal{A}_n^d$ of agents use the resource r_n on day d then the feedback received by every agent in $\mathcal{A}_n^d$ at the end of the day d is $perf = f(r_n, \sum_{a \in \mathcal{A}_n^d} L(a))$. An agent $a_k \in \mathcal{A}_n^d$ evaluates the performance of r_n by the the satisfaction it obtained and is given by $s = S(a_k, perf)$. An agent is satisfied if this performance s is above a threshold γ_k . Consequently, two agents may have different satisfactions for the same resource performance.

Agents are interested in obtaining satisfactions above a threshold. Our aim is to design an environment where every agent will have the opportunity to find satisfying resources. The following definitions formalize this idea.

Definition 1 (Distribution of agents over providers)

We call **distribution of agents over resources** the set $D = \{\mathcal{A}_n\}_{n=1..N}$ where $\mathcal{A}_n$ is the set of agents using resource r_n . The set of distributions is denoted by $\mathcal{D}$.

Definition 2 (Γ -acceptable distribution) A distribution D is said to be **Γ -acceptable** if each agent is satisfied by the resource they use in D . The set of Γ -acceptable distributions is denoted by $\mathcal{D}_\Gamma$.

A Γ -acceptable distribution is an equilibrium concept and our goal is to develop mechanisms that enable a group of agents to converge to such a distribution. We now present two properties needed by any resource-selection algorithm to ensure convergence to satisfying distributions. In the following, $\mathcal{Alg} = \{\mathcal{Alg}_k\}_{k=1..K}$ is the set of algorithms used by

agents to choose a resource and $\mathcal{H}_d$ denotes the set of possible histories after d days and $\mathcal{H} = \bigcup_{d=1}^{+\infty} \mathcal{H}_d$ the set of all possible histories.

Definition 3 (Weighted directed distribution graph) : A directed graph $\mathcal{G} = (V, E, \omega)$ is called a **weighted directed distribution graph** associated with $\mathcal{Alg}$ if and only if (i) $V \subset \mathcal{D}$, (ii) $E \in V \times V$, (iii) $(D_1, D_2) \in E \implies \exists d \in \mathbb{N}, \exists H_d \in \mathcal{H}_d$ s.t. $(\omega(D_1, D_2, H_d) \neq 0)$ where $\omega : \mathcal{H}_d \rightarrow [0, 1]$. $\omega(H_d)$ is the probability to reach the distribution D_2 given agents are in distribution D_1 and they use algorithms in $\mathcal{Alg}$.

In weighted directed distribution graphs, an arrow from D_1 to D_2 is present if it is possible to reach D_2 from D_1 with a non-zero probability using algorithm $\mathcal{Alg}$.

Property 1 A distributed algorithm $\mathcal{Alg}$ stops when a Γ -acceptable distribution is reached.

Property 2 Let $\mathcal{G} = (V, E, \omega)$ be the graph associated to $\mathcal{Alg}$. $\mathcal{G}$ is such that at least one node of $\mathcal{G}$ is a Γ -acceptable distribution. A Γ -acceptable distribution is reachable from any distribution with a non-zero probability.

Theorem 1 Every algorithm $\mathcal{Alg}$ which respect Properties 1 and 2 converge to a Γ -acceptable distribution.

Theorem 2 Let $\forall k = 1..K, \alpha_k : \mathcal{H} \rightarrow [0, 1]$. $\mathcal{Alg}$ converges to a Γ -acceptable distribution if the following conditions are satisfied: (i) $\mathcal{Alg}$ respects Property 1, (ii) a Γ -acceptable distribution exists, (iii) agents are identical, (iv) agents explore with probability $\alpha_k(H)$ if they are unsatisfied, (v) $\forall k = 1..K, \exists \delta_k > 0$ s.t. $\forall H \in \mathcal{H} \alpha_k(H) \geq \delta_k$.

We omit the proof of the above theorems due to space limitations. Theorem 2 concludes that exploration ensures convergence. The nature of the agents, e.g., self-interested, co-operative, deceptive does not influence the existence of convergence as long as minimum exploration is guaranteed. Another strength of the theorem is that it does not require the probability of exploration to be constant. An agent may adapt this probability based on whether it believes it should explore more or less.

3 Speed of convergence

In this paper, we will reason using two axes: a measurement of goodness of a state that we will call *entropy, the number of agents who can move at the same time*.

Definition 4 (Entropy) Given an environment where agents are identical and resource r_n has capacity C_n , we call entropy of a distribution D : $\mathcal{E}(D) = \sum_{n=1}^N \max(0, |\mathcal{A}_n| - C_n)$.

Each Γ -acceptable distribution has zero entropy. The lower the entropy the better the distribution since less agents are unsatisfied.

We claim that when agents choose their actions based on local perspective only, the system is likely to move from a distribution with a low entropy to one with a higher entropy and vice versa. Such oscillations can be controlled by limiting the number of agents moving simultaneously, K_{move} . We experimentally show K_{move} has a critical influence on

the convergence speed. A high value for K_{move} leads to system oscillations and hence is undesirable. A low value for K_{move} guides the system toward a Γ -acceptable distribution but the convergence process slows down progressively when approaching a perfectly coordinated distribution. The decrease in entropy is almost monotonic. With an intermediate value of K_{move} , entropy decreases at a faster rate, but has more fluctuations.

4 Related work

Coordinated search of resources is a challenging problem in multiagent systems. Sen et al [2] show that information can negatively impact agent coordination to find balanced distribution among resources. Rustogi & Singh [1] prove that high inertia hasten convergence when knowledge increases but low inertia performs better with little knowledge.

We believe that a more comprehensive understanding of system behavior can be obtained by studying system entropy and controlling the number of simultaneously moving agents, K_{move} . Our approach provides a more detailed characterization of such systems while confirming the general conclusions from Rustogi & Singh [1].

5 Conclusion

We studied the properties of agent decision functions and their interaction with the distribution of agents over resources so that stable distributions, with no incentive for any agent to move, are produced. We characterized the behavior of such a system of agents in terms of the entropy of the system. We showed that the entropy of a system can be steadily reduced, and hence convergence accelerated, by increasing the inertia or reducing the number of agents that can move simultaneously. A distributed implementation of controlling inertia is proposed in the form of an exploration parameter. We are evaluating the scale-up property of such a system. Our work provides guidance to selection of system protocols and parameters, including satisfaction functions, exploration attitude, etc.

References

- [1] Sudhir K. Rustogi and Munindar P. Singh. Be Patient and Tolerate Imprecision: How Autonomous Agents can Coordinate Effectively. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 512–519, Morgan Kaufmann Publishers Inc. 1999 ISBN: 1-55860-613-0
- [2] Sandip Sen and Neeraj Arora and Shounak Roychowdhury. Using limited information to enhance group stability. In the journal *Int. J. Hum.-Comput. Stud.*, Volume 48, Number 1, Year 1998, Pages 69–82 ISSN 1071-5819, Academic Press, Inc.,

A rule language for modelling and monitoring social expectations in multi-agent systems

Stephen Cranefield

Department of Information Science, University of Otago
PO Box 56, Dunedin, New Zealand
scranefield@infoscience.otago.ac.nz

Abstract

This paper proposes a rule language for defining social expectations based on a metric interval temporal logic with past and future modalities and a current-time binding operator. An algorithm for run-time monitoring compliance of rules in this language based on formula progression is also outlined.

1 Introduction

The study of *electronic institutions*—explicit declarative models of the rules governing particular open systems of autonomous agents—has gained much recent attention [Cortés, 2004]. There has been a significant amount of research on statically verifying properties of institutions as well as interpreting institutions to manage or guide agent interaction. However, there has been little attention paid to mechanisms for run-time compliance checking, i.e. monitoring events in a running agent system, determining the future expectations of agents’ behaviour according to norms of the institution, and checking if these are fulfilled or violated. This paper focuses on this issue by presenting (i) a logic named hyMITL[±] that combines CTL[±] [Verdicchio and Colombetti, 2003] with Metric Interval Temporal Logic (MITL) [Alur *et al.*, 1996], as well as features of hybrid logics [Blackburn *et al.*, 2001], and (ii) a subset of hyMITL[±] that provides a rule language for defining social expectations. We have implemented an algorithm for monitoring social expectations encoded in this language, based on the technique of formula progression from the planning system TLPlan [Bacchus and Kabanza, 1998].

2 Syntax of hyMITL[±]

The syntax of hyMITL[±] is defined as follows:

$$\begin{aligned}\phi &::= p \mid \neg\phi \mid \phi \wedge \phi \mid \forall x.\phi_x \mid X^+\phi \mid X^-\phi \mid \\ &\quad \phi U_I^+\phi \mid \phi U_I^-\phi \mid A\phi \mid E\phi \mid \downarrow^u x.\phi_x \mid I \\ I &::= (-\infty, +\infty) \mid [b, b] \mid [b, b) \mid (b, b] \mid (b, b) \\ b &::= a \mid +d \mid -d\end{aligned}$$

where

- p is an atomic formula from a first order language L .

- ϕ_x denotes a formula ϕ in which variable x is free (i.e. not bound by $\forall$ or $\downarrow$).
- a and d are terms, possibly containing variables, that denote (respectively) absolute points in time and durations (e.g. using the language proposed by Verdicchio and Colombetti [2004]).
- u is a unit selector on the $\downarrow$ binding operator, referring to the desired granularity of time (e.g. year or minute) for binding x to the current time (with default value now).

We constrain the use of variables within interval bounds b : any such variables must be bound by an enclosing $\downarrow$ operator.

In this logic, the temporal operators X^+ and X^- mean “in the next state” and “in the previous state” respectively. $\phi U_I^+\psi$ asserts that ϕ will remain true from the current state for some (possibly empty) sequence of consecutive future states, followed by a state that is within the time interval I and for which ψ holds. X^- and U_I^- are defined similarly, but in the past direction.

The bounds of intervals can be specified either relatively or absolutely—a prefix of “+” or “−” indicates a relative time value. When qualifying U^- , the interval bounds are written in the reverse order from usual, e.g. $[-2 \text{ hours}, -3 \text{ hours}]$.

A and E are temporal path quantifiers. They assert that the formula that follows the operator applies to all, or respectively at least one, of the possible sequences of states passing through the current state.

The $\downarrow$ operator is based on the “binder” operator used in hybrid logics [Blackburn *et al.*, 2001]. It binds a variable to a term denoting the current date/time, possibly ‘rounded down’ to a particular degree of precision, e.g. to the start of the current year, month or day, depending on the unit indicated by the superscript u .

We use the usual derived operators of first order predicate logic, as well as the standard abbreviations for existential and universal quantification over states in a path: $F_I^+\phi \equiv \text{true } U_I^+\phi$ and $G_I^+\phi \equiv \neg F_I^+\neg\phi$, with similar definitions for F_I^- and G_I^- . We define future and past “weak until” operators in the following way:

$$\phi W_{[l,u]}^+\psi \equiv \downarrow t. (G_{[t,u]}^+\phi \vee \phi U_{[l,u]}^+\psi)$$

with similar definitions for intervals with open bounds and for W^- .

$$\begin{aligned}
& \text{AG}^+ (\text{Done}(c, \text{make_payment}(c, p, \text{amount}, \text{prod_num})) \wedge [t, t+1 \text{ week}] \rightarrow \\
& \quad \downarrow^{\text{week}} w. ((\neg \text{F}_{[-0, w]}^- \text{Done}(p, \text{send_report}(c, \text{prod_num}, w)) \rightarrow \text{F}_{(+0, w+1 \text{ week})}^+ \text{Done}(p, \text{send_report}(c, \text{prod_num}, w)))) \\
& \quad \text{W}_{[+0, w+52 \text{ weeks}]}^+ \\
& \quad \text{Done}(c, \text{cancel_order}(c, p, \text{prod_num})))
\end{aligned}$$

Figure 1: A rule expressing the terms of service offered by agent p

3 The Rule Language

We identify a subset of $\text{hyMITL}^\pm$ that is suitable for encoding social expectations in a form that can be used in run-time compliance testing: the set of all formulae of the following form:

$$\text{AG}^+ \forall_{1 \leq i \leq n} x_i. (\phi \rightarrow \psi)$$

for $n \geq 0$, where ϕ and ψ are linear-time formulae (i.e. they do not contain **A** or **E**), the free variables of ϕ and ψ are precisely the quantified variables $\{x_1, \dots, x_n\}$, ϕ and ψ do not contain any occurrences of $\forall$, and any occurrence of $\exists$ must be of a certain restricted form that is equivalent to TLPlan's bounded existential quantification [Bacchus and Kabanza, 1998].

Figure 1 shows an example rule encoding a service offer by an agent p that can provide weekly reports on a particular market for an annual fee. This rule states that once payment of the required amount is made by the potential client c within the validity period of the offer (one week), the service-providing agent is committed to sending a report to the client once a week for 52 weeks or until the client cancels the order.

4 The Compliance Monitoring Process

Rules in our language are intended to be used in the following compliance-testing process:

Given a current state and the history of all prior states and their associated times, for each rule, match the left hand side against the current state and history, resulting in a set of instances of the right hand side. Add these instances to the set of current expectations, then check all expectations to see which are fulfilled or violated. Any expectations that cannot yet be evaluated because they involve future states are ‘progressed’ to the next state when it is created by an event observation.

The progression of unfulfilled expectations from one state to the next uses a modified version of the progression algorithm of Bacchus and Kabanza [1998]. This algorithm generates a formula expressing what needs to be true in the new state if the input expectation was required to be true in the previous state, but was not yet able to be evaluated there.

Once new expectations have been computed by matching the rules to the current state and history, each formula in the combined set of old and new expectations is partially evaluated, resulting in true or false if the truth of the formula can be determined yet, and otherwise returning a formula equivalent to the original one (given the facts in the history states), but modified where possible to make progression and future evaluation easier. This partial evaluation is separated out from

the progression step as it is performed in the current state—before a new state is created by a new event occurrence, and therefore before formula progression can be done (this needs the time difference between the old and new state as an input).

5 Conclusion

This paper has defined a rule language for defining social expectations based on a metric interval temporal logic and has outlined an algorithm that can be used at run time in a multi-agent system to monitor when expectations are generated, fulfilled and violated. A prototype implementation of the compliance testing algorithm has been implemented using SWI Prolog. Most of the features described here have been implemented, although currently the system is not connected to an agent—it uses a static database of states and their facts—and integers are used to represent times.

The semantics for $\text{hyMITL}^\pm$, the algorithms used, a discussion of related work and future extensions to this work, and acknowledgements can be found in the full version of this paper [CraneField, 2005].

References

- [Alur *et al.*, 1996] R. Alur, T. Feder, and T. A. Henzinger. The benefits of relaxing punctuality. *Journal of the ACM*, 43(1):116–146, 1996.
- [Bacchus and Kabanza, 1998] F. Bacchus and F. Kabanza. Planning for temporally extended goals. *Annals of Mathematics and Artificial Intelligence*, 22(1-2):5–27, 1998.
- [Blackburn *et al.*, 2001] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, chapter 7, pages 436–447. Cambridge University Press, 2001.
- [Cortés, 2004] U. Cortés. Electronic institutions and agents. *AgentLink News*, 15:14–15, September 2004.
- [CraneField, 2005] S. CraneField. A rule language for modelling and monitoring social expectations in multi-agent systems. Discussion Paper 2005/01, Department of Information Science, University of Otago, 2005. <http://www.otago.ac.nz/informationscience/pubs/papers/papers/dp2005-01.pdf>.
- [Verdicchio and Colombetti, 2003] M. Verdicchio and M. Colombetti. A logical model of social commitment for agent communication. In *Proc. of AAMAS 2003*, pages 528–535. ACM Press, 2003.
- [Verdicchio and Colombetti, 2004] M. Verdicchio and M. Colombetti. Dealing with time in content language expressions. In *Proc. of the AAMAS 2004 Workshop on Agent Communication*, pages 90–104, 2004.

A Framework for Communication Planning on Mobile Devices

Joseph B. Kopena and William C. Regli

{tjkopena, regli}@cs.drexel.edu

Drexel University

Philadelphia, PA, USA.

Abstract

In mobile computing, communicative acts are not free. Costs such as power and bandwidth consumption are prominent issues. In addition, resources vary widely across hardware and operating context. Agents in these settings must account for these costs and adapt to available capabilities. This poster presents a planning optimization formalization of this problem, enabling service-based agents to reason about and conduct communication using local and network accessible resources.

1 Introduction

Mobile computing is increasingly common in urban settings. Laptops, PDAs, and other small, network-capable computing devices pervade society. These devices along with the many and varied applications of mobile computing highlight an important observation: communicative actions are not free. In fact, there are a variety of associated costs, including:

- *Power.* On many devices, networking hardware consumes more power than all other systems combined [Vahdat *et al.*, 2000]. Transmissions consume substantial amounts of limited battery supplies. Frequent communication significantly reduces operational lifespan, a critical obstacle against many applications.
- *Bandwidth.* Wireless link capacities are typically much lower. Throughput in 802.11b MANETs is often less than 50% the theoretical maximum of 11 Mbps [Xylomenos *et al.*, 2001]. In contrast, traditional wired networks can reach upward of 100 Mbps. Communication must be sparing to conserve such limited bandwidth.

Communication resources also vary between hosts and operating context. Hardware such as 802.11 cards, Bluetooth chips, and CDMA modules possess different properties and capabilities. Routing algorithms and other software present dissimilar interfaces and sensing abilities an agent may utilize [Kopena *et al.*, 2005]. However, it cannot necessarily know *a priori* which communication services will be available on its host, and must instead autonomously discover and utilize them. In addition, this set may change over time, through user actions such as undocking the host or natural events, e.g. network cards failing as power dwindles.

2 Example: Calendar Agent

Consider a calendar on a PDA with cellular and wireless capabilities. As events are posted it synchronizes with a base computer, which in turn notifies family and friends. Wireless ethernet can only connect to the base computer when in range of an access point. The cellular module can always make an Internet connection to the base computer via a satellite, but consumes substantial amounts of power and charges a fee.

To post updates in all situations, the calendar must discover and utilize both interfaces. However, cellular connections should be minimized: they should not be used if wireless is available, or to post minor or far-off events. These should be collected until several may be transmitted or ethernet can be used. However, cellular connections should be used to post imminent changes so that the base computer may begin synchronizing affected friends, family, or co-workers.

3 Formalization

Utilizing available network services and intelligently, effectively conducting communications may be achieved by releasing agents and applications from fixed application-layer protocols and messages. Communicative actions are not static. An agent may plan, schedule, alter, and parameterize them, adapting and responding to conditions and events.

Even a simple agent may deliberate on when to deliver a fixed message in order to minimize communication costs. It may also plan on which network services to use based on advertised descriptions. More sophisticated agents may alter and plan on the propositional content of messages to maximize their utility: power-conscious agents may aggregate low priority messages into one burst, economizing fixed costs.

In this work, such reasoning is accomplished through utility-based planning. Services such as network interfaces are advertised and treated as plan actions with attached costs. Communication goals are modeled as utilities of doxastic states. These elements are represented within a formal situation calculus, enabling exchange of well-defined service descriptions and limited commitment to reasoning mechanisms.

Planning Framework. This work defines the above planning problem using the Process Specification Language (PSL) [Grüniger and Menzel, 2003], an ISO standard (ISO 18629) for process modeling. At its core, PSL is an extensive first-order situation calculus axiomatization. A deductive

planning problem $(A, I, \psi, C) \rightarrow \rho$ is defined on top of the PSL Core, Occurrence Trees, and Discrete States theories:

- A are the *activity axioms*, preconditions and effects.
- I defines *initial fluent state*: $\forall \delta_i \cdot \text{initial}(\delta_i) \supset \delta_i \models I$.
- A *chain* δ is a sequence of successive legal occurrences such that $\text{initial}(\delta_1) \wedge \delta_n \models$ the episode delimiter ψ .
- C defines *utility summands* μ over occurrences, related by $\text{occ-util-summ}(\delta_i, \mu)$.
- *Occurrence utility*, $\text{occ-util}(\delta_i)$, is defined as $\sum \mu$ over $\{\mu | \text{occ-util-summ}(\delta_i, \mu)\}$.
- *Utility of a chain*, $\text{utility}(\delta)$, equals $\sum \text{occ-util}(\delta_i)$.
- A *rational chain* δ is a chain such that there does not exist a δ' such that $\text{utility}(\delta') > \text{utility}(\delta)$.
- A *rational plan* ρ is then the sequence of activities associated with the occurrences in a rational chain.

Agents, Hosts, and Message Actions. A small theory weakly defining agents, hosts, messages, and message activities is also incorporated on top of PSL. This provides a shared ontology defining a base by which an agent may communicate about, recognize, and utilize communication resources.

Agent Beliefs. A first order theory of belief based on k -accessibility between occurrences is also included in the theory. It defines KD (see e.g. [Fagin *et al.*, 1995]), the logic of consistent belief, within the fluent space of PSL. This enables the expression of agent interfaces in the form of doxastic activity effects, as well as an agent's communication goals. Belief, and specifically KD , was chosen due to its relatively weak commitments on agent reasoning abilities and trust.

4 Example: Calendar Agent Axiomatized

Agents and communication resources may then describe their interfaces and abilities, e.g. the base computer of Section 2:

$$\begin{aligned} &\forall o, a, s, m \cdot \text{occurrence_of}(o, a) \wedge \text{legal}(o) \\ &\quad \wedge \text{message-activity}(a, s, \text{SYNC}, m) \supset \\ &\quad [\forall c \cdot \text{holds}(\text{contents}(m, c), o) \supset \\ &\quad \quad \text{holds}(\text{believes}(\text{SYNC}, c), o)]. \end{aligned}$$

The cellular and wireless interfaces may be described as:

$$\begin{aligned} &\forall o, a, s, d, m \cdot \text{occurrence_of}(o, a) \wedge \text{legal}(o) \wedge \\ &\quad \text{sat-msg}(a, s, d, m) \supset \text{message-activity}(a, s, d, m) \wedge \\ &\quad \quad \text{occ-util-summ}(o, -0.5). \\ &\forall o, a, s, d, m \cdot \text{occurrence_of}(o, a) \wedge \text{legal}(o) \wedge \\ &\quad \text{radio-msg}(a, s, d, m) \supset \text{prior}(\text{in-range}(s, d), o) \wedge \\ &\quad \quad \text{message-activity}(a, s, d, m) \wedge \text{occ-util-summ}(o, -0.1). \end{aligned}$$

The agent's internal communication activities would be:

$$\begin{aligned} &\text{activity}(\text{defer}). \\ &\forall o, m, d, n \cdot \text{occurrence_of}(o, \text{attach}(m, \text{update}(d, n))) \wedge \\ &\quad \text{legal}(o) \supset \wedge \text{holds}(\text{contents}(m, \text{update}(d, n)), o). \end{aligned}$$

The agent's behavior may then be captured as the time-dependent utility of delivering updates to the base computer:

$$\begin{aligned} &\forall o, c, d, n \cdot \text{holds}(\text{current-day}(c), o) \wedge \\ &\quad \text{holds}(\text{believes}(\text{SYNC}, \text{update}(d, n)), o) \supset \\ &\quad \quad \text{occ-util-summ}(o, 1 - ((d - c)/7)). \end{aligned}$$

With the addition of an episode delimiter defined as occurrences of *defer* or *message-activity*, supporting axioms not listed above such as the definition of *update*, and closure axioms, a plan based on these definitions will:

- Deliver any update if a wireless connection exists.
- Use the satellite interface if warranted by an update or some collection of updates. For example, any update concerning the next 3.5 days will be posted immediately.
- Otherwise defer until new updates are posted or conditions change, e.g. wireless contact is made.

5 Summary and Future Work

This poster presents an approach to efficiently performing communication tasks by planning on the costs and capabilities of advertised resources. In contrast with power-aware networking, this work supports application-layer preferences and mixed-infrastructure settings. Although similar to research on communication planning via epistemic goals, this work acknowledges that communicative acts have costs. In addition, this framework is oriented toward a service-based system, enabling agents to reason on internal as well as external capabilities. This work also introduces an application area of web services, along with a formalization in the situation calculus and a principled, practical representational differentiation between mental and world activity effects.

Future work includes algorithms and protocols for exchanging descriptions; accounting for the unpredictable nature of mobile networks in composing a plan; defining subclasses of the planning problem amenable to pragmatically feasible computation; techniques for integrating heterogeneous and untrusted costs; and the knowledge acquisition task of evaluating preconditions involving unknown terminology.

More information, details, and citations to related work are available at <http://edge.cs.drexel.edu/services/>.

References

- [Fagin *et al.*, 1995] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge, Mass., 1995.
- [Grüninger and Menzel, 2003] M. Grüninger and C. Menzel. The Process Specification Language: Theory and applications. *AAAI AI Magazine*, 24(3):63–74, 2003.
- [Kopena *et al.*, 2005] J. Kopena, V. Cicirello, M. Peysakhov, K. Malfettone, A. Mroczkowski, G. Naik, E. Sultanik, M. Kam, and W. Regli. Network awareness and the Philadelphia-Area Urban Wireless Network Testbed. In *AAAI Spring Symp. on AI for Homeland Security*, 2005.
- [Vahdat *et al.*, 2000] Amin Vahdat, Alvin Lebeck, and Carla Schlatter Ellis. Every joule is precious: The case for revisiting operating system design for energy efficiency. In *9th ACM SIGOPS European Workshop*, September 2000.
- [Xylomenos *et al.*, 2001] G. Xylomenos, G.C. Polyzos, P. Mahonen, and M. Saaranen. TCP performance issues over wireless links. *IEEE Comm.*, 39(4):52–58, 2001.

Trust No One: Evaluating Trust-based Filtering for Recommenders

John O'Donovan, Barry Smyth

Adaptive Information Cluster
Department of Computer Science
University College Dublin
Belfield, Dublin 4, Ireland

john.odonovan@ucd.ie, barry.smyth@ucd.ie

Abstract

To be successful recommender systems must gain the trust of users. To do this they must demonstrate their ability to make reliable predictions. We argue that collaborative filtering recommendation algorithms can benefit from explicit models of trust to inform their predictions. We present one such model of trust along with a cost-benefit analysis that focuses on the classical trade-off that exists between recommendation coverage and prediction accuracy.

1 Introduction

Recommender systems have been developed as a solution to the well documented information overload problem. [Resnick *et al.*, 1994], [Breese *et al.*, 1998]. These systems employ techniques from user profiling, machine learning and information filtering to produce individual recommendations of items to suit users' requirements. Collaborative filtering (CF) recommenders operate on the assumption that similar users share similar tastes; recommendations are generated for a target user by analysing the rating histories of a set of suitable recommendation partners.

The traditional CF approach relies heavily on the similarity between the target user and its partner as a way to weight each partner's predictions [Resnick *et al.*, 1994]. In this paper we propose that, in addition, it is possible to model the trustworthiness of these partners, and to use this as another factor to influence their prediction contributions. Indeed the idea of explicitly modeling and using trust in filtering tasks is becoming increasingly popular. For example, [Golbeck and Hendler, 2004] presents a trust-based email filter, trust scores in this system are calculated through inference and propagation, of the form $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \Rightarrow C)$, where A , B and C are users with interpersonal trust scores. The Trust-Mail application [Golbeck and Hendler, 2004] looks up an email sender in the reputation/trust network, and provides an inline rating for each mail. These trust values can tell a user if a mail is important or unimportant. Trust values in this system can be defined with respect to a certain topic, or on a general level, in a similar manner to work in [O'Donovan and Smyth, 2005a] and [O'Donovan and Smyth, 2005b].

[Avesani *et al.*, 2004] describe a trust-based recommender system in the skiing domain. However these approaches rely on models of trust that are built from the direct feedback of users; in short, individual users are expected to indicate those partners that they place the most trust in and a trust model is generated from the resulting graph of relationships.

[Massa and Bhattacharjee, 2004] build a trust model directly from explicit user-provided trust ratings. This work is carried out using the popular *epinions.com* service. *Epinions.com* is a web site that allows users to review various items (cars, books, music, etc.). In addition they can assign a trust rating to reviewers based on the degree to which they have found them to be helpful and reliable in the past. [Massa and Bhattacharjee, 2004] argue that this trust data can be extracted and used as part of the recommendation process, especially as a means to relieve the sparsity problem (lack of overlapping user ratings) that has hampered traditional collaborative filtering techniques [O'Sullivan *et al.*, 2002]. [Massa and Bhattacharjee, 2004] argue that it is possible to compare users according to their degree of connectedness in the trust-graph encoded by *Epinions.com*, but do not show that this method of comparison maintains recommendation accuracy.

Our benchmark algorithm uses Resnick's standard prediction formula which is reproduced below as Equation 1; see also [Resnick *et al.*, 1994]. In this formula $c(i)$ is the rating to be predicted for item i in consumer profile c and $p(i)$ is the rating for item i by a producer profile p who has rated i . In addition, $\bar{c}$ and $\bar{p}$ refers to the mean ratings for c and p respectively. The weighting factor $sim(c, p)$ is a measure of the *similarity* between profiles c and p , which is traditionally calculated as Pearson's correlation coefficient.

$$c(i) = \bar{c} + \frac{\sum_{p \in P(i)} (p(i) - \bar{p}) sim(c, p)}{\sum_{p \in P(i)} |sim(c, p)|} \quad (1)$$

As we have seen above Resnick's prediction formula discounts the contribution of a partner's prediction according to its degree of similarity with the target user so that more similar partners have a large impact on the final ratings prediction.

We argue that there is another factor which might be used in conjunction with similarity to influence recommendation and prediction. We believe that the reliability of a partner

profile to deliver accurate recommendations in the past is another important factor, one that we refer to as the *trust*. Intuitively, if a profile has made lots of accurate predictions in the past, then they can be viewed as more trustworthy than another profile that has made many poor predictions. We describe a computational model of trust that can be generated unobtrusively, during the normal operation of a CF recommender system, by mining the recommendation histories of different recommendation partners. We re-evaluate work in [O'Donovan and Smyth, 2005b] which shows that trust-based methods can improve prediction accuracy when compared to existing CF approaches. However, we describe a more comprehensive cost-benefit analysis by considering three accuracy benefits against changes in recommendation coverage.

2 A Computational Model of Trust

Intuitively, if a recommendation partner (profile) has made many good predictions in the past, it can be viewed as more trustworthy than one with many poor predictions. The trust model [O'Donovan and Smyth, 2005b] is based on this idea. We differentiate between profiles generating recommendations (*producer profiles*) and those receiving recommendations (*consumer profiles*) in a particular recommendation session. To generate a predicted rating, $p(i)$, for item i for some consumer c , conventional CF systems draw on the services of a number of producer profiles, combining their individual recommendations according to some suitable function, such as Resnick's formula. (see Equation 1). Our trust model depends on whether these predicted ratings are correct relative to the true ratings of the consumer, $c(i)$; see Equation 2.

$$Correct(i, p, c) \Leftrightarrow |p(i) - c(i)| < \epsilon \quad (2)$$

2.1 Item-Level Trust

We define the *item-level* trust for each producer p with respect to a given profile item i to be the percentage of times that p has made a *correct* rating prediction for i across some set of consumers; see Equation 3. To do this we consider the rating that p alone predicts for i , for the consumer in question. We define the $RecSet(p)$ (Equation 4) to be the total set of rating predictions that p has made; each (r_k, i_k) refers to a rating prediction, r_k that p has made for item i_k . Similarly, $CorrSet(p)$ is the subset of these ratings that are considered to be correct; Equation 5.

$$Trust^I(p, i) = \frac{|\{(r_k, i_k) \in CorrSet(p) : i_k = i\}|}{|\{(r_k, i_k) \in RecSet(p) : i_k = i\}|} \quad (3)$$

$$RecSet(p) = \{(r_1, i_1), \dots, (r_n, i_n)\} \quad (4)$$

$$CorrSet(p) = \{(c_k, i_k) \in RecSet(p) : Correct(i_k, p, c_k)\} \quad (5)$$

Thus, the trust of p in relation to item i is a measure of how often p 's predicted ratings for i have been considered correct in the past. This information can be accumulated during the normal course of operation of a CF recommender system in a variety of ways. For example, users could be asked their

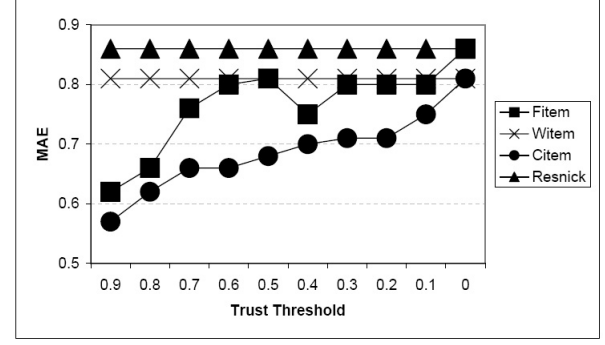


Figure 1: Recommendation Error.

opinions of the predicted ratings, or this might be confirmed by evaluating the user's actions on the basis of the rating predictions; if a user buys a highly rated item then we might assume that the high rating was justified.

2.2 Trust-Based Recommendation

We incorporate our trust-model into CF by modifying the standard Resnick prediction algorithm in 3 ways to produce 3 different trust-based variations. Resnick's standard prediction formula is given in Equation 1. Normally it uses the similarity between the target user profile and each recommendation partner profile to weight their prediction contributions, shown as $sim(c, p)$ in Equation 1; Equation 6 shows our modifications to this standard equation by adding the $w(c, p, i)$ weighting term. Our first variation (*WItem*) modifies this so that the weighting term is a combination of item trust and profile similarity; we use the harmonic mean of trust and similarity. The *FItem* approach differs in that it uses profile similarity as the weight factor, but filters out profiles that fall below a given trust level for the target item prior to recommendation. Finally, the *CItem* approach uses the obvious combination of *WItem* and *FItem*.

$$r(i) = \bar{r} + \frac{\sum_{p \in P(i)} (p(i) - \bar{p})w(c, p, i)}{\sum_{p \in P(i)} |w(c, p, i)|} \quad (6)$$

3 Evaluation

For the following evaluation we use the 943 profiles from the MovieLens data-set, split into 80% as training profiles and the remaining 20% as test profiles. During training we use a leave-one-out approach to build our trust model over the training profiles. Briefly, each training profile is used as a consumer with the remaining acting as producers. Item-level trust values are computed on the basis of the correctness or otherwise of the producer predictions. During testing, we evaluate the predictions of the training profiles for each of the items in the test profiles using our 4 basic algorithms (*Resnick*, *WItem*, *FItem*, *CItem*).

In this evaluation we are especially interested in the trade-off between the coverage of a recommender (the percentage of items that a rating can be predicted for) (Figure 2) and

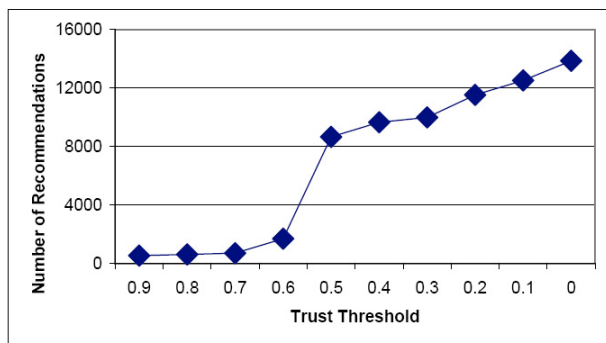


Figure 2: Recommendation Coverage.

the error over these predictions (Figure 1). The error graph shows a positive response to error for the trust-based methods, especially those that employ trust-based filtering and in particular for the higher trust-levels. For example, at a trust level of 0.9 only those profiles that have previously been correct 90% or more of the time that they have been called upon to rate an item, are included as recommendation partners for the filter-based approaches (*FItem* and *CItem*). And for these approaches we see significant error reductions of up to 57% compared to the baseline Resnick. However, the coverage results indicate that these error reductions come at a cost. In particular, the minimal error rates at the highest trust thresholds reduce coverage by over 90%, which is unlikely to be acceptable in most recommendation scenarios. However, for trust thresholds below 0.5 we get significant error reductions while preserving coverage to a reasonable degree. In particular, the error for *CItem* is seen to drop most rapidly up to a trust threshold of 0.2, at which point it offers 85% coverage and an error of 0.71; Resnick's error is 22% higher than this.

4 Conclusions

We believe that computational models of trust can improve the effectiveness of recommender systems. We have shown that by integrating an item-level model of trust into standard collaborative filtering we can increase accuracy by up to 57% by using only the top 1% most trustworthy profiles as recommendation partners. While this benefit comes at a significant coverage cost, more reasonable coverage can be achieved with reduced error rates by less drastic filtering thresholds. In addition to improving prediction accuracy, we believe that this trust-based approach may make recommenders more robust to attack by malicious users, as discussed in [O'Mahony *et al.*, 2002], [Levien, 2003] and [Kushmerick, 2002]. This is a matter that we will investigate as part of future work.

5 Acknowledgments

This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361

References

[Avesani *et al.*, 2004] Paolo Avesani, Paolo Massa, and Roberto Tiella. Moleskiing: a trust-aware decentralized

recommender system. *1st Workshop on Friend of a Friend, Social Networking and the Semantic Web. Galway, Ireland*, 2004.

[Breese *et al.*, 1998] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Gregory F. Cooper and Serafin Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco, July 24–26 1998. Morgan Kaufmann.

[Golbeck and Hendler, 2004] Jennifer Golbeck and James Hendler. Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *Proceedings of EKAW'04*, pages LNAI 2416, p. 278 ff., 2004.

[Kushmerick, 2002] N. Kushmerick. Robustness analyses of instance-based collaborative recommendation. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Proceedings of the European Conference on Machine Learning, Helsinki, Finland.*, volume 2430, pages 232–244. Lecture Notes in Computer Science Springer-Verlag Heidelberg, 2002.

[Levien, 2003] Raph Levien. Attack resistant trust metrics. *Ph.D Thesis, UC Berkeley*, 2003.

[Massa and Bhattacharjee, 2004] Paolo Massa and Bobby Bhattacharjee. Using trust in recommender systems: an experimental analysis. *Proceedings of 2nd International Conference on Trust Managment, Oxford, England*, 2004.

[O'Donovan and Smyth, 2005a] John O'Donovan and Barry Smyth. Eliciting trust values from recommendation errors. In *Proceedings of the 18th International FLAIRS Conference*. AAAI Press, 2005.

[O'Donovan and Smyth, 2005b] John O'Donovan and Barry Smyth. Trust in recommender systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 167–174. ACM Press, 2005.

[O'Mahony *et al.*, 2002] Michael P. O'Mahony, Neil Hurley, and Guenole C. M. Silvestre. An attack on collaborative filtering. In *Proceedings of the 13th Int. Conf. on Database and Expert Systems Applications*, pages 494–503. Springer-Verlag, 2002.

[O'Sullivan *et al.*, 2002] Derry O'Sullivan, David C. Wilson, and Barry Smyth. Improving case-based recommendation: A collaborative filtering approach. In *Proceedings of the Sixth European Conference on Case Based Reasoning.*, pages LNAI 2416, p. 278 ff., 2002.

[Resnick *et al.*, 1994] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM CSCW'94 Conference on Computer-Supported Cooperative Work, Sharing Information and Creating Meaning*, pages 175–186, 1994.

Inter-Agent Communication: A Cost-Reduction Approach Using an Autonomous Mobile Mailbox

Armin Stranjak¹, Igor Cavrak², Mario Zagar²

¹Rolls-Royce plc, Strategic Research Centre
SINA-28, PO BOX 31, Derby DE24 8BJ, UK
armin.stranjak@rolls-royce.com

²Faculty of Electrical Engineering and Computing
Unska 3, HR-10000 Zagreb, Croatia
{igor.cavrak,mario.zagar}@fer.hr

Abstract

In this paper, we consider a mobile mailbox communication scheme to reduce inter-agent communication costs. We employ a mailbox mobility strategy based on the ability of the mailbox to predict variations in inbound message rates and to migrate, if necessary, to a potentially better position in the network.

1 Introduction

Effective and efficient communication between mobile agents requires a different approach than the one employed in traditional distributed system communication, as difficulties may be encountered when tracking the location of the target agent, together with message delivery reliability issues. In [Feng *et al.*, 2001] [Cao *et al.*, 2002a], a non-autonomous mailbox based message delivery protocol is proposed. As the mailbox can travel independently of its owner, there is the possibility to reduce the costs of message delivery to an agent but the mailbox does not decide about its own migration.

In this paper, we consider performance of autonomous mailbox agent whose migration strategy is influenced by reducing the costs of message delivery from senders to a receiver or target agent, which is the owner of the mailbox [Cavrak *et al.*, 2004]. Cost reduction mainly relies on lower mailbox mobility rates and in choosing optimal relay network positions, cutting the inbound message delivery costs.

2 Cost Model

The communication network is designed as a fully connected, undirected, weighted graph. We employ the assumption that the costs associated with messages transferred from senders to the receiver using the mailbox depend on the following:

Cost of message transfer from sender to mailbox in the time interval $[t_i - T_e, t_i]$:

$$C_{trms}(t_i) = \sum_{p=1}^m c_{i_p b} \cdot \lambda_p(t_i) \quad (1)$$

where $c_{i_p b}$ is associated with the cost of transfer of one message between host h_i (agent a_p location) and host h_b (mailbox location), while $\lambda_p(t_i)$ is the amount of messages sent by agent a_p in the time period considered.

Cost of mailbox registration and deregistration procedure in the synchronised push model when mailbox moves from host h_b to host h_{b^*} :

$$C_{reg} = Q \cdot \sum_{p=1}^m (2 \cdot c_{i_p b} + c_{i_p b^*}) \quad (2)$$

where Q is a ratio between the delivery cost of one control and one data message ($Q < 1$). Whenever mailbox changes location it requests and waits for synchronisation release confirmation (2 messages) and sends resynchronisation request on arrival (1 message).

Cost of mailbox migration depends on the number of messages (collected in time interval $[T_1, T_2]$) present in the mailbox's internal buffer at the moment of migration from host h_b to host h_{b^*} :

$$C_{mig} = c_{bb^*} \cdot \sum_{p=1}^m \sum_{t=T_1}^{T_2} \lambda_p(t) \quad (3)$$

Cost of mailbox position update at corresponding Home object:

$$C_{hmreg} = Q \cdot c_{b^* h} \quad (4)$$

where $c_{b^* h}$ is the cost incurred by sending one message from the future mailbox's host h_{b^*} to host h_h where the Home object is placed.

Cost of message transfer from mailbox to the receiver associated with the cost of pulling the collected messages within time interval $[T_1, T_2]$ by the receiver agent:

$$C_{pull} = c_{br} \cdot \sum_{p=1}^m \sum_{t=T_1}^{T_2} \lambda_p(t) \quad (5)$$

where c_{br} is the cost of transfer of one message between host h_b where the mailbox is located, and the host h_r where the receiver is located.

3 Mailbox Migration Strategy and Model

The mailbox holds the data that include network topology and communication costs, the number of message sources and their network locations, and the set of hosts where the receiver agent will be located. Reducing the cost of communication by planning optimal mailbox positions during the course of system execution is the main task.

At simulated time t_i , a mailbox does the following: (a) predicts message rate vector $\lambda(t_i + T_e)$ for all senders using extrapolated straight line fit method, (b) based on those predictions, calculates $C_{trms}(t_i + T_e)$ for all possible mailbox posi-

tions within the network and finds future position that will incur the minimal cost. (c) based on cost calculations, migrates to the next proposed host or remains on the current one.

4 Simulation and Results

A simulated mobile agent system consists of n hosts and m agents that are distributed randomly in the fully connected network. Message rate functions of sender agents are randomly picked from the set of functions such as *periodic step function*, *constant function* and *periodic triangular function* superimposed with positive or negative random errors. *Efficiency of the mailbox* is defined as a ratio between total costs occurred in the system with mobile mailbox and total costs in the system without the mailbox.

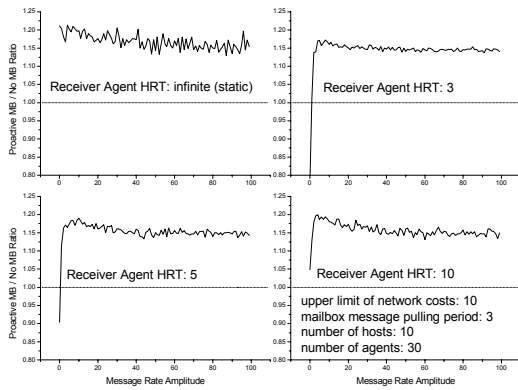


Figure 1: Efficiency for different message rate amplitudes (heterogeneous network)

Figure 1 shows the dependency between the efficiency of the mailbox and the message rate amplitude of sender agents in a heterogeneous network. Increasing the network heterogeneity, the usability of the mailbox is still justified even with increased host residence time (HRT) of the receiver agent, if the message rate amplitude is low. The heterogeneous network will decrease the influence of a two-stage communication effect as the ratio between paths used for message delivery from senders to the receiver agent in the system with the mailbox when compared with the system without the mailbox will be lower than the system with the homogeneous network.

Figure 2 shows the dependency between the efficiency of the mailbox and the number of hosts in the network. The number of hosts has a significant influence on the efficiency of the mailbox. In a network with fewer hosts, the mailbox is not provided with enough opportunities to mitigate the effect induced by a two-stage communication process and that is the reason the ratio is above 1. Increasing the number of hosts, the mailbox is offered more alternatives to deliver messages with lower costs. The higher number of sender agents will increase the dispersion of message sources in the network which will decrease the likelihood for the mailbox to be “too far” from agents. It will result in a more even

message transfer cost over various positions of the mailbox. The efficiency of the mailbox will be less variable as the number of sender agents increases and the graph will be smoother. The number of agents does not significantly influence the efficiency of the mailbox as both the mailbox and the receiver agent need to synchronise with them whenever the migration occurs, and those costs do not affect the efficiency ratio.

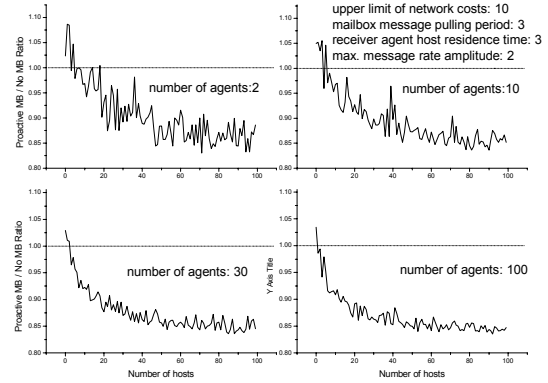


Figure 2: Efficiency for different numbers of hosts in the network

5 Conclusion

In this paper, we investigated performance of a novel approach for message-delivery cost-reduction in mobile agent systems. The approach uses an autonomous mobile mailbox acting as an intermediary message relay station. Intelligent behaviour of the mailbox relies on its ability to predict incoming message rates on-the-fly and decide about potential migration accordingly. Its migration strategy is motivated by a desire to reduce the costs of message delivery to the agent to which it belongs. Possible cost savings are achieved through lower mailbox mobility rates as well as deciding about optimal mailbox position within the network.

References

- [Feng *et al.*, 2001] X. Feng, J. Cao, J. Lu, and H. Chan, “An Efficient Mailbox-Based Algorithm for Message Delivery in Mobile Agent Systems”, *5th IEEE International Conference on Mobile Agents*, Atlanta, USA, Dec. 2001.
- [Cao *et al.*, 2002a] J. Cao, X. Feng, J. Lu, and S. K. Das, “Design of Flexible and Reliable Mobile Agent Communication Protocols”, *22nd IEEE International Conference on Distributed Computing Systems*, Vienna, Austria, July, 2002.
- [Cao *et al.*, 2002b] J. Cao, X. Feng, J. Lu, and S. K. Das, “Reliable Message Delivery for Mobile Agents: Push or Pull”, *Proceedings of the 9th International Conference on Parallel and Distributed Systems*, Taiwan, China, 2002.
- [Cavrak *et al.*, 2004] I. Cavrak, A. Stranjak and M. Zagar, “Autonomous Mobile Mailbox Model for Communication Cost Reduction”, *26th International Conference on Information Technology Interfaces, ITI'04*, Dubrovnik, Croatia, June 2004.

Coalitional Games in Open Anonymous Environments

Makoto Yokoo[†], Vincent Conitzer[‡], Tuomas Sandholm[‡], Naoki Ohta[†], Atsushi Iwasaki[†]

[†]Faculty of Information Science

and Electrical Engineering

Kyushu University

6-10-1 Hakozaki, Higashi-ku,

Fukuoka, 812-8581 Japan

yokoo/ohta/iwasaki@is.kyushu-u.ac.jp

[‡]Computer Science Department

Carnegie Mellon University

5000 Forbes Avenue

Pittsburgh, PA 15213 USA

conitzer/sandholm@cs.cmu.edu

1 Introduction

Coalition formation is a key capability in automated negotiation among self-interested agents. A coalition of agents can sometimes accomplish things that the individual agents cannot, or can do things more efficiently. To make coalition formation successful, a key question that must be answered is how the gains from cooperation are to be distributed. *Coalitional game theory* provides a number of solution concepts for this. Some of these solution concepts have already been adopted in the multi-agent systems literature [Zlotkin and Rosenschein, 1994; Shehory and Kraus, 1998].

However, existing solution concepts have limitations when applied to open anonymous environments such as the Internet. In such environments, a single agent can use multiple identifiers (or *false names*), pretending to be multiple agents, and distribute its ability (skills) among these identifiers. Alternatively, multiple agents can collude and pretend to be a single agent that combines all of their skills. Furthermore, an agent might try to hide some of its skills.

These manipulations are virtually impossible to detect in open anonymous environments, and have thus become an issue in such environments specifically. That is also the reason why the gamut of these manipulations has not received much research attention previously. In this paper, we develop a new solution concept for coalitional games called the *anonymity-proof core*, which is robust to the manipulations described above. We show that the anonymity-proof core is characterized by certain axiomatic conditions (including that an agent does not have an incentive to use the basic manipulations mentioned above).

2 Model

Traditionally, value division in coalition formation is studied in *characteristic function games*, where each potential coalition (i.e., each subset X of the agents) has a value $w(X)$ that it can obtain. This assumes that utility is transferable (e.g., utility can be transferred using side payments), and that a coalition's value is independent of what non-members of the coalition do. The characteristic function by itself does not give us sufficient information to assess what manipulations may be performed by agents in an open anonymous environment. Thus, instead of defining the characteristic function over agents, we define it over *skills* that the agents possess.

Definition 1 (skills and agents) Assume the set of all possible skills is T . Each agent i has one or multiple skills $S_i \subset T$. For simplicity, we assume each skill is unique, that is, $\forall i \neq j, S_i \cap S_j = \emptyset$ holds.

Definition 2 (characteristic function defined over skills)

A characteristic function $v : 2^T \rightarrow \mathbb{R}$ assigns a value to each set of skills.

We will denote by w the characteristic function defined over agents, and by v the characteristic function defined over skills. For a given set of agents X , let $S_X = \bigcup_{i \in X} S_i$. Then, we have $w(X) = v(S_X)$. The characteristic function over skills is a more fine-grained representation than the characteristic function over agents.

We assume the following three types of manipulation (or any combination of them) are possible for agents.

hiding skills: If agent i has a set of skills S_i , for any $S'_i \subseteq S_i$, it can declare that it has only S'_i .

false-name: Agent i can use multiple identifiers and declare that each identifier has a subset of S_i .

collusion Multiple agents can collude and pretend to be a single agent, who has the union of their skills.

3 Manipulability of Traditional Solution Concepts

In this section, we show that a well-known (perhaps the best known) solution concept called the *core* [Gillies, 1953; von Neumann and Morgenstein, 1947] is vulnerable against these manipulations. This is true for other solution concepts such as *Shapley value* [Shapley, 1953], *least-core*, or *nucleolus*.

Definition 3 (core) Given a set of agents W , an outcome, that is, a value division $c^W = (c_1^W, c_2^W, \dots)$ among agents is in the core if the following two conditions hold:

1. $\forall X \subset W, \sum_{i \in X} c_i^W \geq w(X)$,
2. $\sum_{i \in W} c_i^W = w(W)$.

The first condition is called the *non-blocking* condition: if this condition does not hold for some set of agents X , then the agents in X have an incentive to collectively deviate from the mechanism and to divide $w(X)$ among themselves. The second condition is called the *feasibility* condition: if $\sum_{i \in W} c_i^W > w(W)$, this outcome is infeasible.

Due to the space limitation, we only show an example where the core is manipulable by a collusion.

Example 1 Let there be three skills a, b , and c . Let the characteristic function over skills be as follows.

- $v(\{a, b\}) = v(\{a, c\}) = v(\{a, b, c\}) = 1$,
- $v(\{a\}) = v(\{b\}) = v(\{c\}) = v(\{b, c\}) = 0$.

Let there be three agents 1, 2, and 3 with skills a, b , and c , respectively. Then, the characteristic function over agents is as follows.

- $w(\{1, 2\}) = w(\{1, 3\}) = w(\{1, 2, 3\}) = 1$,
- $w(\{1\}) = w(\{2\}) = w(\{3\}) = w(\{2, 3\}) = 0$.

In this example, there is only one outcome in the core, namely outcome $(1, 0, 0)$. This is because if agent 2 (or 3) obtains any value, then the non-blocking condition is violated because agent 1 and agent 3 (or 2) have an incentive to deviate from the mechanism and form their own coalition. We can see that since the skill b and c are completely substitutable, the agents who have these skills have no bargaining power. The least core and the nucleolus also give this outcome.

Now, let us assume that agent 2 and 3 collude and pretend to be a single agent $2'$, who has b and c .

Then, the characteristic function over agents is as follows.

- $w(\{1\}) = w(\{2'\}) = 0$,
- $w(\{1, 2'\}) = 1$.

Now, agent 1 and $2'$ are symmetric and have the same bargaining power. Then, if we use the least core or the nucleolus, each agent receives $1/2$. Thus, agent 2 and 3 can increase their value division using a collusion.

4 Anonymity-Proof Core

We develop a new solution concept for our setting which we call *anonymity-proof core*. As we will show, the anonymity-proof core can be characterized by certain axiomatic conditions.

We require that the outcome function π is *anonymous*, that is, the payoff to an agent does not depend on the identifiers of the agents; it depends only on the skills of the agent and the distribution of skills over other agents.

More specifically, given an agent i and a set of other agents Y , let S_i be the set of skills that agent i has, and let $SS_Y = \{S_j \mid j \in Y\}$, where S_j is the set of skills that agent j has. Then, the outcome function $\pi(S_i, SS_Y)$ takes S_i and SS_Y as arguments, and returns the payoff to agent i , when agent i declares its skills as S_i and the other agents declare their skills as SS_Y .

Let the set of agents who joined the mechanism be W , and let the profile of the skills that the agents declared be $k = (k_1, k_2, \dots)$, where k_i is the set of skills that agent i declared. Let $S_X = \bigcup_{i \in X} k_i$, that is, S_X is the union of the skills declared by a set of agents X ; let $S = S_W$; and let $SS_X = \{k_i \mid i \in X\}$. Also, let $SS_{\sim i} = \{k_1, \dots, k_{i-1}, k_{i+1}, \dots\}$, that is, a set, each of whose elements is the set of skills corresponding to agent j ($j \neq i$).

We now give six axiomatic conditions that the outcome function π should satisfy.

1. The outcome function π is anonymous.
2. π is never blocked by any coalition, that is, $\forall k, \forall X \subseteq W, \sum_{i \in X} \pi(k_i, SS_{\sim i}) \geq v(S_X)$ holds.
3. π is always feasible and always distributes all of the value, that is, $\forall k, \sum_{i \in W} \pi(k_i, SS_{\sim i}) = v(S)$ holds.
4. π is robust against hiding skills, that is, $\forall S', S''$, where $S'' \subseteq S'$, $\forall SS, \pi(S'', SS) \leq \pi(S', SS)$ holds.
5. π is robust against false-name manipulations, that is, $\forall k, \forall X \subseteq W, Y = W \setminus X, \sum_{i \in X} \pi(k_i, SS_{\sim i}) \leq \pi(S_X, SS_Y)$ holds.
6. π is robust against collusions, that is, $\forall k, \forall X \subseteq W, Y = W \setminus X, \sum_{i \in X} \pi(k_i, SS_{\sim i}) \geq \pi(S_X, SS_Y)$ holds.

In order to define the anonymity-proof core, we first formally define the core for skills. For a set of skills $S = \{s_1, s_2, \dots\}$, we define a set of core outcomes for skills $Core(S)$ as follows.

Definition 4 (core for skills) $c^S = (c_{s_1}^S, c_{s_2}^S, \dots)$ is in $Core(S)$ if it satisfies the following two conditions.

- $\forall S' \subset S, \sum_{s_j \in S'} c_{s_j}^S \geq v(S')$,
- $\sum_{s_j \in S} c_{s_j}^S = v(S)$.

Now we are ready to define the anonymity-proof core.

Definition 5 (anonymity-proof core) We say the outcome function π_{ap} is in the anonymity-proof core if π_{ap} satisfies the following two conditions.

1. For any set of skills $S \subseteq T$, there exists a core outcome for S , that is, some $c^S = (c_{s_1}^S, c_{s_2}^S, \dots) \in Core(S)$, such that for any skill profile $k = (k_1, k_2, \dots)$ with $\bigcup_i k_i = S$, $\pi_{ap}(k_i, SS_{\sim i}) = \sum_{s_j \in k_i} c_{s_j}^S$ holds.
2. $\forall S', S''$, where $S'' \subseteq S'$, $\forall SS, \pi_{ap}(S'', SS) \leq \pi_{ap}(S', SS)$ holds.

The first condition says that for any set of skills reported by the agents, some outcome in the core for that set of skills should be used to distribute the value. The second condition says that an agent has no incentive to hide (some of) its skills.

The following theorems show that the anonymity-proof core is characterized by the six axiomatic conditions. We omit the proof due to space constraint.

Theorem 1 Any outcome function π_{ap} in the anonymity-proof core satisfies the six axioms.

Theorem 2 Any outcome function π that satisfies the six axioms is in the anonymity-proof core.

References

- [Gillies, 1953] Donald Gillies. *Some theorems on n-person games*. PhD thesis, Princeton University, Dept of Mathematics, 1953.
- [Shapley, 1953] Lloyd S. Shapley. A value for n-person games. In H Kuhn and A Tucker, eds, *Contributions to the Theory of Games*, vol 2 of *Annals of Mathematics Studies*, 28, 307–317. Princeton U Press.
- [Shehory and Kraus, 1998] Onn Shehory and Sarit Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1–2):165–200, 1998.
- [von Neumann and Morgenstein, 1947] John von Neumann and Oskar Morgenstein. *Theory of games and economic behavior*. Princeton University Press, 1947.
- [Zlotkin and Rosenschein, 1994] Gilad Zlotkin and Jeffrey S Rosenschein. Coalition, cryptography and stability: Mechanisms for coalition formation in task oriented domains. In *AAAI*, pages 432–437, 1994.

Value-Centric Trust Model with Improved Familiarity Measurement

Jie Zhang and Ali. A. Ghorbani

Faculty of Computer Science
University of New Brunswick
Fredericton, NB, E3B 5A3, Canada
Email: {jie.zhang, ghorbani}@unb.ca

Abstract

Formalization of familiarity contributes to formalization of trust through a value-centric trust model. However, familiarity was assumed to be the similarity of values (fixed for two agents), and stability of the trust model was relatively low. To increase the stability, we propose an improved familiarity measurement. Experiments are carried out to compare the stability of the trust model with the improved familiarity measurement and with the fixed familiarity value. It is observed that the stability is increased by 33.47% through the improved familiarity measurement.

1 Introduction

Trust has always been bundled with familiarity to become a popular topic in the fields of psychology, sociology, and computer science. The relationship between trust and familiarity has been further clarified through the value-centric trust model proposed by Carter and Ghorbani [Carter and Ghorbani, 2004]. The new model proposes that trust is a combination of self-esteem, reputation, and familiarity within a multi-agent system (MAS) context. Trust is also multidimensional in that it can be facilitated through familiarity. Familiarity was assumed to be the similarity of values. In consequence, stability of the trust model is relatively low. However, people in nature prefer relatively stable societies. To increase stability of the trust model, we propose an improved familiarity measurement by exploring a variety of human factors that may affect the feeling of familiarity.

The rest of the paper is organized as follows. Section 2 describes the four major factors affecting familiarity in agent societies. The way of measuring familiarity is proposed in Section 3. Experimental results are presented and discussed in Section 4. Finally, the conclusions of the present study are presented in Section 5.

2 Factors Affecting Familiarity

Familiarity is affected by four major factors: prior experience, repeated exposure, level of processing, and forgetting rate [Zhang and Ghorbani, 2004]. Prior experience is determined by knowledge of similar agents in the agent society.

Repeated exposure is represented by how many transactions are established between the two agents. Level of processing is determined by the quantity of widgets in each transaction. Forgetting rate is calculated by the interval between the last transaction and the current transaction, and the factor of the agent society.

3 Familiarity Measurement

For an agent society A with n agents, $A = \{a_1, a_2, \dots, a_n\}$, let $F(a_i, a_j)$ and $S(a_i, a_j)$ represent the familiarity and similarity between agents a_i and a_j , respectively. The initial familiarity value that the agent a_i has with the agent a_j can be calculated through the formula as follows:

$$F_0(a_i, a_j) = \max_{k=1}^n F(a_j, a_k) S(a_i, a_k) \quad (k \neq i \neq j) \quad (1)$$

The value of familiarity can be calculated from the knowledge that the agent a_i has about the agent a_j as follows:

$$F_c(a_i, a_j) = \frac{2}{1 + e^{-K_c(a_i, a_j)}} - 1, \quad (2)$$

where $F_c(a_i, a_j)$ and $K_c(a_i, a_j)$ represent the familiarity value and the knowledge value that the agent a_i has from the perspective of the agent a_j before the current, c , transaction, respectively.

Since the familiarity value is affected by the previous level of processing and the forgetting rate, and it is determined by the agent's knowledge, a simple formula for updating the agent's knowledge may be as follows:

$$K_c(a_i, a_j) = K_p(a_i, a_j) + L_p(a_i, a_j) - R_p(a_i, a_j), \quad (3)$$

where $K_p(a_i, a_j)$ represents the knowledge value that agent a_i had about agent a_j before the previous transaction, $L_p(a_i, a_j)$ is the level of processing of agents a_i and a_j during the previous transaction, and $R_p(a_i, a_j)$ represents the forgetting value since the previous transaction. The initial knowledge value of agent a_i , $K_0(a_i, a_j)$ can be determined by Equations 1 and 2.

The formula to calculate the previous level of processing of the agents a_i and a_j may be calculated by:

$$L_p(a_i, a_j) = K_p(a_i, a_j)(1 - e^{-Q_p/l}), \quad (4)$$

where Q_p represents the quantity of widgets in the previous transaction and l represents the learning coefficient. The value of l may differ for different agent societies.

The forgetting value of agent a_i and agent a_j can be calculated as follows:

$$R_p(a_i, a_j) = K_p(a_i, a_j)(2 - e^{-Q_p/l})(1 - e^{-\Delta t_p/m}), \quad (5)$$

where m represents the memory coefficient. Δt_p represents the time difference between the current transaction and the previous transaction of agents a_i and a_j .

4 Analysis of Stability

The stability of the model is considered with respect to trustworthiness ranking. Both the two kinds of familiarity measurements, improved familiarity measurement and fixed familiarity value calculated by the similarity of two agents, are implemented and embedded in the trust model. For later use, TMIFM is defined as the trust model with improved familiarity measurement, and TMFFV is defined as the trust model with fixed familiarity value. Within this work, stability is connected to the idea of ranking. The stability refers to the degree of change of rankings of sellers. Thus, Stability is measured through an examination of the average variance of the selling agents' ranks on a daily basis.

Table 1: Comparison of Stability of TMIFM and TMFFV

Test #	TMIFM	TMFFV	Percentage Difference
1	3.92	5.77	32.06%
2	5.61	7.91	29.08%
3	6.11	10.12	39.62%
4	5.36	8.62	37.82%
5	4.00	4.96	19.35%
6	3.94	5.61	29.77%
7	4.39	7.75	43.35%
8	5.11	9.51	46.27%
9	4.47	6.10	26.72%
10	6.35	7.00	9.29%
Average	4.73	7.11	33.47%

The comparative stability of TMIFM and TMFFV is presented in Table 1. On average, the average variance of TMIFM is 33.47% lower than that of TMFFV, which means that the former is more stable than the latter. The result can be further illustrated by analyzing the change of rank of any given agent as shown in Figure 1. From this figure, it is obvious that the variance of the rank produced by TMIFM is lower than that produced by TMFFV. Therefore, TMIFM is more stable than TMFFV.

The reason that TMIFM has higher stability can be explained by analyzing two phenomena in both of the two trust models. One phenomenon is that agents are pushed faster to the right spot that they should be on in TMIFM than in TMFFV, which can be seen from Figure 1. The agent in TMIFM nearly reaches the average line earlier (approximately on day 15) than in TMFFV (approximately on day 40). This happens because that the improved familiarity measurement increases the speed of pushing the agent to the right

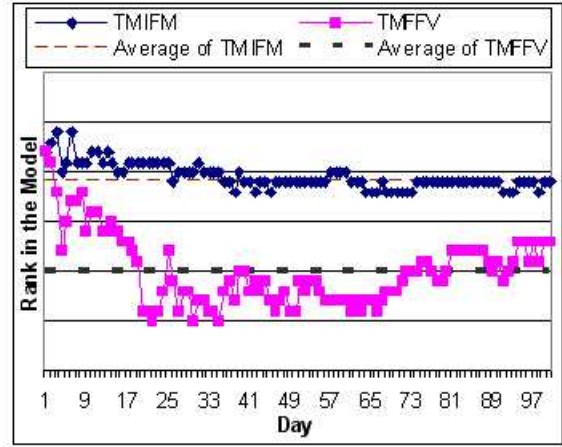


Figure 1: Change of Rank of Any Given Agent

spot. Another phenomenon is that once agents have been given a spot, they remain close to that spot. This phenomenon can also be seen in Figure 1. From day 25 on, the rank of the agent in TMIFM stays close to the average line, whereas the rank of the agent in TMFFV keeps changing. This happens because the selling agents with higher/lower rank have more/less possibility of being selected to establish transactions with buying agents in both TMIFM and TMFFV.

5 Conclusions

We proposed the improved familiarity measurement by exploring the factors mainly affecting familiarity. The four factors include prior experience, repeated exposure, level of processing, and forgetting rate. We then devised a convenient way to measure and update familiarity value. The improved familiarity measurement has been integrated into the value-centric model. Experiments were carried out to compare the stability of the trust model with the improved familiarity measurement and with the fixed familiarity value. Experimental results show that the stability has been increased by 33.47% through the improved familiarity measurement.

6 Acknowledgments

This work was funded by the National Science and Engineering Research Council of Canada (NSERC) to Dr. Ali A. Ghorbani through grant RGPIN 227441.

References

- [Carter and Ghorbani, 2004] Jonathan Carter and Ali A. Ghorbani. Towards a formalization of value-centric trust in agent societies. *Journal of Web Intelligence and Agent Systems*, 2(3):167–184, 2004.
- [Zhang and Ghorbani, 2004] Jie Zhang and Ali A. Ghorbani. Familiarity and trust: Measuring familiarity with a web site. In *Proceedings of the 2nd Annual Conference on Privacy, Trust and Security (PST 2004)*, pages 23–28, October 2004.

POSTER PAPERS

NATURAL LANGUAGE AND USER INTERFACES

Towards More Intelligent Mobile Search

Karen Church, Mark T. Keane and Barry Smyth

Adaptive Information Cluster,
Smart Media Institute, Department of Computer Science,
University College Dublin, Belfield, Dublin 4, Ireland
{Karen.Church, Mark.Keane, Barry.Smyth}@ucd.ie

Abstract

As the mobile Internet continues to grow there is an increasing need to provide users with effective search facilities. In this paper we argue that the standard Web search approach of providing snippet text alongside each result is not appropriate given the interface limitations of mobile devices. Instead we evaluate an alternative approach involving the use of related queries in place of snippet text for *result gisting*.

1 Introduction

The mobile world is growing at an astonishing rate. In 2004, the number of mobile users worldwide reached 1.52 billion, in Europe alone the number of users exceeded 340 million and the volume of pages viewed on the Mobile Web continues to rise [CellularOnline, 2004]. Five years has passed since the publication of the renowned WAP usability report, which detailed the various limitations of both WAP (as a means of mobile Web access) and mobile devices [Ramsay and Nielsen, 2000]. However, the Mobile Internet has since been successfully re-branded by leading mobile operators like Vodafone and O2, thus encouraging users to take advantage of next-generation mobile devices.

If we examine current mobile devices, we can divide them into three classes. At the lowest level we have standard WAP enabled phones. Although the screen real-estate of these devices is still very small, most of them come equipped with high-resolution color interfaces and predictive text input capabilities - a significant improvement over 1st generation WAP-handsets. The second class of device includes next-generation smart phones with 3G capabilities. The majority of these devices encompass larger screens, higher-resolution interfaces, enhanced Web-browser support and multiple input options. For example, most of these phones support XHTML, (e.g. Nokia 6630, Siemens SX1) and some of the phones incorporate flip-out messaging keyboards, (e.g. Nokia 6820, Siemens SK65). The final class of mobile device includes PDA's, XDA's and other small handhelds. These devices have much larger screens, higher-resolution interfaces, integrated Web-browsers supporting standard HTML as well as a multitude of input options including full messaging keyboards and

stylus. So while previous incarnations of the Mobile Internet were not viewed as a success there is increasing optimism about the future of this important information medium. These new handsets, coupled with improved content, more attractive billing arrangements and greater bandwidth appear to be attracting more and more users to the Mobile Internet on a daily basis. This in turn is likely to lead to a need for improved search facilities, facilities that we argue need to be carefully adapted to the features of the Mobile Internet, its devices and the differing needs of its users

2 Characteristics of Mobile Search

We do acknowledge that even current mobile devices suffer from inherent limitations that can greatly effect the usability of many mobile applications [Ramsay and Nielsen, 2000]. This is largely due to the fact that very little effort has been made to cater for the unique requirements of the Mobile Internet. Such limitations can be especially problematic for mobile search applications. Small displays restrict the number of results that can be displayed on a single screen and input limitations impact the type of queries that are provided. Most mobile search engines offer only simple adaptations to standard Web search interfaces. For example, fewer results are returned per page and the *snippet* text associated with each result may be truncated or simply removed [Jones *et al.*, 2002]. We believe that more fundamental adaptations are necessary.

In particular we believe that displaying snippet text alongside each search result is not feasible in the mobile realm. However we also think that displaying just a title or a url to represent each search result does not provide the user with meaningful information about the context of a given result. Instead we propose the use of *related queries* in place of snippet text as a more economical form of result gisting. Economy of screen-space is a major concern in mobile search and the use of related queries can significantly increase the number of results displayed per screen on mobile devices. Moreover, in this paper we also show that the terms in these related queries can be more informative, when it comes to capturing the content of an associated page, than the page's snippet text. Similar work has also been carried out in [Jones *et al.*, 2004], where key phrases, automatically extracted from web pages are used to provide users with accurate, meaningful representations of search results on mobile devices.

3 Towards Intelligent Result Gisting

A personalized search engine called I-SPY, [Smyth *et al.*, 2003], can provide these related queries by tracking and recording past queries that have resulted in the selection of a given result page. I-SPY is a community-based meta search engine that records the search histories of communities of like-minded users [Smyth *et al.*, 2003; Forthcoming]. Each time a user selects a result, p , for a query, q , I-SPY updates a counter in a *hit-matrix* to reflect this selection. This information is used to adapt future result-lists for similar queries by promoting results that have been selected in the past.

One of I-SPY's distinguishing features is that it can recommend related queries alongside a search result. For each search result p , that is associated with some target query q , I-SPY can generate a set of related queries from those queries, $q_1, \dots, q_n$, that have also led to the selection of p [Smyth *et al.*, Forthcoming]. These related queries can also be ranked according to the likelihood that they match a user's current requirements. These ranking strategies are described in [Balfe and Smyth,]. Figure 1 illustrates these related queries in action. If we take, for example, the query 'MP3' we can see that the first search result, 'mp3.com' is associated with queries such as 'music' and 'charts'. These *related queries* can inform the user as to the context of the search result in question.

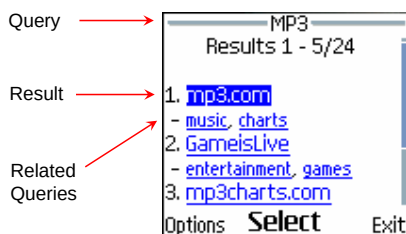


Figure 1: Query 'MP3' on a mobile phone

It is interesting to consider whether these related queries could be used as an alternative to snippet text to gist search results, especially given the interface limitations of mobile devices. Preliminary empirical evidence generated from a similar evaluation we carried out suggests that these related queries may be as informative as snippet text [Church *et al.*, 2005]. In our previous evaluation we used a smaller dataset and a different independent search engine, namely Google. In order to validate our earlier findings, we decided to carry out a more in-depth evaluation using a different independent search engine, namely HotBot and a larger dataset. The details of this evaluation are described in the following section.

4 Evaluation

The data for this evaluation was collected by mining the search logs of I-SPY. Specifically we collected a set of 684 result-pages that were selected by searchers in response to various queries. Each of these pages was associated with at least 2 related queries and a unit of snippet text generated by Google. The purpose of our experiment was to consider how well these related queries represented the page in question, relative to its snippet text. To do this we supposed that

the representativeness of a set of terms relative to some page could be measured by the position of the page in the result-list generated by some search engine when using these terms as a query. Hence, in our evaluation we transformed each set of related queries and piece of snippet text into a new search engine query, submitted the queries to HotBot and compared the position of the target page, p , in the two result-lists produced. We compared a number of variations of this approach by constructing our search queries in different ways.

4.1 Query Generation

Probably the most important part of our evaluation was the generation of the test queries from both the related queries and snippet text. In all, six query generation strategies were tested: two that produced queries from the terms contained in the related queries for a page and four that used terms from the page's snippet text. Strategy *RQ1* produces a test query by concatenating the related query terms into a single query. Strategy *RQ2* uses a similar approach but duplicate terms are removed.

The snippet text conversion was slightly more complicated. To ensure a fair evaluation, we generated a test query from the snippet text using the same number of terms as the test queries produced from the related queries. To generate the test queries we parsed the snippet text to remove stop-words and special characters and then selected terms from the remaining snippet text using four different strategies. In strategy *S1* we select a random set of k terms, where k is the number of terms in the test query produced by *RQ1*. Strategy *S2* selects the top k most common terms in the snippet text, where k is the number of terms in the test query produced by *RQ2*. For strategy *S3* we again select a random set of k terms but this time k is the number of terms produced by *RQ2*. Finally, strategy *S4* selects the top k most common terms in the snippet text, where k is the number of terms produced by *RQ1*.

4.2 Relevance Assessment

After submitting each test query to the HotBot search engine, we examined the top 500 HotBot results only and compared the position of p , the target result-page, in the result-lists produced for each test query. The higher p is in the result-list the more representative the test query must be as an indicator of p 's content and hence the more representative the related queries or snippet text. Along with this positional information we also examined the percentage of results matched by each test query strategy as well as the average length in number of terms of the related queries vs. snippet-based strategies.

4.3 Results

First we looked at the average position of each p in the result-lists produced by HotBot, see Figure 2. When calculating the average position, a penalty is applied when the target result cannot be found for the query in question. If the target result cannot be found, p is given a default position value of 501 (because we are only examining the first 500 HotBot results). Both related query strategies, *RQ1* and *RQ2*, perform very well. *RQ2* locates p at an average position of 39 in the result-

list compared to the best performing snippet-based strategy which locates p at an average position of 103.

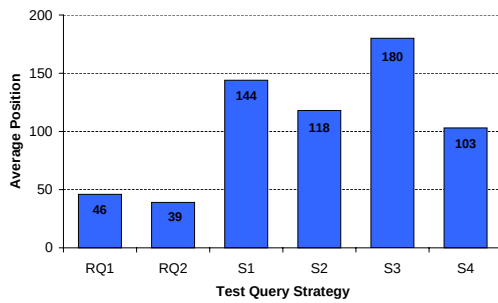


Figure 2: Average position of p in result-lists

The poor showing for the snippet text strategies could have been due to frequent penalties being incurred when p was not present in the top 500 results. To understand this we examined the percentage of results found for each test query strategy in the top 500 HotBot results, see Figure 3. The related query strategies succeed in producing result-lists that contain p for between 92% and 94% of queries, a significant improvement when compared to the snippet text strategies which return p for between 66% and 80% of the test queries.

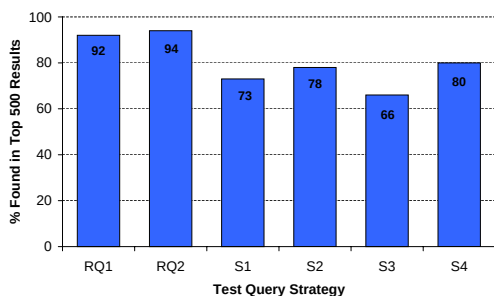


Figure 3: Percentage found in the top 500 HotBot results

These results suggest that the terms contained in the related queries are more representative of the pages they refer to than an equivalent number of terms taken from the snippet text associated with these pages. And by using related queries instead of snippet text we can achieve a significant saving in display-space. For example, as shown in Figure 4, our related queries contained only 4 unique terms on average, compared to snippet texts with an average of 35 terms (words), or 21 terms with stop-words removed.

5 Conclusion

One of the main issues with mobile search engines concerns the manner in which search results are displayed. In this paper we have suggested using related queries as an economical alternative to snippet text for result gisting. Our evaluation suggests that the terms in related queries have the potential to better capture the essence of their associated pages than the terms in the snippet texts, and so may serve to be a more

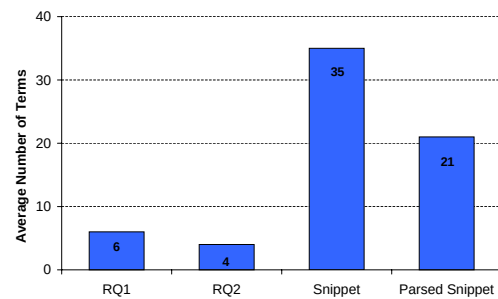


Figure 4: Average number of terms in test queries

informative gisting approach. Moreover, related query terms take up a small fraction of the screen space associated with the display of snippet text, which is a major advantage for the provision of mobile search.

6 Acknowledgements

This material is based on works supported by the Science Foundation Ireland under Grant No. 03/IN.3/I361.

References

- [Balfe and Smyth,] E. Balfe and B. Smyth. Collaborative Query Recommendation for Web Search. In *Proceedings of 16th European Conference on Artificial Intelligence*, pages 268–272, Valencia, Spain.
- [CellularOnline, 2004] CellularOnline. See, <http://www.cellular.co.za/>, 2004.
- [Church *et al.*, 2005] K. Church, M. T. Keane, and B. Smyth. An Evaluation of Gisting in Mobile Search. In *Proceedings of the 27th European Conference on Information Retrieval*, Santiago de Compostela, Spain, 2005.
- [Jones *et al.*, 2002] M. Jones, G. Buchanan, and H. Thimbleby. Sorting out Searching on Small Screen Device. In *Proceedings of the 4th International Symposium on Human Computer Interaction with Mobile Devices*, Pisa, Italy, 2002.
- [Jones *et al.*, 2004] S. Jones, M. Jones, and S. Deo. Using Keyphrases as Search Result Surrogates on Small Screen Devices. *International Journal of Personal and Ubiquitous Computing*, 8(1):55–68, 2004.
- [Ramsay and Nielsen, 2000] M. Ramsay and J. Nielsen. The WAP Usability Report, 2000.
- [Smyth *et al.*, 2003] B. Smyth, E. Balfe, P. Briggs, M. Coyle, and J. Freyne. Collaborative Web Search. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1417–1419, Acapulco, Mexico, 2003.
- [Smyth *et al.*, Forthcoming] B. Smyth, E. Balfe, J. Freyne, P. Briggs, M. Coyle, and O. Boydell. Exploiting Query Repetition & Regularity in an Adaptive Community-Based Web Search Engine. *User Modeling and User Adapted Interaction*, Forthcoming.

An Inference Model for Semantic Entailment in Natural Language

Rodrigo de Salvo Braz Roxana Girju Vasin Punyakanok

Dan Roth Mark Sammons

Department of Computer Science

University of Illinois at Urbana-Champaign

Urbana, IL, 61801, USA

{braz, girju, punyakan, danr, mssammon}@cs.uiuc.edu

Semantic entailment is the problem of determining if the meaning of a given sentence entails that of another. This is a fundamental problem in natural language understanding that provides a broad framework for studying language variability and has a large number of applications. We present a principled approach to this problem that builds on inducing re-representations of text snippets into a hierarchical knowledge representation along with a sound inferential mechanism that makes use of it to prove semantic entailment.

1 Introduction

Semantic entailment is the task of determining, for example, that the sentence: “WalMart defended itself in court today against claims that its female employees were kept out of jobs in management because they are women” entails that “Wal-Mart was sued for sexual discrimination”.

Determining whether the meaning of a given text snippet entails that of another or whether they have the same meaning is a fundamental problem in natural language understanding that requires the ability to abstract over the inherent syntactic and semantic variability in natural language [Dagan and Glickman, 2004]. This challenge is at the heart of many high level natural language processing tasks including Question Answering, Information Retrieval and Extraction, Machine Translation and others that attempt to reason about and capture the meaning of linguistic expressions.

Research in natural language processing in the last few years has concentrated on developing resources that provide multiple levels of text analysis (both syntactic and semantic), resolve various context sensitive ambiguities, and identify abstractions (from syntactic categories like POS tags to semantic ones like named entities), and the text relational structures.

Indeed, several decades of research in natural language processing and related fields have made clear that the use of deep structural, relational and semantic properties of text is a necessary step towards supporting higher level tasks. However, beyond these resources, in order to support fundamental tasks such as inferring semantic entailment between two texts snippets, there needs to be a unified knowledge representation of the text that (1) provides a hierarchical encoding of the structural, relational and semantic properties of the given text, (2) is integrated with learning mechanisms that can be used to induce such information from raw text, and (3) is equipped with an inferential mechanism that can be used to support inferences with respect to such representations.

Just resorting to general purpose knowledge representations – FOL based representations, probabilistic representations or hybrids – along with their corresponding general purpose inference algorithms is not sufficient.

We have developed an integrated approach that provides solutions to all challenges mentioned above. We formally define the problem of *semantic entailment* for Natural Language and present a computational approach to solving it, that consists of a hierarchical knowledge representation language into which we induce appropriate representations of the given text and required background knowledge, along with a sound inferential mechanism that makes use of the induced representation to determine entailment. Our inference approach is formalized as an optimization algorithm that we model as an integer linear programming problem. The preliminary evaluation of our approach is very encouraging and illustrates the significance of some of the key contributions of this approach.

1.1 General Description of Our Approach

Given two text snippets S (source) and T (target) (typically, but not necessarily, S consists of a short paragraph and T , a sentence) we want to determine if $S \models T$, which we read as “ S entails T ” and, informally, understand to mean that *most people would agree that the meaning of S implies that of T* . Somewhat more formally, we say that S entails T when some representation of T can be “matched” (modulo some meaning-preserving transformations to be defined below) with some (or part of a) representation of S , at some level of granularity and abstraction.

The approach consists of the following components:

KR: A Description Logic based hierarchical knowledge representation, EFDL, into which we re-represent the surface level text representations, augmented with induced syntactic and semantic parses and word and phrase level abstractions.

KB: A knowledge base consisting of syntactic and semantic rewrite rules, written in EFDL.

Subsumption: An extended subsumption algorithm which determines subsumption between EFDL expressions (representing text snippets or rewrite rules). “Extended” here means that the basic unification operator is extended to support several word level and phrase level abstractions.

First a set of machine learning based resources are used to induce the representation for S and T . The entailment algorithm then proceeds in two phases: (1) it incrementally gen-

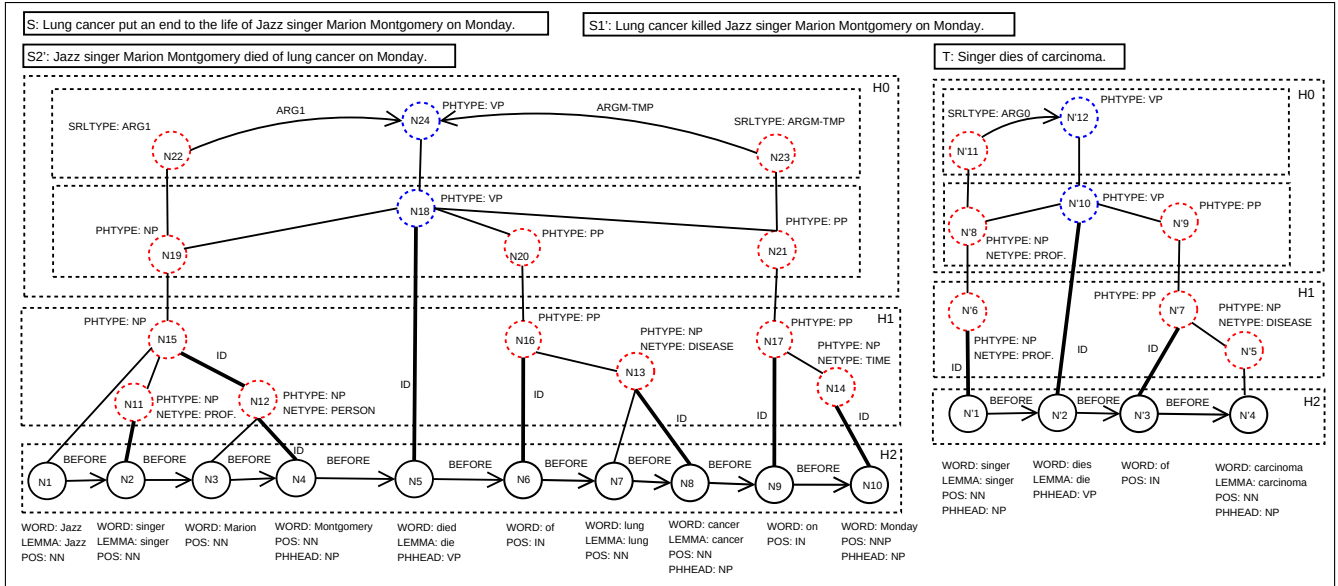


Figure 1: Example of *Re-represented Source & Target* pairs as concept graphs. The original source sentence S generated several alternatives including S'_1 and the sentence in the figure (S'_2). Our algorithm was not able to determine entailment of the first alternative (as it fails to match in the extended subsumption phase), but it succeeded for S'_2 . The dotted nodes represent phrase level abstractions. S'_2 is generated in the first phase by applying the following chain of inference rules: #1 (genitives): “Z’s $W \rightarrow W$ of Z”; #2: “X put end to Y’s life $\rightarrow$ Y die of X”. In the extended subsumption, the system makes use of WordNet hypernymy relation (“lung cancer” IS-A “carcinoma”) and NP-subsumption rule (“Jazz singer Marion Montgomery” IS-A “singer”). The rectangles encode the hierarchical levels (H_0 , H_1 , H_2) at which we applied the extended subsumption. Also note that in the current experiments we don’t consider noun plurals and verb tenses.

erates re-representations of the original representation of the source text S and (2) it makes use of an (extended) subsumption algorithm to check whether any of the alternative representations of the source entails the representation of the target T . The subsumption algorithm mentioned above is used in both phases in slightly different ways.

Figure 1 provides a graphical example of the representation of two text snippets, along with a sketch of the extended subsumption approach to decide the entailment.

Along with the formal definition developed here of semantic entailment, our knowledge representation and algorithmic approach provide a novel solution that addresses some of the key issues the natural language research community needs to address in order to move forward towards higher level tasks of this sort. Namely, we provide ways to represent knowledge, either external or induced, at multiple levels of abstractions and granularity, and reason with it at the appropriate level.

2 Experimental Evaluation

Data. We tested our approach on the PASCAL challenge data set (<http://www.pascal-network.org/Challenges/RTE/>). As the system was designed to test for semantic entailment, the PASCAL data set is ideally suited, being composed of 276 source - target sentence pairs, indicating whether the source logically entails the target. The set is split into various tasks: CD (Comparable Documents), IE (Information Extraction), MT (Machine Translation), PP (Prepositional Paraphrases), QA (Question Answering), and RC (Reading Comprehension).

In Table 1 we show the system’s performance. The base-

line is a lexical-level matching based on bag-of-words representation with lemmatization and normalization (LLM).

Perform.	Overall [%]	Task [%]						
		CD	IE	IR	MT	PP	QA	RC
System	64.8	74.0	35.0	62.0	87.5	63.8	84.0	49.0
LLM	54.7	64.0	50.0	50.0	75.0	55.2	50.0	52.9

Table 1: System’s performance obtained for each experiment on the Pascal corpora and its subtasks.

Acknowledgement

We thank Dash Optimization for the free academic use of their Xpress-MP software. This work was supported by the Advanced Research and Development Activity (ARDA)’s Advanced Question Answering for Intelligence (AQUAINT) Program, NSF grant ITR-IIS-0085980, and ONR’s TRECC and NCASSR programs.

References

[Dagan and Glickman, 2004] I. Dagan and O. Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. In *Learning Methods for Text Understanding and Mining*, Grenoble, France, 2004.

Discovering Inconsistency through Examination Dialogues

Paul E. Dunne Sylvie Doutre Trevor Bench-Capon

Department of Computer Science

University of Liverpool

Liverpool L69 7ZF

United Kingdom

{ped,sd,tbc}@csc.liv.ac.uk

Abstract

In this paper we introduce *examination* dialogues, an addition to the dialogue typology of Walton and Krabbe. In educational settings the purpose of dialogue is often to elicit the position of a student, e.g. to test understanding. In other settings, a frequently adopted tactic is to attack an opponent's stance by exposing internal inconsistencies in their argument. In real debate such inconsistencies will often be rather more subtle than elementary logical fallacies since they arise from contradictions apparent in the opponent's value system. Protagonists will be better positioned to judge the applicability of this tactic as more information is determined concerning the exact nature of their opponent's case, e.g. the arguments favoured and values endorsed. One obstacle, however, is that following a request to state a view, the challenged party may refuse to comment. In this paper we present an approach to modelling the evolution of examination dialogues based on the concept of value-based argument frameworks and outline some algorithmic issues regarding argument selection.

1 Introduction

One important class of dialogue not found in the influential typology of [Walton and Krabbe, 1995] is the *examination* dialogue. In such dialogues one party – the *Questioner Q* – elicits statements and opinions from another – the *Responder R* – with the aim of discovering *R*'s position on some topic, either to gain insight into *R*'s understanding and knowledge of the topic, or to expose an inconsistency in *R*'s position. Examples include education by the Socratic method, *viva voce* examination, cross examination of witnesses, and political interviews. In contrast to information seeking or inquiry dialogues, *Q* may already have beliefs on the topic: unlike persuasion dialogues, *Q* may have no intention of converting *R* to his position. Examination dialogues may, however, be nested within information seeking dialogues: probing for inconsistency increases confidence in the veracity of *R*'s beliefs; similarly, in persuasion dialogues, exposing inconsistency is a useful prelude to persuasion.

In examination dialogues the dialogue process occurs within an environment of relevant facts, arguments, and related issues that is (at least implicitly) recognised and understood by both parties, e.g. common knowledge of the subject in educational settings, matters of specific policies and legislation in political interviews; issues of evidence and witness testimony in trials; details of incidents under investigation in questioning suspects. In this paper our aim is to describe a generic abstract model within which the evolution of such examination dialogues can be analysed. In addition we outline an approach intended to aid *Q* in selecting questions.

The foundational element for our model is provided by the Value-based Argument Frameworks (VAFs) of [Bench-Capon, 2003], a development of the classical argument systems of [Dung, 1995], which offer a richer interpretation of the concept of an argument x “attacking” an argument y . In addition to the computational advantages outlined in [Bench-Capon, 2003], further indications that VAFs provide a suitable mechanism for our analysis is presented by the methods discussed in [Dunne and Bench-Capon, 2004b]. A brief overview of VAFs is given in Section 2 and our model of examination dialogue presented in Section 3. In Section 4 we outline some algorithmic aspects involving argument selection in this scheme. Conclusions are presented in Section 5.

2 Value-Based Argument Frameworks

Value-based argumentation frameworks are introduced in [Bench-Capon, 2003] as a development of the model of argument systems via directed graphs $\mathcal{H} = \langle \mathcal{X}, \mathcal{A} \rangle$ proposed in [Dung, 1995]. A *value-based argumentation framework* (VAF) is defined as $\langle \mathcal{X}, \mathcal{A}, \mathcal{V}, \eta \rangle$, where $\langle \mathcal{X}, \mathcal{A} \rangle$ is an argument system, $\mathcal{V} = \{v_1, v_2, \dots, v_k\}$ is a set of k values, and $\eta : \mathcal{X} \rightarrow \mathcal{V}$ associates a value $\eta(x) \in \mathcal{V}$ with each argument $x \in \mathcal{X}$. An *audience* ϑ for a VAF $\langle \mathcal{X}, \mathcal{A}, \mathcal{V}, \eta \rangle$ is a transitive and irreflexive binary relation on $\mathcal{V}$. A pair $\langle v_i, v_j \rangle$ in ϑ is referred to as ‘ v_i is preferred to v_j ’ with respect to ϑ .

Ideas analogous to those in Dung's argument system are now defined relative to some audience: $S \subseteq \mathcal{X}$ is *conflict-free* w.r.t. ϑ if $\forall x, y \in S: \langle x, y \rangle \in \mathcal{A} \Rightarrow \langle \eta(y), \eta(x) \rangle \in \vartheta$; x is acceptable to S w.r.t. ϑ if $\forall y \in \mathcal{X}: \langle y, x \rangle \in \mathcal{A} \Rightarrow (\langle \eta(x), \eta(y) \rangle \in \vartheta \text{ or } \exists z \in S \text{ s.t. } \langle z, y \rangle \in \mathcal{A} \text{ and } \langle \eta(y), \eta(z) \rangle \notin \vartheta)$; S is admissible w.r.t. ϑ if each $x \in S$ is acceptable to S w.r.t. ϑ .

3 Examination Dialogues in VAFs

In keeping with the observation that debates of interest take place within some recognised context, we view this context as a VAF $\mathcal{G} = \langle \mathcal{X}, \mathcal{A}, \mathcal{V}, \eta \rangle$. A *debate in the context of $\mathcal{G}$* is a sequence $\mu_1 \mu_2 \dots \mu_j$ of at most $2|\mathcal{X}| + 1$ moves: odd indexed moves being played by **Q** and even indexed moves by **R**.

Suppose the next move to be made is μ_{2i-1} ($i \geq 1$). The options for the next move(s) – $\mu_{2i-1} \cdot \mu_{2i}$ – are,

(a) $\mu_{2i-1} = \text{ASK}(p)$: **Q** asks **R** their view on $p \in \mathcal{X}$; $\text{ASK}(p)$ is legal provided that it has not occurred at an earlier point in the debate, i.e. $\forall j < i \mu_{2j-1} \neq \text{ASK}(p)$. The response μ_{2i} is one of $\text{ACCEPT}(p)$ (**R** agrees with p); $\text{DENY}(p)$ (**R** does not agree with p); or $\text{NO-COMMENT}(p)$ (**R** refuses to say anything about p).

(b) $\mu_{2i-1} = \text{FOUND}(p, \text{STAT}(p))$: here p is an argument to which **R** has responded $\text{NO-COMMENT}(p)$ after $\text{ASK}(p)$ earlier. **Q** expresses a claim to know **R**'s view of p as accepted ($\text{STAT} = \text{ACCEPT}$) or otherwise ($\text{STAT} = \text{DENY}$).

(c) $\mu_{2i-1} = \text{RESOLVE}(S)$: S is a set of arguments all of which have been asked about with none receiving a “NO-COMMENT” reply. **Q** asserts that *at least* one of these replies cannot be correct with respect to the context $\mathcal{G}$.

(d) $\mu_{2i-1} = \text{CONCEDE}$: **Q** can neither determine **R**'s view in respect of NO-COMMENT arguments nor expose inconsistencies in the collection of “definite” responses.

The debate ends once one of the moves (b)–(d) is played. Noting that we assume (b) or (c) can be played *only if* **Q** may justify the claim made, these result in a “win” for **Q**, whereas (d) results in a “win” for **R**.

As a very simple example consider: $\mathcal{X} = \{x, y, z\}$; $\mathcal{A} = \{\langle x, y \rangle, \langle y, z \rangle, \langle z, x \rangle\}$; $\mathcal{V} = \{A, B\}$; and $\eta(x) = \eta(z) = A$; $\eta(y) = B$. If **R** replies $\text{NO-COMMENT}(y)$ or $\text{DENY}(y)$ to $\text{ASK}(y)$ then **Q** wins (by playing $\text{FOUND}(y, \text{ACCEPT})$ or $\text{RESOLVE}(y)$) since, regardless of whether **R**'s audience includes $\langle A, B \rangle$ or $\langle B, A \rangle$, **R** must accept y . On the other hand – assuming **R** replies $\text{ACCEPT}(y)$ – the only responses regarding **R**'s view of $\{x, z\}$ that could lead to a win for **Q** are when *exactly one* NO-COMMENT occurs; or it is claimed both are accepted; or both denied. Notice that may **R** report $\langle \text{ACCEPT}(x), \text{DENY}(z) \rangle$ (or *vice versa*) despite believing otherwise and **Q** will be unable to detect this.

In general, within such debate contexts, both players have a number of non-trivial strategic issues to resolve. In particular, assuming that **R** wishes to hide information (and therefore will either respond “NO-COMMENT” and/or misreport some views), **R** may have to form a partition of $\mathcal{X}$ determining those queries to which false and true responses will be given. We note that, as indicated in the small example above, refusing to comment on any argument or misreporting the status of every argument may fail to be a successful strategy; moreover it is an easy consequence of [Dunne and Bench-Capon, 2004a] that the decision problem faced by **R** determining if classifying p as NO-COMMENT loses, is coNP-complete.

4 Choosing Arguments

We conclude the brief technical discussion, by outlining an approach to query selection that is based on a translation to propositional formulae. Given $\langle \mathcal{X}, \mathcal{A}, \mathcal{V}, \eta \rangle$, in order for $p \in$

$\mathcal{X}$ to be accepted w.r.t. at least one audience ϑ , it must belong to an admissible w.r.t. ϑ set $S \subseteq \mathcal{X}$. Consider disjoint sets of propositional variables Z and $W = \{w_{i,j} : 1 \leq i \neq j \leq k\}$. Define the function, $f(Z, W)$ via: $f(\alpha, \beta) = \top$ if and only if the subset S of $\mathcal{X}$ indicated by α , i.e. $x_i \in S \Leftrightarrow \alpha_i = \top$, is admissible with the binary relation ϑ encoded in β , i.e. $\langle v_i, v_j \rangle \in \vartheta \Leftrightarrow \beta_{i,j} = \top$ and this relationship is an audience, i.e. irreflexive and transitive. This function can be described by a “short” propositional formula $\Phi_f(Z, W)$ of length $O(|\mathcal{A}| + k^3)$ that with the addition of at most $O(|\mathcal{A}|)$ auxiliary variables (D) translates to an equivalent CNF formula, $\Psi_f(Z, W, D)$ with $O(|\mathcal{A}| + k^3)$ clauses, the longest of which contains at most $2 + \max_{x \in \mathcal{X}} |\{y : \langle y, x \rangle \in \mathcal{A}\}|$ literals. Using Ψ_f the strategy pursued by **Q** will be to obtain responses from **R** that result in Ψ_f evaluating to $\perp$ (i.e. **R** is inconsistent) or to infer the unique instantiation of z_i that satisfies Ψ_f with the replies already given.

5 Conclusions and Development

We have introduced a class of question-response dialogues – *examination dialogues* – and modelled them using VAF schemata as the basis for representing related knowledge. A number of issues – both theoretical and applicative – form the subject of work in progress. In addition to on-going empirical investigations of argument selection strategies employing the propositional translation briefly outlined in Section 4 there are several questions of interest regarding properties of the model when the protagonists are “limited” in different ways. Thus, to what extent is **Q** at an advantage if an upper bound is placed on the number of “NO-COMMENT” and/or false responses **R** can make? Similarly, if **Q** is restricted to some maximum number of questions (rather than being allowed to ask about the status of every argument in $\mathcal{X}$) it is likely that **Q** must adopt a dynamic selection mechanism (rather than choosing a set to query in advance) so as to adapt effectively to **R**'s responses.

References

- [Bench-Capon, 2003] T. J. M. Bench-Capon. Persuasion in Practical Argument Using Value-based Argumentation Frameworks. *Journal of Logic and Computation*, 13(3):429–448, 2003.
- [Dung, 1995] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games. *Artificial Intelligence*, 77:321–357, 1995.
- [Dunne and Bench-Capon, 2004a] P. E. Dunne and T. J. M. Bench-Capon. Complexity in value-based argument systems. In *Proc. 9th JELIA*, volume 3229 of *LNAI*, pages 360–371. Springer-Verlag, 2004.
- [Dunne and Bench-Capon, 2004b] P. E. Dunne and T. J. M. Bench-Capon. Identifying audience preferences in legal and social domains. In *Proc. DEXA'04*, volume 3180 of *LNCS*, pages 518–527. Springer-Verlag, 2004.
- [Walton and Krabbe, 1995] D. N. Walton and E. C. W. Krabbe. *Basic Concepts of Interpersonal Reasoning*. State Univ. of New York Press, 1995.

A Probabilistic Lexical Approach to Textual Entailment*

Oren Glickman, Ido Dagan, Moshe Koppel

Bar Ilan University

Computer Science Department

Ramat-Gan, Israel

{glikmao, dagan, koppel}@cs.biu.ac.il

Abstract

The textual entailment problem is to determine if a given text entails a given hypothesis. This paper describes first a general generative probabilistic setting for textual entailment. We then focus on the sub-task of recognizing whether the lexical concepts present in the hypothesis are entailed from the text. This problem is recast as one of text categorization in which the classes are the vocabulary words. We make novel use of Naïve Bayes to model the problem in an entirely unsupervised fashion. Empirical tests suggest that the method is effective and compares favorably with state-of-the-art heuristic scoring approaches.

Textual Entailment

Many Natural Language Processing (NLP) applications need to recognize when the meaning of one text can be expressed by, or inferred from, another text. Information Retrieval (IR), Question Answering (QA), Information Extraction (IE) and text summarization are examples of applications that need to assess such semantic overlap between text segments. Textual Entailment Recognition has recently been proposed as an application independent task to capture such semantic inferences and variability [Dagan et al., 2005]. A text t textually entails a hypothesis h if t implies the truth of h . Textual entailment captures generically a broad range of inferences that are relevant for multiple applications. For example, a QA system has to identify texts that entail the expected answer. Given the question "Where was Harry Reasoner born?", a text that includes the sentence "*Harry Reasoner's birthplace is Iowa*" entails the expected answer form "Harry Reasoner was born in Iowa." In many cases, though, entailment inference is uncertain and has a probabilistic nature. For example, a text that includes the sentence "*Harry Reasoner is returning to his Iowa hometown to get married.*" does not deterministically entail the above answer form. Yet, it is clear that it does add substantial information about the correctness of the hypothesized assertion.

A Probabilistic Setting

We propose a general generative probabilistic setting for textual entailment. We assume that a language source generates texts within the context of some state of affairs. Thus, texts are generated along with hidden truth assignments to hypotheses. We define two types of events over the corresponding probability space:

I) For a hypothesis h , we denote as Tr_h the random variable whose value is the truth value assigned to h in the world of the generated text. Correspondingly, $Tr_h=1$ is the event of h being assigned a truth value of 1 (True).

II) For a text t , we use t to denote also the event that the generated text is t .

Textual entailment relationship: We say that t probabilistically entails h (denoted as $t \Rightarrow h$) if t increases the likelihood of h being true, that is if $P(Tr_h = 1 | t) > P(Tr_h = 1)$.

Entailment confidence: We quantify the marginal amount of information contributed by the text to assessing the truth of the hypothesis relative to its prior with the pointwise mutual information: $I(Tr_h=1, t) = \log(P(Tr_h = 1 | t) / P(Tr_h = 1))$.

An Unsupervised Lexical Model

The proposed setting above provides the necessary grounding for probabilistic modeling of textual entailment. However, it is important to bear in mind that it is not trivial to estimate the constituent probabilities in the definition of textual entailment since the truth assignments of hypotheses for the corpus' texts are not observed.

Lexical Entailment as Text classification

As modeling the full extent of the textual entailment problem is a long term research goal, we focus on a sub task we term lexical entailment - recognizing if the individual lexical concepts in a hypothesis are entailed from a given text.

When estimating the entailment probability we assume that the truth probability of a term in a hypothesis h is independent of the truth of the other terms in h , obtaining:

$$\begin{aligned} P(Tr_h = 1 | t) &= \prod_{u \in h} P(Tr_u = 1 | t) \\ P(Tr_h = 1) &= \prod_{u \in h} P(Tr_u = 1) \end{aligned} \quad (1)$$

At this point, it is perhaps best to think of the entailment problem as a text classification task in which the classes are an abstract binary notion of lexical truth (for the different words in the vocabulary). First, we construct the initial la-

* This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

being based solely on the explicit presence or absence of each u in t . Then we apply Naïve Bayes in an unsupervised fashion that derives analytically from the defined probabilistic setting.

Initial Labeling

As an initial approximation, we assume that for any document in the corpus the truth value corresponding to a term u is determined by the explicit presence or absence of u in that document.

In some respects the initial labeling is similar to systems that perform a Boolean search (with no expansion) on the keywords of a textual hypothesis in order to find candidate (entailing) texts. Of course, due to the semantic variability of language, similar meanings could be expressed in different wordings, which is addressed in the subsequent model. The initial labeling, however, may provide useful estimates for this model.

Naïve Bayes Refinement

Following the standard naïve Bayes assumption, we can rewrite the probability $P(\text{Tr}_u=1|t)$ as in (2). In this way we are able to estimate $P(\text{Tr}_u=1|t)$ based on the prior $P(v|\text{Tr}_u=1)$ and the lexical probabilities $P(v|\text{Tr}_u=1)$ and $P(v|\text{Tr}_u=0)$ for every u, v in the vocabulary V . These probabilities are easily estimated from the corpus given the initial model's estimate of truth assignments, assuming a multinomial event model for documents and Laplace smoothing ([McCallum and Nigam, 1998]).

$$P(\text{Tr}_u=1|t) = \frac{P(\text{Tr}_u=1) \prod_{v \in t} P(v|\text{Tr}_u=1)}{\sum_{c \in \{0,1\}} P(\text{Tr}_u=c) \prod_{v \in t} P(v|\text{Tr}_u=c)} \quad (2)$$

From above equations we have a refined probability estimate for $P(\text{Tr}_h=1|t)$ and $P(\text{Tr}_h=1)$ for any arbitrary text t and hypothesis h . The criterion for turning probability estimates into classification decisions is derived analytically from our proposed probabilistic setting of textual entailment. We classify positively for entailment if $P(\text{Tr}_h=1|t) > P(\text{Tr}_h=1)$ and assign a confidence score of $\log(P(\text{Tr}_h=1|t) / P(\text{Tr}_h=1))$ for ranking purposes. In fact, the empirical evaluation showed this analytic threshold to be almost optimal.

1 Empirical Evaluation

Experimental Setup

Though empirical modeling of semantic inferences between texts is commonly done within application settings, there is no common dataset available to specifically evaluate a textual entailment system. In order to test our model we therefore needed an appropriate set of text-hypothesis pairs. We chose the information seeking setting, common in applications such as QA and IR, in which a hypothesis is given and it is necessary to identify texts that entail it. The evaluation criterion is application-independent based on human judgment of textual entailment.

Experiments were done on the *Reuters Corpus Volume 1*. An annotator chose 50 sentential hypotheses from the corpus sentences. We required that the hypotheses convey a reasonable information need in such a way that they might

correspond to potential questions, semantic queries or IE relations. We created a set of candidate entailing texts for the given set of test hypotheses, by following common practice of morphological and WordNet-based expansion. Boolean search (with expanded words) was performed at the paragraph level over the full Reuters corpus. Paragraphs containing all original words of the hypothesis or their morphological derivations were excluded from the result set and selected a random set of 20 texts for each of the hypotheses.

The resulting dataset was given to two judges to be annotated for entailment. Corresponding to the notion of textual entailment, judges were asked to annotate a text-hypothesis pair as true if, given the text, they could infer with some confidence that the hypothesis is true. They were instructed to annotate the example as false if either they believed the hypothesis to be false given the text or if the text is unrelated to the hypothesis. A subset of 200 pairs was cross-annotated for agreement, resulting in a moderate Kappa statistic of 0.6. Overall, the annotators deemed 48% of the text-hypothesis pairs as positive examples of entailment.

Empirical Results

We trained our model on the Reuters corpus, classified the text-hypothesis pairs of the dataset and compared the model's prediction with the human judgments. The resulting (macro) average accuracy was 70%. Since our dataset did not include texts containing all content words of the hypotheses, the baseline model would have predicted none of the pairs to be correct (i.e. the text entails the hypothesis) yielding an average accuracy of only 52%.

We also compared our system's ranking ability. The entailment confidence score was used to rank the various texts of each hypothesis. The average *confidence weighted score* (cws) was measured for each hypothesis. The resulting cws macro average was 0.54 compared to average cws of 0.49 for random ordering. For further comparison, a state of the art *idf* semantic overlap measure [Saggion et al., 2004; Monz and de Rijke, 2001] achieved a score of 0.51. Though our model performs just slightly better, the results are statistically significant at the 0.02 level.

References

- [Dagan et al., 2005] Ido Dagan, Oren Glickman and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. *PASCAL Challenges Workshop*.
- [McCallum and Nigam, 1998] Andrew McCallum and Kamal Nigam. A Comparison of Event Models for Naive Bayes Text Classification. *AAAI-98 Workshop on "Learning for Text Categorization"*.
- [Monz and de Rijke, 2001] Christof Monz, Maarten de Rijke. Light-Weight Entailment Checking for Computational Semantics. *The third workshop on inference in computational semantics (ICoS-3)*.
- [Saggion et al., 2004] Horacio Saggion, Rob Gaizauskas, Mark Hepple, Ian Roberts and Mark A. Greenwood. Exploring the Performance of Boolean Retrieval Strategies For Open Domain Question Answering. *SIGIR04 Workshop on Information Retrieval for Question Answering*.

Structural Representation and Matching of Articulatory Speech Structures based on the Evolving Transformation System (ETS) Formalism

Alexander Gutkin

Centre for Speech Technology Research
University of Edinburgh
Edinburgh, EH8 9LW, UK
alexander.gutkin@ed.ac.uk

David Gay

Faculty of Computer Science
University of New Brunswick
Fredericton, NB, E3B 5A3, Canada
dave.gay@unb.ca

Abstract

A formal structural representation of speech is presented in this paper. The representation is developed within the Evolving Transformation System (ETS) formalism and encapsulates speech processes at the articulatory level. We show how the *class structure* of several consonantal phonemes of English can be expressed via articulatory gestures. Elements of these classes can be detected in a corresponding *structural representation* of continuous speech. Our experimental results on the MOCHA articulatory corpus support the hypothesis that the proposed articulatory representation captures sufficient information for the accurate structural identification of phonemic classes.

1 Introduction

Despite the evident success of numerical approaches to modelling of speech, they are often criticised for having little relation to actual human speech production/recognition [Jelinek, 1997, p. 10]. The alternative, structural, means of pattern representation have, however, received little attention. In our view, one of the main reasons for this situation is the apparent lack of suitable structural frameworks possessing the necessary formal power to accommodate the representation of classes of complex linguistic phenomena (e.g. phonemes and syllables). It is hypothesised that the appearance of such a systematic analytical framework and the development of appropriate representations within it may help to bridge the gap between speech recognition and linguistic research.

On the linguistic side, the motivation for this work comes from the theory of articulatory phonology [Browman and Goldstein, 1992], in which vocal tract action during speech production is decomposed into discrete, re-combinable atomic units known as *gestures*. Compared to traditional approaches—such as distinctive phonological features [Jakobson and Halle, 1971]—the gestural approach is more physiologically concrete and offers a compact means of representing the truly asynchronous nature of speech, allowing for better interpretations of complex phonological phenomena (such as co-articulation).

The Evolving Transformation System (ETS), outlined in [Goldfarb *et al.*, 2004], is a radically new formal frame-

Organ	Semantics	Group	Organs	<i>G</i>
UL	upper lip	bilabial closure	UL, LL	6
LL	lower lip	tongue dorsum height	TD	4
UI	upper incisor	tongue tip height	TT	4
TD	tongue dorsum	labiodental contact	UI, LL	4
TT	tongue tip	velic aperture	VL	4
VL	velum	velar contact	TD, VL	2
HP	hard palate	alveolar contact	TT, AR	2
AR	alveolar ridge	palatal contact	TT, HP	2
VF	vocal folds	voicing	VF	2

Table 1: Articulatory organs (left) involved in the production of various groups of primitive gestures (right).

work for the structural representation of “natural” processes. ETS suggests that the representation of each such process should include its “formative history”, which is a series of “operations” (*primitives*) acting on the constituent elements (*sites*) of the process. Such formative histories (*structs*) should contain some regular “chunks” (*transforms*). A class representation in ETS is a finite set of closely related transforms, out of which the corresponding class elements—processes—can be constructed. In our ETS representation of gestural speech structure, the natural process we model is the physiological process of articulation.

2 Articulatory Structure according to ETS

The units of representation corresponding to articulatory organs and primitive articulatory gestures are ETS *sites* and *primitives*, respectively. Informally, an ETS primitive is a unit of temporal structure of a process, describing the structural transformation of its set of “initial” sites into its set of “terminal” sites, where an ETS site is the smallest/unstructured representational unit within a process. In the articulatory representation, a primitive is identified with a change in the interaction of one or more of the associated articulatory organs, which are expressed as sites.

The left-hand side of Table 1 lists all of the ETS sites used in the representation, along with the corresponding interpretation. The right-hand side of Table 1 shows the groups of primitive gestures used in this study. For each group, the relevant sites (articulators) and the number of distinct constituent gestures (*primitives*) *G* are shown. Informally, a group consists of semantically and syntactically related primitive gestures involving similar articulators. Pictorially, it is convenient to

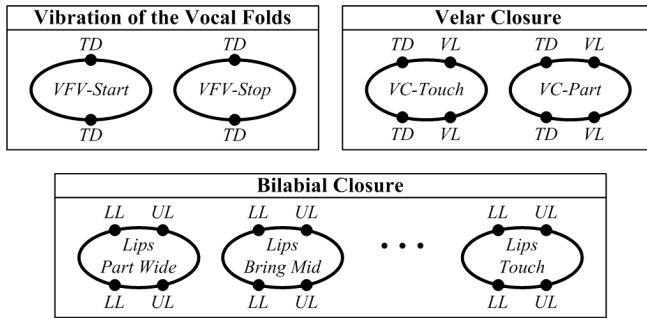


Figure 1: Several of the ETS primitives used in this representation, grouped by articulatory category.

represent primitives as convex shapes, with initial sites depicted as points on the upper half, and terminal sites depicted as points on the lower half. Some of the 30 primitives used in the articulatory representation are given in Figure 1. Three groups are shown: two articulatory gestures of the vocal folds resulting in voiced or unvoiced sounds, the two gestures participating in velar closure, and three of the six gestures modelling the aperture of the lips (bilabial closure).

An ETS *struct* is a temporally-ordered sequence of connected primitives capturing the “history” of the corresponding process. Within our articulatory representation, a struct is identified with a sequence of primitive gestures, which are hypothesised to encapsulate the gestural structure of any given utterance (note that any utterance can itself be interpreted as a highly non-trivial gesture). Figure 2 depicts the gestural structure of the word “get” in our representation. The articulation of /g/, for instance, has a simple interpretation: a velar constriction is formed and then released. The formation of the constriction is achieved when the tongue dorsum TD first rises to its maximum position (TD-RaiseMax) at 0.248 sec completing the constriction before the phoneme boundary by touching the velum VL (VC-Touch) at 0.266 sec). The constriction release is accomplished by lowering the tongue dorsum TD (TD-LowerMid) at 0.416 sec, then parting the tongue dorsum TD from the velum VL (VC-Part) at 0.460 sec). Note: vibration of the vocal folds VF (VFV-Start) occurs at the onset of /g/ at 0.380 sec.

An ETS *transform* is an encapsulation of a regular temporal pattern of primitives. An ETS *supertransform* is a set of closely-related transforms specifying the description of a class, where structural variations account for noise in the class. Note that the learning of class structure is described in [Goldfarb *et al.*, 2004] and is outside of the scope of this paper. In the articulatory representation, a supertransform is identified with the family of articulatory gestures that collectively describe the class structure of a single, general phoneme. We defined 14 consonantal classes (see, for example, the identified phoneme /g/ in Figure 2).

We developed a *structural matching algorithm* for detecting the presence of an ETS transform in a given struct. It performs a rooted depth-first search in $O(n^2 \log n)$ time, where n is the number of primitives in the transform to be matched.

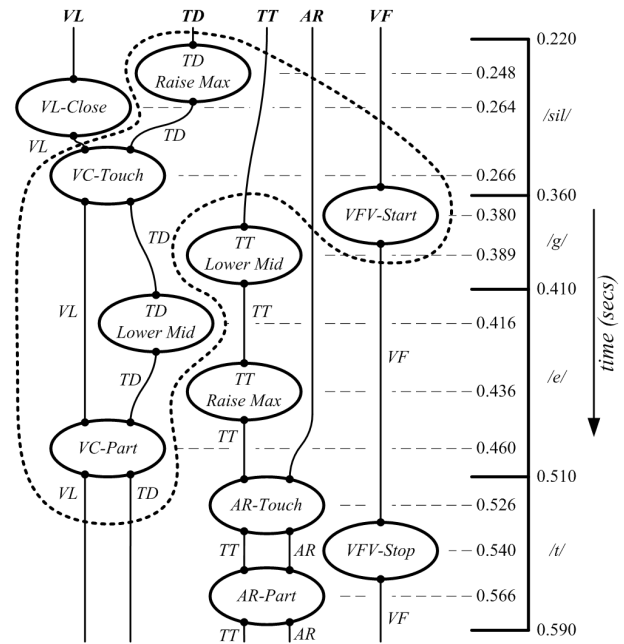


Figure 2: Partial ETS struct describing the gestural structure of the word “get”, constructed using primitive gestures detected on-the-fly in sample data. The transform corresponding to phoneme /g/ is identified with a dotted line.

3 Experiments and Discussion

Our goal was to assess the performance of the 14 ETS supertransforms via the structural matching algorithm against structures derived following [Gutkin and King, 2005] from real data (920 MOCHA utterances, 32169 ETS primitives).

Out of 9878 phonemes, 7679 were classified correctly and 278 failed to match against any of the 14 available classes. The overall accuracy is 77.74%. Analysis shows that /n/ was often misclassified as /d/ and /m/ as /b/, both due to a failure of the pre-processor to detect the corresponding physiological changes. We expect performance to improve with a more accurate pre-processor and refined class descriptions.

References

- [Browman and Goldstein, 1992] C. Browman and L. Goldstein. *Articulatory Phonology: An Overview*. *Phonetica*, 49:155–180, 1992.
- [Goldfarb *et al.*, 2004] L. Goldfarb, D. Gay, O. Golubitsky, and D. Korkin. What is a structural representation? Technical Report TR04-165, Faculty of Computer Science, University of New Brunswick, Canada, April 2004.
- [Gutkin and King, 2005] A. Gutkin and S. King. Detection of Symbolic Gestural Events in Articulatory Data for Use in Structural Representations of Continuous Speech. In *Proc. ICASSP*, Philadelphia, March 2005.
- [Jakobson and Halle, 1971] R. Jakobson and M. Halle. *Fundamentals of Language*. Mouton de Gruyter, 1971.
- [Jelinek, 1997] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, March 1997.

Sentence Extraction for Legal Text Summarisation

Ben Hachey and Claire Grover

University of Edinburgh

School of Informatics

2 Buccleuch Place

Edinburgh EH8 9LW, UK

{bhachey,grover}@inf.ed.ac.uk

Abstract

We describe a system for generating extractive summaries of texts in the legal domain, focusing on the relevance classifier, which determines which sentences are abstract-worthy. We experiment with naïve Bayes and maximum entropy estimation toolkits and explore methods for selecting abstract-worthy sentences in rank order. Evaluation using standard accuracy measures and using correlation confirm the utility of our approach, but suggest different optimal configurations.

1 Introduction

In the SUM project we are developing a system for summarising legal judgments that is generic and portable and which maintains a mechanism to account for the rhetorical structure of the argumentation of a case. Following Teufel and Moens [2002], we are developing a text extraction system that retains a flavour of the fact extraction approach. This is achieved by combining sentence selection with information about *why* a certain sentence is extracted—e.g. is it part of a judge’s argumentation, or does it contain a decision regarding the disposal of the case? In this way we are able to produce flexible summaries of varying length and for various audiences. Sentences can be reordered, since they have rhetorical roles associated with them, or they can be suppressed if a user is not interested in certain types of rhetorical roles.

We have prepared a new corpus of UK House of Lords judgments (HOLJ) for this work which contains two layers of manual annotation: rhetorical role and relevance. The rhetorical roles represent the sentence’s contribution to the overall communicative goal of the document. In the case of HOLJ texts, the communicative goal for each lord is to convince their peers of the soundness of their argument. In the current version of the corpus there are 69 judgments which have been annotated for rhetorical role. The second manual layer is annotation of sentences for ‘relevance’ as measured by whether they match sentences in hand-written summaries. In the current version of the corpus, 47 of the 69 judgments which have been annotated for rhetorical role have also been annotated for relevance. A third layer of annotation is automatic linguistic annotation, which provides the features which are used by the rhetorical role and relevance classifiers.

2 Classification and Relevance

Following from [Kupiec *et al.*, 1995], machine learning has been the standard approach to text extraction summarisation as it provides an empirical method for combining different information sources about the textual unit under consideration. For relevance prediction, we performed experiments with publicly available naïve Bayes (NB) and maximum entropy (ME) estimation toolkits. The NB implementation, found in the *Weka* toolkit, is based on John and Langley’s [1995] algorithm incorporating statistical methods for nonparametric density estimation of continuous variables. The ME estimation toolkit, written by Zhang Le, contains a C++ implementation of the LMVM [Malouf, 2002] estimation algorithm. For ME, we use the *Weka* implementation of Fayyad and Irani’s [1993] MDL algorithm to discretise numeric features.

The features that we have been experimenting with for the HOLJ corpus are broadly similar to those used by Teufel and Moens [2002]. They consist of **location** features encoding the position of the sentence in document, speech and paragraph; a **thematic words** feature encoding the average *tf*idf* weight of the sentence terms; a **sentence length** feature encoding the number of tokens in the sentence; **quotation** features encoding percentage of sentence tokens inside and in-line quote and whether or not the sentence is inside a block quote; **entity** features encoding the presence or absence of named entities in the sentence; and **cue phrase** features.

The term ‘cue phrase’ covers the kinds of stock phrases which are frequently good indicators of rhetorical status (e.g. phrases such as *The aim of this study* in the scientific article domain and *It seems to me that* in the HOLJ domain). Teufel and Moens invested a considerable amount of effort in building hand-crafted lexicons where the cue phrases are assigned to one of a number of fixed categories. A primary aim of the current research is to investigate whether this information can be encoded using automatically computable linguistic features. If they can, then this helps to relieve the burden involved in porting systems such as these to new domains. Our preliminary cue phrase feature set includes syntactic features of the main verb (voice, tense, aspect, modality, negation). We also use sentence initial part-of-speech and sentence initial word features to roughly approximate formulaic expressions which are sentence-level adverbial or prepositional phrases. Subject features include the head lemma, entity type, and entity subtype. These features approximate

	NB			ME		
	P	R	F	P	R	F
Cue	34.9	21.5	26.6	66.6	15.2	24.8
Entities	30.7	26.4	28.4	66.8	15.4	25.1
Them. Words	32.2	26.9	29.3	68.6	15.7	25.5
Location	31.6	27.2	29.2	73.4	16.4	26.9
Quotations	31.2	27.7	29.4	71.7	17.4	28.0
Sent. Length	31.7	29.4	29.8	71.4	16.9	27.3

Table 1: Accuracy measures for yes predictions.

the hand-coded *agent* features of Teufel and Moens. A main verb lemma feature simulates Teufel and Moens’s *type of action* and a feature encoding the part-of-speech after the main verb is meant to capture basic subcategorisation information.

3 Experimental Results

Table 1 contains cumulative precision (*P*), recall (*R*) and f-scores (*F*) for the naïve Bayes (NB) and maximum entropy (ME) classifiers on the relevance classification task.¹ Though only the cue phrase feature set performs well individually, all feature sets contribute positively to the cumulative scores with the exception of sentence length for ME and quotation for NB. Both classifiers perform significantly better than a baseline created by selecting sentences from the end of the document, which obtains *P*, *R* and *F* scores of 46.7, 16.0 and 23.8. F-scores for the best feature combinations are similar to the partial results reported in Teufel and Moens [2002]. Taking the f-score as the best metric to optimise would lead us to choose NB.

However, a basic aspect of summarisation system design, especially a system that needs to be flexible enough to suit various user types, is that the size of the summary will be variable. For instance, students may need a 20 sentence summary containing, for example, quite detailed background information, to get the same information a judge would get from a 10 sentence summary. Furthermore, any given user might want to request a longer summary for a certain document. So, what we actually want to do is rate *how* relevant/extract-worthy a sentence is in such a way that will allow us to select sentences in rank order. Bearing this in mind, precision is probably the more important metric given that recall will be controlled by the size of the summary. So, ME with all but sentence length features actually appears to be the better approach for sentence extraction.

Since we need a ranking rather than a *yes/no* classification, this might actually be considered a regression task. However, due to the way the corpus was annotated, the target attribute is in fact binary. As both of our classifiers are probabilistic, we use $p(y = \text{yes}|\vec{x})$ as a way to rank sentences. To evaluate the ranking methods with respect to our binary gold standard, we use the point-biserial correlation coefficient (r_{pb}). Table 2 contains correlation coefficients between the gold standard *yes/no* classification and $p(y = \text{yes}|\vec{x})$ for naïve Bayes (NB)

¹Note that this is a strict evaluation that counts only *yes* predictions. Micro- and macro-averaging over *yes* and *no* predictions give e.g. f-scores of 87.6 and 67.3 respectively for ME.

	NB		ME	
	I	C	I	C
Cue	0.187	0.187	0.208	0.208
Entities	0.103	0.211	0.056	0.219
Them. Words	0.016	0.211	0.000	0.227
Location	0.104	0.229	-0.031	0.166
Quotations	0.092	0.233	0.093	0.187
Sent. Length	0.069	0.235	0.000	0.175

Table 2: Point-biserial correlation coefficients.

and maximum entropy (ME).² The I column has scores for the individual feature sets and the C column has cumulative scores. The correlation results are strikingly different for NB and ME. While NB successfully incorporates all features ($r_{pb} = 0.235$), ME performs best using only cue phrase, entity and thematic word features ($r_{pb} = 0.208$). For ME, the location feature set actually gives a negative correlation. Judging by these results, we would again be likely to choose NB.

4 Conclusions and Future Work

In this paper, we have presented work on the automatic summarisation of legal texts for which we have compiled a new corpus with annotation of rhetorical status, relevance and linguistic markup. We presented sentence extraction results in classification and ranking frameworks. Naïve Bayes and maximum entropy classifiers achieve significant improvements over the baseline according to standard accuracy measures. We have also used the point-biserial correlation coefficient for quantitative evaluation of our extraction system, the results of which suggest different optimal configurations. In current work, we are developing a user study that will help determine empirically whether correlation coefficients are a better evaluation metric than precision and recall accuracy measures.

References

- [Fayyad and Irani, 1993] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *IJCAI*, 1993.
- [John and Langley, 1995] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *UAI*, 1995.
- [Kupiec *et al.*, 1995] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *SIGIR*, pages 68–73, 1995.
- [Malouf, 2002] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *CoNLL*, 2002.
- [Teufel and Moens, 2002] S. Teufel and M. Moens. Summarising scientific articles- experiments with relevance and rhetorical status. *Computational Linguistics*, 28(4):409–445, 2002.
- [Wolf and Gibson, 2004] F. Wolf and E. Gibson. Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance. In *ACL*, 2004.

²It has been argued that this is actually a better evaluation than standard accuracy measures, which do not account for degree of agreement [Wolf and Gibson, 2004].

Fault-Tolerant Context-Based Interpretation of Mathematical Formulas

Helmut Horacek and Magdalena Wolska

Fachrichtung Informatik Fachrichtung Computerlinguistik
Universität des Saarlandes, Postfach 15 11 50, D-66041 Saarbrücken, Germany
{horacek@ags,magda@coli}.uni-sb.de

Abstract

We present a fault-tolerant formula interpreter that aims at finding plausibly intended, formally correct and contextually meaningful specifications from user statements containing formal inaccuracies.

1 Motivation

Strong efforts are invested in supporting scientific publication environments in formal areas such as mathematics, with powerful tools, including semantically-informed editors and proof-checkers. Similarly, several research activities aim at teaching reasoning and problem-solving skills in tutorial systems. In both settings, effective communication is frequently hindered by sloppiness and low-level errors in specifying formulas on behalf of the user, resp. student. Hence, not only diagnosing errors in formulas, but also building hypotheses for correcting errors is an important issue, a difficult subproblem in mathematical knowledge management according to [2].

2 Our Working Context

Our interest in fault-tolerant formula interpretation originates from work on a tutorial system which teaches proving mathematical theorems [1]. In this context, a corpus of written interactions with a simulated computer-based tutor in the domain of naive set theory has been collected [5]. In this corpus, lengthy clarification subdialogs are often carried out to resolve low-level issues, which one wants to avoid since they distract the student from the higher-level tutorial goal.

In Table 1, we present examples of flawed formulas from the corpus. In (1), a segmentation error is shown: not only a space between the operator symbol P and identifier C , but also parentheses are missing. (2) is an example of a typing error, where an operator symbol p has been used in place of the identifier b . In (3), the types of arguments of the main operator are invalid. (4) shows a well-formed formula but it is not relevant in the context of the task: a stronger assertion about an intersection rather than union of the sets on the right-hand side of the equation is required¹. In (5), similarly to (4), a weaker operator of set inclusion ($\subseteq$) rather than equality is correct. Finally, (6) and (7) are examples of commonly confused relations of *subset* and *membership*.

¹This is the only kind of error which we do not address here.

Table 1: Examples of flawed formulas from the corpus

	Example Formula	Error Category
(1)	$P((A \cup C) \cap (B \cup C)) = PC \cup (A \cap B)$	3
(2)	$(p \cap a) \in P(a \cap b)$	2
(3)	$(x \in b) \notin A \quad x \subseteq K(A)$	2
(4)	$P((A \cup C) \cap (B \cup C)) = P(A \cup C) \cup P(B \cup C)$	1
(5)	$P((A \cap B) \cup C) = P(A \cap B) \cup P(C)$	1
(6)	$(A \cap B) \subseteq P(A \cap B)$	1
(7)	If $A \subseteq K(B)$ then $A \notin B$	2

In the context of the tutorial system mentioned above, the proof development environment Ω MEGA [3] is used to check validity of (possibly ambiguous) proof-step interpretations and consistency with the proof context. Communication with Ω MEGA is mediated by a *proof manager* whose task is to build and maintain a representation of the constructed proof. The proof manager also maintains a discourse memory of identifiers, operators, and their properties (e.g., type, arity).

3 Error Categories and Correction Attempts

Finding purposeful changes in a formula that aim at building a corrected and possibly intended version of that formula is substantially guided by its *correctness state*. We distinguish three categories of errors: 1) *logical*, 2) *type*, and 3) *structural* errors. A formula analyzer may find it impossible to build a formula analysis tree on the basis of the defined constructors (category 3), or to resolve a type mismatch in an analysis tree built successfully (category 2). Only for correctly analyzed formulas, consulting the proof manager yields a distinction between a formula expressing a *wrong* (category 1) or a correct statement. In case of an error, attempts are undertaken to remedy the error by applying local and contextually justified modifications to the formula. In order to obtain meaningful changes, we associate a set of replacement rules with each error category, aiming to achieve an improvement of at least one category level. Each of these rules is justified by evidence about the plausibility of the given error occurring in the domain, the nature of the task, and user capabilities. Some rules developed on the basis of errors observed in the corpus, their associated error categories, and examples are illustrated in Table 2. Testing applications of these rules is distributed between formula analysis and formula modifying algorithms.

Table 2: Replacement rules attempting to remedy errors

Replacement Rules	Error Categories	Examples
dual operator	1	$\cap \Leftrightarrow \cup, \subset \Leftrightarrow \supset, \subseteq \Leftrightarrow \supseteq$
stronger/weaker operator	1	$\supset \Leftrightarrow \supseteq, \subseteq \Leftrightarrow =, \supseteq \Leftrightarrow =$
confused operator	1,2	$\subset \Leftrightarrow \in, K \Leftrightarrow P$
confused identifier	1,2	$a \Leftrightarrow b, p \Leftrightarrow b, p \Leftrightarrow a$
delete character	2	$PC \Rightarrow P, PC \Rightarrow C$
insert blank or brackets	3	$PC \Rightarrow P C, PC \Rightarrow P(C)$

4 The Enhanced Formula Analysis

Formula analysis is part of an input interpretation component whose task is to produce a representation that can be further analysed by the proof manager. The formula modification procedure presented here extends the method of parsing mathematical expressions embedded within natural language text [4]. Extended formula analysis consists of: (1) identification of mathematical expressions within the word-tokenized text; (2) verification of the identified sequence as to syntactic validity and correction, in case of a parentheses mismatch or further tokenization needed;² (3) parsing of the expression.

1. The mathematical expression tagger uses a list of operation and identifier symbols. Identification of mathematical expressions is based on: single character tokens (including parenthesis), multiple-character tokens consisting only of known relevant characters, mathematical symbol unicodes, and new-line characters. Multiple-character candidate tokens are further segmented into operators and identifiers by inserting the missing spaces.
2. Once a candidate string is identified, “chains” of formulas are separated into individual segments, and parentheses match is verified. Missing parentheses are inserted while obeying to the following preferences: (i) provide parentheses for operators that require bracketed arguments, (ii) avoid redundant parentheses (i.e. double parentheses around the same substring).
3. Syntactically correct candidate strings are parsed by a tree-building algorithm accessing standard requisite information (e.g. list of operators and their precedence).

The output is a set of formula trees with nodes marked as to type compatibility and bracketing where applicable.

5 The Formula Modifying Algorithm

The algorithm incrementally generates formula modifications in a best-first fashion and tests their impact on resolving the original error, within specified resource limitations, that is, a maximum number of modified formulas and a time limit:

1. If the formula analyzer has produced a set of interpretations by applying rules addressing syntactic errors, these modifications are taken as successors of the original formula. Otherwise, successors are generated for the formula considered best by applying replacement rules in the category associated with the error reported for the original formula, unless the error is already resolved.

²This implements replacement rules of category 3. Parentheses correction may yield multiple plausible interpretations. From this stage on, further processing is applied to all the interpretations.

2. If resources are not exceeded, the most promising successor of a formula generated so far is promoted into the new best one, and its correctness state is assessed by the proof manager, continuing at step 1. Otherwise, an ordered list of modified formulas examined is returned.

Preferred orderings among created formulas are established by the error-related category and a similarity-assessing function, the former dominating the latter. The assessment function combines the number of replacement rules applied and the differences to the structurally most similar formula in the context, which comprises the set of formulas consisting of the goal expression, the previous proof step and possible follow-up steps. This function is approximated by simply adding the number of rule applications to the number of operators and variables appearing in both formulas compared, multiple occurrences counted according to their frequency.

6 Preliminary Results

Finally, we give analyses of some examples from Table 1. For utterance (1), we get two interpretations, depending on whether the formula analyzer separates PC and inserts parentheses (2 alternatives), or whether it reports a type error, flagging PC . In the latter case, replacing PC by any type compatible identifier yields an error of category 1. The same holds for the parenthesis insertion with narrower scope, $P(C)$, but the other alternative, $P(C \cup (A \cap B))$ yields no error and wins. Example (2) is relatively simple, since only replacing the occurrence of p , flagged as a type clash, is to be changed. Only replacements by a and b yield no error, b winning over a since it gets a better context agreement score. In utterance (5), many rules are applicable. Changing variables gives lower agreement scores than changing an operator to its dual counterpart, but all these choices remain within error category 1. Only replacing “=” by “ $\supseteq$ ” resolves the error.

In the future, we intend to look into other subdomains of mathematics and to incorporate results of formula modifications in tutorial strategies, e.g., asking for clarification, offering a small set of modified formulas. Similar interactions could be also tailored for scientific publication environments.

References

- [1] C. Benz Müller et al. Tutorial Dialogs on Mathematical Proofs. In *Proc. of IJCAI Workshop on Knowledge Representation and Automated Reasoning for E-Learning Systems*, pages 12–22, Acapulco, Mexico, 2003.
- [2] B. Buchberger and K. Nakagawa. Mathematical Knowledge Editor: A Research Plan. SFB Report 2004-40, Johannes Kepler University Linz, 2004.
- [3] J. Siekmann et al. Proof Development with Ω MEGA. In *Proc. of CADE-02*, pages 144–149, Copenhagen, Denmark, 2002.
- [4] M. Wolska and I. Kruijff-Korbyová. Analysis of Mixed Natural and Symbolic Language Input in Mathematical Dialogs. In *Proc. of ACL-04*, pages 24–32, 2004.
- [5] M. Wolska et al. An Annotated Corpus of Tutorial Dialogs on Mathematical Theorem Proving. In *Proc. of LREC-04*, pages 1007–1010, Lisbon, Portugal, 2004.

MDL-based Acquisition of Substitutability Relationships between Discourse Connectives

Ben Hutchinson

University of Edinburgh

School of Informatics

2 Buccleuch Place, Edinburgh, EH8 9LW, UK

B.Hutchinson@sms.ed.ac.uk

Abstract

Knowledge of which lexical items convey the same meaning in a given context is important for many Natural Language Processing tasks. This paper concerns the substitutability of discourse connectives in particular. This paper proposes a data-driven method based on a Minimum Description Length (MDL) criterion for automatically learning substitutability of connectives. The method is shown to outperform two baseline classifiers.

1 Introduction

Discourse connectives are words or phrases which explicitly signal coherence relations in texts, e.g. *however*, *even though* and *whereas*. Studies of connectives have provided insights into natural language discourse, however formal semantic analysis of connectives is a difficult task, and most connectives have not been the subject of much study. The empirical study of the substitutability of discourse connectives can inform and complement theoretical analyses [Knott, 1996]. The substitutability of discourse connectives is also important for Natural Language Processing applications such as Natural Language Generation and text simplification [Moser and Moore, 1995; Siddharthan, 2003].

This paper investigates acquiring substitutability relationships between connectives automatically. Previous attempts to do so have been hindered by the relatively low prior likelihood of two connectives being substitutable [Hutchinson, 2005]. We introduce an MDL-based approach that addresses this.

2 Substitutability of discourse connectives

Example (1) contains the connective *seeing as*. However, *because* can also be used to achieve the same discourse aims, i.e. it is **substitutable** [Knott, 1996].

- (1) **Seeing as/because** we've got nothing but circumstantial evidence, it's going to be difficult to get a conviction.

However, *because* it is not always substitutable for *seeing as*, and the converse does not hold either. In general, there are five possibilities for a connective X 's relationship to another connective Y :

- X is a **SYNONYM** of Y if X can always be substituted for Y , and vice versa.
- X and Y are **EXCLUSIVE** if neither can ever be substituted for the other.
- X is a **HYPONYM** of Y if Y can always be substituted for X , but not vice versa.
- X is a **HYPERNYM** of Y if X can always be substituted for Y , but not vice versa.
- X and Y are **CONTINGENTLY SUBSTITUTABLE** if each can sometimes, but not always, be substituted for the other.

We would like to automatically predict which of these relationships holds, for any given pair of connectives.

3 An MDL-based model of substitutability

We will refer to a consistent set of relationships between connectives as a *taxonomy*. Our model of taxonomies exploits the following two observations. Firstly, it is unlikely that every connective should be **EXCLUSIVE** with every other one. Secondly, there are logical constraints on substitutability. For example, if A is a **HYPONYM** of B and B is **EXCLUSIVE** with C , then A must also be **EXCLUSIVE** with C .

Our modelling of substitutability is within the Minimum Description Length (MDL) framework. In our case, we wish to evaluate models representing substitutability relationships between connectives. For a taxonomy $\mathbb{T}$ and data $data$, the total description length is given by:

$$L(\mathbb{T}, data) = L(\mathbb{T}) + L(data|\mathbb{T}) \quad (2)$$

We will exploit the fact that description lengths L can be related to probabilities P via the equation $L = -\log_2 P$.

Calculating the prior: We will use the notation $\langle rel, X, Y \rangle \in \mathbb{T}$ to indicate that in the taxonomy $\mathbb{T}$ connectives X and Y are in relationship rel . We calculate $P(\mathbb{T})$ using the following multinomial model:

$$P(\mathbb{T}) = M \prod_{\langle rel, X, Y \rangle \in \mathbb{T}} P(rel) \quad (3)$$

where (i) $P(rel)$ is the prior probability of two connectives being in the relationship rel , which will be estimated empirically, and (ii) M is a multinomial coefficient which ensures

that the most likely taxonomy contains numbers of each pairwise substitutability relationship in proportion to their prior probabilities.

The multinomial model (3) is defined over all sets of pairwise substitutability relationships, but we are only interested in calculating the probabilities of consistent sets. However this does not affect the relative likelihoods of taxonomies with the same number of connectives, so for the purposes of the experiments below this effect can be ignored.

Estimating the posterior probability: Previous work has found a correlation between substitutability and distributional similarity [Hutchinson, 2005]. Therefore, the data that our model of taxonomies aims to explain are the Kullback-Leibler (KL) divergences, $D(p||q)$, between pairs of connectives. That is, we assume that the data to be explained is:

$$data \equiv \{D(X||Y) : \langle rel, X, Y \rangle \in \mathbb{T}\}$$

To estimate the probability of the data, we assume that the likelihood of observing a given distributional divergence $D(X||Y)$ between X and Y is dependent only on the substitutability of X and Y .

$$P(data|\mathbb{T}) \approx \prod_{\langle rel, X, Y \rangle \in \mathbb{T}} P(D(X||Y)|\langle rel, X, Y \rangle) \quad (4)$$

To estimate each of the multiplicands in (4), we use Gaussian models of the distributional divergences corresponding to each substitutability relationship, i.e.

$$P(D(X||Y)|\langle rel, X, Y \rangle) \propto n(D(X||Y); \mu_{rel}, \sigma_{rel}) \quad (5)$$

where μ_{rel} and σ_{rel} are the mean and standard deviation of KL divergences of all pairs of connectives in relationship rel .

4 Experiment

Task: Our task is to predict the substitutability relationships in a manually constructed taxonomy $\mathbb{T}$ of connectives. We iteratively remove a single connective from $\mathbb{T}$ and attempt to re-insert it in its original position. To do this, we insert it in the position that minimises the description length.

Methodology: An existing taxonomy containing 80 discourse connectives was used as a gold standard [Hutchinson, 2005]. Fixed width beam search (width=1000) is used to search for the taxonomy with minimum description length. Co-occurrences with other discourse connectives were used as the distributional representations.

Parameter tuning: Due to unrealistic independence assumptions in Equation (4), our model underestimates the value of $P(data|\mathbb{T})$. As a result, it overestimates $L(data|\mathbb{T})$. To counterbalance this effect, we weight $L(data|\mathbb{T})$ by a parameter $\lambda \in (0, 1]$. Using half the pairs of connectives as a validation set, the setting $\lambda = 0.1$ was found to give good results.

Evaluation metrics: Two evaluation metrics were used: overall accuracy in predicting substitutability relationships, and the amount of information (measured in bits) of the correctly classified instances [Kononenko and Bratko, 1991].

Baselines: Two baseline classifiers were constructed for comparison with the MDL-based model. The first assigned all pairs to the most frequent relationship, namely EXCLUSIVE. The second baseline classifier assumed that the new connective would be a SYNONYM of one other connective X already in the taxonomy. All other relationships involving the new connective were thus identical to X 's. When evaluated on all of connectives, these classifiers had accuracies of 69.9% and 70.7%, and correctly learnt 2280 and 3333 bits of information, respectively.

Results: On the 50% of pairs of connectives not in the validation set, the MDL-based classifier had an accuracy of 73.8%, which is significantly greater than both baselines. The improvement is even greater on the information theoretic metric: correct classifications had 1928 bits of information, from which we can extrapolate a performance of about 3856 bits on the set of all pairs of connectives.

5 Discussion and conclusions

We have introduced a statistical model of the lexicon based on the Minimum Description Length principle. A multinomial prior assigns the greatest probability when the frequencies of pairwise relationships are in proportion to the prior likelihoods of those relationships. The model is applied to extending a taxonomy representing substitutability relationships between discourse connectives, and gives better performance than two classifiers that do not take into account global properties of the lexicon. The model aims to explain the distributional similarity of pairs of connectives, as measured by the Kullback-Leibler divergence function. In future, we will explore the use other distributional similarity functions, as these provide other perspectives of the data that may prove useful for learning substitutability. The utility of including other types of co-occurrence data in our distributional representations also remains to be explored.

References

- [Hutchinson, 2005] Ben Hutchinson. Modelling the substitutability of discourse connectives. In *Proceedings of the 43rd Annual Meeting of The Association for Computational Linguistics*, Ann Arbor, USA, 2005.
- [Knott, 1996] Alistair Knott. *A data-driven methodology for motivating a set of coherence relations*. PhD thesis, University of Edinburgh, 1996.
- [Kononenko and Bratko, 1991] Igor Kononenko and Ivan Bratko. Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6:67–80, 1991.
- [Moser and Moore, 1995] M. Moser and J. Moore. Using discourse analysis and automatic text generation to study discourse cue usage. In *Proceedings of the AAAI 1995 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 92–98, 1995.
- [Siddharthan, 2003] Advaith Siddharthan. Preserving discourse structure when simplifying text. In *Proceedings of the 2003 European Natural Language Generation Workshop*, pages 103–110, 2003.

Can we assign attitudes to a computer based on its beeps? —Toward an effective method for making humans empathize with artificial agents

Takanori Komatsu

Future University-Hakodate

School of Systems Information Science

Kamedanakano 116-2, Hakodate 041-8655, JAPAN

komatsu@fun.ac.jp

Abstract

Can we assign attitudes to a computer based on its beeps? If so, which kinds of beeps are perceived as specific attitudes, such as “disagreement”, “hesitation” or “agreement”? To examine this issue, I carried out an experiment to observe how participants perceive or assign an attitude to a computer according to beeps of different durations and F0 contour’s slopes. The results revealed that 1) beeps with increasing intonation regardless of duration were perceived by participants as “disagreement”, 2) flat sounds with longer duration were interpreted as “hesitation”, and 3) decreasing intonations with shorter duration were as “agreement.”

1 Introduction

The purpose of this study is to clarify an effective method for making users perceive artificial agents as having attitudes and empathize with them. Actually, many researchers have developed humanoid robots [for example, Breazeal and Velasquez, 1998] that can exist in the same physical space with users and life-like (or character) agents [for example, Morishima *et al.*, 1995] that appear on computer screens. These anthropomorphic agents are believed to facilitate communication between these agents and humans. Some researchers, however, have reported that humans perceive the attitudes of others from the information (e.g., humming without any language information), even if this information is simpler than that derived from gestures or facial expressions [for example, Suzuki *et al.*, 2004].

Therefore, one may reasonably assume that the simple information presented by an agent without a human-like shape or face would also cause people to perceive different attitudes and to assign specific attitudes to such agents. Moreover, this perception might induce the users’ sympathy toward the agent and eventually trigger smooth communication between them.

As a preliminary study aimed at achieving such a relationship between users and agents, I have focused on a normal laptop computer as the agent without a human-like shape or face, and on beeps (sound stimuli) as simple information presented to humans. Specifically, I conducted an experiment

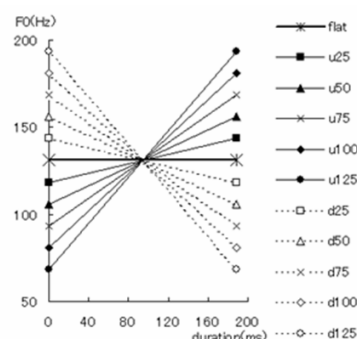


Figure 1. Eleven different F0 contours (duration; 189 ms): “u25” indicates that F0 transition range was 25 Hz with upward slope (increasing intonation).

in which various beeps were presented to participants from a laptop computer. The purpose of the experiment was to observe and analyze how participants assign specific attitudes to the computer based on the presented beeps.

2 Experiment

I prepared 44 different types of beeps with four different durations and 11 different F0 contours. Specifically, the four durations were 189, 418, 639, and 819 ms. The 11 F0 contours were set so that the transition range of F0 values that were linearly downward or upward (Figure 1). Participants were 23 Japanese (17 men and 6 women; 20 - 42 years old).

An experimenter asked participants to assume that these presented beeps were answering speeches in a telephone conversation (actually this was not true). As an experimental trial, one randomly selected beep among the 44 beeps was presented to participants. They were then asked to answer a question like “Did you feel that [***] was this computer’s attitude based on this beep?”; [***] was the randomly selected attitude among the following three that were complementary to each other. In addition, these three attitudes might correspond to valence evaluations; 1) Disagreement, e.g., surprising, doubting (*negative valence*), 2) Hesitation, e.g., being lost for words (*neutral valence*), 3) Agreement, e.g., acceptance (*positive valence*). Specifically, each participant experienced 132 trials (44 beeps x 3 attitudes).

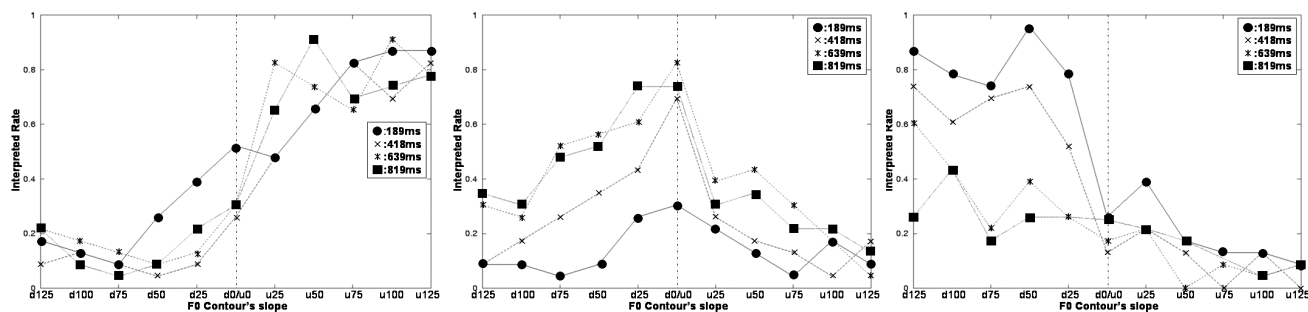


Figure 2. Relationship between F0 contours and rate of “disagreement” interpretation for each duration (left), rate of “hesitation” (center) and “agreement” (right).

3 Results

Case 1: Disagreement Figure 2 (left) shows which beeps were interpreted as the computer’s attitude of “disagreement.” In this figure, the horizontal axis lists the types of F0 contours and the labels on this axis, such as “u25” or “d100,” are the same ones used in Figure 1. The vertical axis shows the interpretation rates (IR) that give the proportion of participants answering “yes” in the corresponding trials. There are four lines indicating the duration of sound stimuli.

In the results, no significant differences were found among different durations ($F(3,66)=.70$, n.s.) and different F0 transition ranges in falling tones ($F(4,88)=1.68$, n.s.). On the other hand, rising tones showed a significant higher IR than falling tones ($F(1,22)=219.7$, $p<.01$), and wider F0 transition ranges in a rising tone showed higher IR ($F(4,88)=9.81$, $p<.01$). To sum up, **when the beeps with upward slopes (increasing intonation) regardless of their duration were presented from the computer, this computer’s attitude was interpreted as “disagreement.”**

Case 2: Hesitation Figure 2 (center) shows which beeps were interpreted as “hesitation.” Longer duration showed the higher IR ($F(3,66)=18.1$, $p<.01$), and falling tones showed the higher rate than the rising tones ($F(1,22)=11.64$, $p<.01$). Moreover, narrower F0 transition range in both rising and falling tones showed the higher IR than wider ones ($F(4,88)=11.23$, $p<.01$, $F(4,88)=8.89$, $p<.01$). In sum, **when the slower slopes (flat sounds) with longer duration were presented, the computer’s attitude was interpreted as “hesitation.”**

Case 3: Agreement Figure 2 (right) shows which beeps were “agreement.” Shorter duration showed the higher IR ($F(3,66)=19.42$, $p<.01$) and falling tones showed the significantly higher rate than rising tones ($F(1,22)=67.5$, $p<.01$). And wider F0 transition range in falling tone showed rather higher rates than narrower ones ($F(4,88)=3.12$, $p<.05$), and in rising tones, narrower transition range showed higher rates than wider ones ($F(4,88)=5.48$, $p<.01$). To sum up, **when the downward slopes with shorter duration were presented, the computer’s attitude was interpreted as “agreement.”**

4 Discussion and Conclusions

The results have established that different kinds of simple information “beeps” can be interpreted as presenting the different attitudes of an information sender “a laptop computer.” Therefore, sufficient possibilities exist to justify applying the results to an actual application. The same argument was presented by Lin and Picard [2003] in their “subtle expressivity” study. At first glance, these results seem to show that users can perceive only three agent attitudes; however, implementing the findings of this study into a laptop computer would create effective interactions with users. For example, when a computer responds to a certain action of users by presenting “beeps with increasing intonation,” users would intuitively assign a particular attitude (in this case “disagreement”) to the computer from this beep sound: “Umm... this computer seems to have some doubts about my last command...” And also implementing these results into a simple mobile robot would realize a *Star Wars R2D2 type robot*, which can not speak clearly but can express its attitudes by its electric sounds or behaviors. Thus, the results of this study can contribute to achieving an agent that enables users to assign or interpret intuitively and effectively particular agent attitudes.

References

- [Breazeal and Velasquez, 1998] Breazeal, C., and Velasquez, J. Toward teaching a robot ‘infant’ using emotive communication acts. In *Proceedings of ISAB workshop on Social Situated Intelligence*, pages 25-40, 1998.
- [Lin and Picard, 2003] Liu, K., and Picard, W. R. Subtle Expressivity in a Robotic Computer. In *Proceedings of the CHI2003 Workshop on Subtle Expressivity for Characters and Robot*, 2003.
- [Morishima et al., 1995] Morishima, S., Iwasawa, S., Sakaguchi, T., Kawakami, F., and Ando, M. Better Face Communication. In *Visual Proceedings of SIGGRAPH95*, pages 117, 1995.
- [Suzuki et al., 2004] Suzuki, N., Kakehi, K., Takeuchi, Y., and Okada, M. Social effects of the speed of hummed sounds on human-computer interaction. *International Journal of Human-Computer Studies*, 60, 455-468, 2004.

Correspondence-guided Synchronous Parsing of Parallel Corpora

Jonas Kuhn

The University of Texas at Austin, Department of Linguistics
jonask@mail.utexas.edu

Abstract

We present an efficient dynamic programming algorithm for synchronous parsing of sentence pairs from a parallel corpus with a given word alignment. Unless there is a large proportion of words without a correspondence in the other language, the worst-case complexity is significantly reduced over standard synchronous parsing. The theoretical complexity results are corroborated by a quantitative experimental evaluation.

Our longer-term goal is to induce monolingual grammars from a parallel corpus, exploiting implicit information about syntactic structure obtained from correspondence patterns.¹ Here we provide an important prerequisite for parallel corpus-based grammar induction: an efficient algorithm for synchronous parsing, given a particular word alignment (e.g., the most likely option from a statistical alignment).

Synchronous grammars. We assume a straightforward extension of context-free grammars (compare the *transduction grammars* of [Lewis II and Stearns, 1968]): (1) the terminal and non-terminal categories are pairs of symbols (or NIL); (2) the sequence of daughters can differ for the two languages; we use a compact rule notation with a numerical ranking for the linear precedence in each language. The general form of a rule is $N_0/M_0 \rightarrow N_1:i_1/M_1:j_1 \dots N_k:i_k/M_k:j_k$, where N_l, M_l are NIL or a (non-)terminal symbol for language L_1 and L_2 , respectively, and i_l, j_l are natural numbers for the rank in the sequence for L_1 and L_2 (for NIL categories a special rank 0 is assumed). Compare fig. 1 for a sample analysis of the German/English sentence pair *Wir müssen deshalb die Agrarpolitik prüfen/So we must look at the agricultural policy*. We assume a normal form in which the right-hand side is ordered by the rank in L_1 .² The formalism goes along with the **continuity assumption** that every complete constituent is continuous in both languages.³

Synchronous parsing. Our dynamic programming algorithm can be viewed as a variant of Earley parsing and generation, which again can be described by inference rules. For

instance, the central *completion* step in Earley parsing can be described by the rule⁴

$$(1) \frac{\langle X \rightarrow \alpha \bullet Y \beta, [i, j] \rangle, \langle Y \rightarrow \gamma \bullet, [j, k] \rangle}{\langle X \rightarrow \alpha Y \bullet \beta, [i, k] \rangle}$$

The input in synchronous parsing is not a one-dimensional string, but a pair of sentences, i.e., a two-dimensional array of possible word pairs (or a multidimensional array if we are looking at a multilingual corpus). The natural way of generalizing context-free parsing to synchronous grammars is thus to use string indices in both dimensions. So we get inference rules like the following (there is another one in which the i_2/j_2 and j_2/k_2 indices are swapped between the two items above the line):

$$(2) \frac{\langle X_1/X_2 \rightarrow \alpha \bullet Y_1:r_1/Y_2:r_2 \beta, [i_1, j_1, j_2, k_2] \rangle, \langle Y_1/Y_2 \rightarrow \gamma \bullet, [j_1, k_1, i_2, j_2] \rangle}{\langle X_1/X_2 \rightarrow \alpha Y_1:r_1/Y_2:r_2 \bullet \beta, [i_1, k_1, i_2, k_2] \rangle}$$

Since each inference rule contains six free variables over string positions ($i_1, j_1, k_1, i_2, j_2, k_2$), we get a parsing complexity of order $O(n^6)$ for unlexicalized grammars (where n is the number of words in the longer of the two strings from L_1 and L_2) [Wu, 1997; Melamed, 2003].

Correspondence-guided parsing. As an alternative to standard “rectangular indexing” we propose an asymmetric approach: one of the languages (L_1) provides the “primary index” – the string span in L_1 like in monolingual parsing. As a secondary index, L_2 contributes a chart-generation-style bit vector of the words covered, which is mainly used to guide parsing – i.e., certain options are eliminated. A complete sample index for *müssen/must* in fig. 1 would be $\langle [1, 2], [00100000] \rangle$. Completion can be formulated as inference rule (3).⁵ Condition (iii) excludes discontinuity in passive chart items, i.e., complete constituents; active items (i.e., partial constituents) may well contain discontinuities.

$$(3) \frac{\langle X_1/X_2 \rightarrow \alpha \bullet Y_1:r_1/Y_2:r_2 \beta, \langle [i, j], \mathbf{v} \rangle \rangle, \langle Y_1/Y_2 \rightarrow \gamma \bullet, \langle [j, k], \mathbf{w} \rangle \rangle}{\langle X_1/X_2 \rightarrow \alpha Y_1:r_1/Y_2:r_2 \bullet \beta, \langle [i, k], \mathbf{u} \rangle \rangle} \quad \text{where}$$

(i) $j \neq k$; (ii) $\text{OR}(\mathbf{v}, \mathbf{w}) = \mathbf{u}$;
(iii) $\mathbf{w}$ is continuous (i.e., it contains maximally one subsequence of 1's).

¹Cp. the new PTOLEMAIOS project at Saarland University (<http://www.coli.uni-saarland.de/~jonask/PTOLEMAIOS/>).

²However, categories that are NIL in L_1 come last. If there are several, they are viewed as unordered with respect to each other.

³As [Melamed, 2003] discusses, such an assumption is empirically problematic with binary grammars. However, if flat analyses are assumed for clauses and NPs, the typical problematic cases are resolved.

⁴A chart item is specified through a position ($\bullet$) in a production and a string span ($[l_1, l_2]$). $\langle X \rightarrow \alpha \bullet Y \beta, [i, j] \rangle$ is an active item recording that between position i and j , an incomplete X phrase has been found, which covers α , but still misses $Y\beta$. Items with a final $\bullet$ are called passive.

⁵We use the bold-faced variables $\mathbf{v}, \mathbf{w}, \mathbf{u}$ for bit vectors; OR performs bitwise disjunction on the vectors.

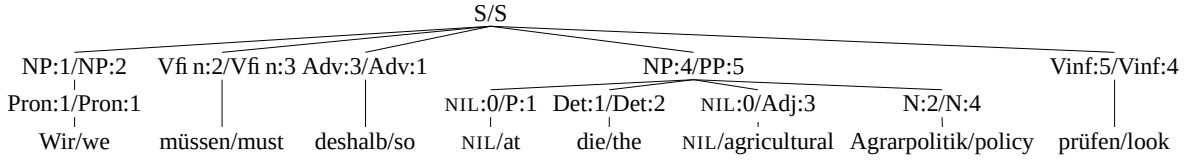


Figure 1: Sample analysis for a synchronous grammar

Parsing is successful if an item with index $\langle [0, N], 1 \rangle$ can be found for the start category pair (where N is the length of the L_1 -string).

Words in L_2 with no correspondent in L_1 (let's call them " L_1 -NIL"s for short) can in principle appear between any two words of L_1 . Therefore they are represented with a "variable" empty L_1 -string span like for instance in $\langle [i, i], [00100] \rangle$. But note that due to the continuity assumption, the distribution of the L_1 -NILs is constrained by the other words in L_2 , as exploited in the inference rule:

$$(4) \quad \frac{\langle X_1/X_2 \rightarrow \alpha \bullet \text{NIL:0}/Y_2:r_2 \quad \beta, \langle [i, j], \mathbf{v} \rangle \rangle, \quad \text{where} \quad \langle \text{NIL}/Y_2 \rightarrow \gamma \bullet, \langle [j, j], \mathbf{w} \rangle \rangle}{\langle X_1/X_2 \rightarrow \alpha \text{NIL:0}/Y_2:r_2 \bullet \beta, \langle [i, j], \mathbf{u} \rangle \rangle}$$

(i) $\mathbf{w}$ is adjacent to $\mathbf{v}$ (i.e., unioning vectors $\mathbf{w}$ and $\mathbf{v}$ does not lead to more 0-separated 1-sequences than $\mathbf{v}$ contains already);
(ii) $\text{OR}(\mathbf{v}, \mathbf{w}) = \mathbf{u}$.

The rule has the effect of finalizing a cross-linguistic constituent after all the parts that have correspondents in both languages have been found.

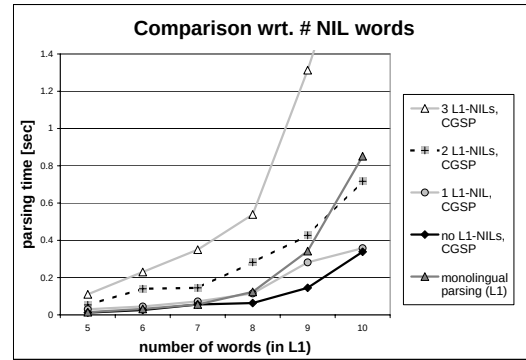
Complexity. We assume that the two-dimensional chart is initialized with the correspondences following from a word alignment. Hence, for each terminal that is non-empty in L_1 , both components of the index are known. When two items with known secondary indices are combined with (3), the new secondary index can be determined with minimal expense. Thus, for sentence pairs without any L_1 -NILs, the worst-case complexity for synchronous parsing is identical to the monolingual case of context-free parsing (i.e., $O(n^3)$). The average parsing expense in the absence of L_1 -NILs is even lower than in monolingual parsing: certain hypotheses for complete constituents are excluded because the secondary index reveals a discontinuity.

The complexity is increased by the presence of L_1 -NILs, since with them the secondary index can no longer be uniquely determined. However, with the adjacency condition ((i) in rule (4)), the number of possible variants in the secondary index is a function of the number of L_1 -NILs. Say there are m L_1 -NILs. In each application of rule (4) we pick a vector $\mathbf{v}$, with a variable for the leftmost and rightmost L_1 -NIL element. By adjacency, either the leftmost or rightmost one marks the boundary for adding the additional L_1 -NIL element NIL/Y_2 – hence we need only one new variable for the newly shifted boundary among the L_1 -NILs. So, in addition to the n^3 expense of parsing non-nil words, we get an expense of m^3 for parsing the L_1 -NILs, and end up in $O(n^3 m^3)$. Since typically the number of correspondent-less words is significantly lower than the total number of words, these results are encouraging for medium-to-large-scale grammar learning experiments.⁶

⁶The idealizing assumption of a single, deterministic word align-

Empirical Evaluation. To validate empirically that the *average* parsing complexity for the proposed correspondence-guided synchronous parsing approach (CGSP) for sentences without or with few L_1 -NILs is lower than for standard monolingual parsing, we did a prototype implementation of the algorithm and ran a comparison. A synchronous grammar was extracted (and smoothed) from a manually aligned German/English section of the Europarl corpus. The results are shown as the black line (for the CGSP approach on sentences without L_1 -NILs) and dark gray line (for monolingual parsing) in (5). The diagram shows the average parsing time for sentence pairs of various lengths. Note that CGSP takes clearly less time.

(5) Synchronous parsing with a growing number of L_1 -NILs



(5) also shows comparative results for parsing performance on sentences that do contain L_1 -NILs (curves for 1, 2 and 3 L_1 -NILs are shown). Here too, the theoretical results are corroborated that with a limited number of L_1 -NILs, the CGSP is still efficient.

We also simulated a synchronous parser which does not take advantage of a given word alignment. For sentences of length 5, this parser took an average time of 22.3 seconds (largely independent of the presence/absence of L_1 -NILs).

References

- [Lewis II and Stearns, 1968] P. M. Lewis II and R. E. Stearns. Syntax-directed transduction. *Journal of the Association of Computing Machinery*, 15(3):465–488, 1968.
- [Melamed, 2003] I. Dan Melamed. Multitext grammars and synchronous parsers. In *Proceedings of NAACL/HLT*, 2003.
- [Wu, 1997] Dekai Wu. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3):377–403, 1997.

ment is not realistic (as pointed out by a reviewer). Consideration of various possible alignments may lead to a worst-case combinatorial explosion. It is an empirical question for the future whether an effective heuristic can be found for narrowing down the space of alignments that have to be considered.

Naturalness of an Utterance Based on the Automatically Retrieved Commonsense

Rafal Rzepka, Yali Ge and Kenji Araki

Graduate School of Information Science and Technology, Hokkaido University

Language Media Laboratory

Kita-ku Kita 14 Nishi 9, 060-0814 Sapporo, Japan

kabura,geyali,araki@media.eng.hokudai.ac.jp

Abstract

In this research we investigated user's behavior while facing a system coping with common knowledge about keywords and compared it with not only classic word-spotting method but also with random text-mining. We show how even a simple implementation of our idea can enrich the conversation and increase the naturalness of computer's utterances. Our results show that even very commonsensical utterances are more natural than classic approaches and also methods we developed to make a conversation more interesting. For arousing opinion exchange during the session, we will also briefly introduce our idea of combining latest NLP achievements into one holistic system where the main engine we want to base on commonsense processing and affective computing.

1 Introduction

During one of the NLP community meetings in Japan we attended a discussion about evaluating the freely (meaning "in the open domain") talking computer systems. The participants agreed that such systems are almost ignored by computer scientists while so called "chatter-bots" as Parry [Colby, 1975] or ELIZA [Weizenbaum, 1976] seem to be very often the only known yield of AI for other scientists and non-scholars. But today, even if passing the Turing test is still questionable, we have faster and faster access to massive corpora, which we believe may incline researchers to rethink the need of naturally talking systems and we claim they might be done without much labor and cost. Such renaissance would not only be important for the public recognition of the field but also could bring more young scholars fascinated by the fact that Internet has a chance to replace programmer's will and change the program's behavior depending on the facts discovered in millions of web-pages. Already Rzepka et al.[?] has noticed such possibility proposing an automatically developing average personality of "Mr. Internet" and has confirmed that commonsense knowledge can be retrieved from the WWW. We want to go a step further and show that using even simple usage of commonsensical (S)VO-then-V (Verb-Object and following Verb) and (S)VO-if-(S)VO (Subjects vary depending on the conversation processing so ig-

nored here) phrases text-mined from homepages improve the user's acceptance for talking systems.

2 Commonsense Knowledge Retrieval

2.1 Collected Data Structure

For the set of experiments we performed, we used only nouns as for decades they were usually the main part of open-domain chat programs, although we claim that the commonsensical context is the future of dialogue processing. Using a nouns list and Larbin robot we created 1,907,086 sentences WWW corpus which was a base for a verb, a noun and VO n-gram dictionaries. The noun and verb dictionaries consist of 79,460 verbs and 134,189 nouns extracted with help of ChaSen Analyzer. For creating VO phrases automatically, our system had to search for the relationships between verbs and nouns and also between verbs. In this step, we used the verbs and nouns which had the highest occurrence and are common, as they are used in everyday live, for example [pour/drink/melt]-water, [listen/switch on/enjoy]-music or [go to/buy at/enter]convenient store. We used Japanese language, which has useful grammar features like *rentaikei* where the verb suffix *te* usually joins verbs in a time sequence e.g. *gohan wo tabe-te neru* (to sleep after having a dinner) or *tara, eba* and *to* "if" forms which are able to distinguish different causal connotations. By these useful grammar features we are able to web-mine commonsensical knowledge as "it is usual that some people buy sweets at convenient store even if they didn't wanted". Until now such data had to be collected manually [Singh, 2002] but full automatizing of such knowledge collecting brings new opportunities not only for dialogue but also for storytelling, question answering, machine translation and many other fields.

2.2 Architecture Overview

Basically, our system's architecture for creating commonsensical data can be summarized into the following processing steps:

- a) A noun of is assigned for a keyword;
- b) The system uses our web corpus for frequency check to retrieve 3 most frequent verbs following the keyword noun;
- c) The most frequent particle between noun keyword and 3 most frequent verbs is discovered;

- d) For creating bi-gram the system retrieves a list of most frequent verbs occurring after the previously chosen verb;
- e) By using Yahoo search engine, the system checks if the noun-particle unit occurs with new verb-verb unit for time-sequence actions and verb-if unit for casual dependencies;
- f) If yes - the VO-then-V and VO-if-VO units are stored:

$$\text{VO}_{\text{then}} \text{V} = \text{N} + \text{P}_{\text{max}} + \text{V}_{\text{max1}} + \text{V}_{\text{max2}}$$

N :Triggering noun (keyword);

P_{max} :most frequent particle joining noun and verb;

V_{max1} :most frequent verb occurring after the N ;

V_{max2} :most frequent verb occurring after V_{max1} ;

$$\text{VO}_{\text{if}} \text{V} = \text{N}_1 + \text{P}_{1\text{max}} + \text{V}_{1\text{max}} + \text{if} + \text{N}_2 + \text{P}_{2\text{max}} + \text{V}_{2\text{max}}$$

N_1 :Triggering noun (keyword);

$P_{1\text{max}}$:most freq. particle joining first noun with a verb;

$V_{1\text{max}}$:most freq. verb after the $N_1 P_1$;

N_2 :most freq. noun after $N_1 P_{1\text{max}} V_{1\text{max}}$ and “if”;

$P_{2\text{max}}$:most freq. particle joining N_2 and V_2 ;

$V_{2\text{max}}$:most freq. verb after $N_2 P_2$;

3 Experiments

In order to see user’s perception of the basic commonsense knowledge included in a utterance, we performed a set of experiments basically using four kinds of utterances following input with one noun *keyword*:

- ELIZA’s output [ELI] (input sentence structure changing to achieve different outputs)
- WWW random retrieval output [WRR] (a shortest of 10 sentences retrieved by using *keyword* and query pattern “did you know that?”)
- WWW commonsense retrieval output “high” [CS1] (sentences using common knowledge of highest usualness (most frequent mining results))
- WWW commonsense retrieval output “low” [CS2] (sentences using common knowledge of the lowest usualness (least frequent mining results)).

Typical ELIZA answer is “why do you want to talk about smoking” if the *keyword* is “smoking”. For the same *keyword* WRR retrieved a sentence “did you know that people wearing contact lenses have well protected eyes when somebody is smoking?”. An example of CS1 is “you will get fat when you quit smoking” and CS2 is “smoking may cause mouth, throat, esophagus, bladder, kidney, and pancreas cancers”. We selected 10 most common noun keywords of different kinds (water, cigarettes, subway, voice, snow, room, clock, child, eye, meal) not avoiding ones often used in Japanese idioms (voice, eye) to see if it influences the text-mining results. 13 referees were evaluating every set of four utterances in two categories – “naturalness degree” and “will of continuing a conversation degree” giving marks from 1 to 10 in both cases.

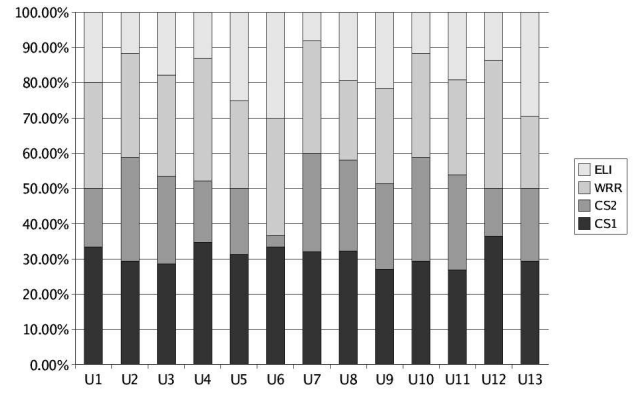


Figure 1: Naturalness Level Evaluation: CS1:29.5%, CS2:21%, WRR:24.5%, ELI: 25%

4 Results and Conclusions

In “continuation will degree” ELIZA unsurprisingly achieved 452 points out of 2919 for four systems (only 15.48%). But the performance of commonsenseal utterances was surprisingly high (CS1:25.38%, CS2:27.14%) which suggest that interlocutor prefers a machine saying “smoking is bad” than one naturally asking questions. The highest result of WRR (32%) tells us how simple tricks can help on keeping up the conversation. On the contrary, the naturalness degree results (see Fig.1) show that the “tricks” of ELIZA and WRR and information overload of CS2 are less natural than the ordinary truth statements. Due to the lack of space, more specific results analysis and graphs we are going to provide during the poster session.

References

- [Weizenbaum, 1976] Joseph Weizenbaum. *Computer Power and Human Reason*. W.H. Freeman and Co., New York, 1976.
- [Colby, 1975] Kenneth Colby. *Artificial Paranoia: A Computer Simulation of Paranoid Process* Pergamon Press., New York, 1975.
- [Rzepka et. al, 2003] Rafal Rzepka, Kenji Araki and Koji Tochinal. Ideas for the Web-Based Affective Processing. *Proceedings of the Seventh Multi-Conference on Systemics, Cybernetics and Informatics; vol. XIV Computer Science, Engineering Applications*, vol XIV:376–381, August 2003.
- [Singh, 2002] Push Singh. The public acquisition of commonsense knowledge. *Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*, Palo Alto, CA: AAAI, 2002.

Induction of Syntactic Collocation Patterns from Generic Syntactic Relations

Violeta Seretan

University of Geneva

Language Technology Laboratory

2 Rue de Candolle, Geneva, CH-1211, Switzerland

Violeta.Seretan@lettres.unige.ch

Abstract

Syntactic configurations used in collocation extraction are highly divergent from one system to another, this questioning the validity of results and making comparative evaluation difficult. We describe a corpus-driven approach for inferring an exhaustive set of configurations from actual data by finding, with a parser, all the productive syntactic associations, then by appealing to human expertise for relevance judgements.

1 Introduction

The term *collocation*, often used in different senses in the literature, is understood here as in the following statement: “The term collocation refers to the idiosyncratic syntagmatic combination of lexical items and is independent of word class or syntactic structure.” [Fontenelle, 1992, 222]. Cross-lingual examples, such as “*heavy smoker*” and “*pay attention*”, equivalent to “*grand fumeur*” and “*faire attention*” in French, show why collocations are crucial from the text encoding perspective: the lexical choice of the modifier and support verb is restricted by the conventional usage, while the alternatives are generally perceived as unnatural or “weird”¹.

Typically, collocation acquisition from corpora relies on statistical significance tests for pairs of words occurring close to each other. The recent developments in syntactic parsing allowed the move towards more linguistically-informed methods of extraction. Current systems rely increasingly on the syntactic pre-processing of source text (such as POS tagging, chunking, shallow or deep parsing) intended to support the identification of collocation candidates, prior to the statistical analysis.

Despite the advantages brought by syntax (well-formedness of results; drastic reduction of combinatorial complexity; partitioning of candidates into syntactically homogeneous classes), a serious problem arises: the arbitrariness in the choice of syntactic configurations (*patterns*) for collocation candidates. The rest of the paper discusses the implications of this problem and presents the solution we proposed in order to cope with it.

¹As, for instance: “*big smoker*”, “*make attention*”, or “*lourd fumeur*”, “*payer attention*”.

2 Syntactic Patterns for Collocations

In the most permissive case (when no extraction patterns are defined), any pair of words is regarded as a valid collocation candidate. This results in much noise, since the most frequent word combinations are also the least interesting (e.g., “of the”, “in the”). Therefore, many extraction systems perform the linguistic analysis of text and apply a linguistic filter on collocation candidates (very often, they perform POS tagging in order to filter out the pairs involving function words²).

There is unfortunately much disagreement with respect to the accepted syntactic configurations for collocations (often as a consequence of the lack of consensus in the understanding of the notion of collocation). There is much divergence, first, with respect to the POS of participating words. Some authors consider open-class words only ([Justeson and Katz, 1995; Hausmann, 1989]), while most of them allow for function words too (as in “agree on”). Second, the syntactic relations proposed are almost always different. The following list shows the diversity of the syntactic patterns used, for English, in various works:

1. Lexical collocations³ in BBI dictionary [Benson *et al.*, 1986]: N-V, N-A, N-V, N-P-N, A-Adv, V-Adv;
2. Hausmann’s collocation definition [1989]: N-A, N-V, V-N, V-Adv, A-Adv, N-[P]-N;
3. Xtract collocation extraction system [Smadja, 1993]: N-A, N-V, V-N, V-P, V-Adv, V-V, N-P, N-D;
4. WordSketch concordance system [Kilgariff *et al.*, 2004]: N-A, N-N, N-P-N, N-V, V-N, V-P, V-A, N-Conj-N, A-P;
5. FipsCo system [Goldman *et al.*, 2001]: N-A, N-N, N-P-N, N-V, V-N, V-P, V-P-N.

These examples show how much the perspectives adopted by various authors differ from the initial view, in which the collocation is seen as independent of the syntactic structure [Fontenelle, 1992, 222].

The arbitrariness in pattern choice questions the quality of the results and makes comparative evaluation difficult. Besides, a set of patterns established for a language may not completely fit another one. It is therefore necessary to find

²This restriction is seen as too strong [van der Wouden, 2001].

³The BBI dictionary also includes a wide range of “grammatical collocations”, such as: N-P, N-Conj, P-N, A-P, A-Conj, etc.

a means to establish, for each language, an exhaustive set of collocation patterns to be used as reference.

3 Pattern Induction Experiments

In order to overcome the problem of arbitrariness in pattern choice, we propose to make a corpus-driven investigation aimed at the discovery of all possible and interesting collocation configurations.

Rather than relying on linguistic prescriptions, we try to induce these configurations from actual data. We do not commit ourselves to pre-defined patterns; instead, we consider any POS combination as a priori possible, and we only require the items of a pair to be syntactically related. We use Fips, a GB-based parser [Wehrli, 2004], to extract such generic relations among words. Then we analyze the obtained results and infer the syntactic patterns from them.

More specifically, we consider the following generic relations: head-modifier, head-complement, verb-argument (both subject and objects). We extract the word pairs represented by the combination of a head with the lexical head of its specifier or of its complement (cf. GB theory).

Two experiments have been performed on English and French corpora of newspaper articles. Several statistics are shown in Table 1: size of corpora; number of word pairs in a generic relation (tokens); number of distinct pairs (types)⁴; and the number of POS combinations detected.

Experimental data	English	French
size (words)	0.5 M	1.6 M
word pairs (tokens)	0.18 M	0.75 M
word pairs (types)	0.07 M	0.17 M
POS combinations	60	57

Table 1: Generic relations extracted

The last row shows that many POS combinations are actually productive, from the total of 98 combinations (with V, N, A, Adv, P, D, Conj in either specifier and complement position). By manually inspecting the obtained POS associations we discovered new collocation patterns that were commonly ignored - especially the patterns involving non-content words. We here provide some examples:

1. English: N-P (*alliance between*), P-N (*across border*), V-Conj (*judge whether*), A-Adv (*mature enough*), Adv-Adv (*much more*), Adv-P (*together with*);
2. French: P-N (*sous pression*) N-P (*débat sur*), N-Conj (*temps que*), A-P (*prêt à*), V-A (*rester impassible*), P-Adv (*comme jamais*).

A more in-depth analysis of results is under way, which aims to identify the relevant configurations from the POS combinations found with our method.

4 Conclusion and Related Work

The performance of collocation extraction systems (in terms of accuracy and coverage) is highly dependent on the de-

gree of syntactical permissiveness. Too much permissiveness leads to the problems of noise and combinatorial explosion; too strong constraints risk not to capture the whole range of collocational phenomena. We proposed a trade-off between these extremes: we only maintained a minimal syntactic constraint (the presence of a syntactic link between collocation's items), then we induced the collocation patterns in a data-driven fashion. The experiments conducted revealed several new collocation patterns that involve closed-class words, and led us to support the claim made by van der Wouden [2001, 17], that "lexical elements of almost any class may show collocational effect".

We compare our work with that of Dias [2003], which attempts to overcome the pattern pre-definition problem by using combined statistics on sequences of adjacent words and their POS. Since this method ignores the sentence structure, it is still affected by combinatorial explosion. In contrast, our method is computationally tractable, and, in addition, allows us to capture long distance collocational pairs⁵.

References

- [Benson *et al.*, 1986] Morton Benson, Evelyn Benson, and Robert Ilson. *The BBI Dictionary of English Word Combinations*. John Benjamins, Amsterdam, 1986.
- [Dias, 2003] Gaël Dias. Multiword unit hybrid extraction. In *Proceedings of the ACL Workshop on Multiword Expressions*, pages 41–48, Sapporo, Japan, 2003.
- [Fontenelle, 1992] Thierry Fontenelle. Collocation acquisition from a corpus or from a dictionary: a comparison. *Proceedings I-II. Papers submitted to the 5th EURALEX International Congress on Lexicography in Tampere*, pages 221–228, 1992.
- [Goldman *et al.*, 2001] Jean-Philippe Goldman, Luka Nerima, and Eric Wehrli. Collocation extraction using a syntactic parser. In *Proceedings of the ACL Workshop on Collocations*, pages 61–66, Toulouse, France, 2001.
- [Hausmann, 1989] Franz Josef Hausmann. Le dictionnaire de collocations. In F. I. Hausmann *et al.*, editor, *Wörterbücher: ein internationales Handbuch zur Lexicographie. Dictionaries, Dictionnaires*, pages 1010–1019. de Gruyter, Berlin, 1989.
- [Justeson and Katz, 1995] John S. Justeson and Slava M. Katz. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1:9–27, 1995.
- [Kilgarriff *et al.*, 2004] Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. The Sketch engine. In *Proceedings of the Eleventh EURALEX International Congress*, pages 105–116, Lorient, France, 2004.
- [Smadja, 1993] Frank Smadja. Retrieving collocations from text: Xtract. *Computational Linguistics*, 19(1):143–177, 1993.
- [van der Wouden, 2001] Ton van der Wouden. Collocational behaviour in non content word. In *Proceedings of the ACL Workshop on Collocations*, pages 16–23, Toulouse, France, 2001.
- [Wehrli, 2004] Eric Wehrli. Un modèle multilingue d'analyse syntaxique. In A. Auchlin *et al.*, editor, *Structures et discours - Mélanges offerts à Eddy Roulet*, pages 311–329. Éditions Nota bene, Québec, 2004.

⁴The parser identifies all the instances of a lexeme pair, irrespectively of the surface realization.

⁵As for instance "submit proposal" in "a proposal which addressed various topics has been submitted".

Evaluating an NLG System using Post-Editing

Somayajulu G. Sripada and Ehud Reiter and Lezan Hawizy

Department of Computing Science

University of Aberdeen

Aberdeen, AB24 3UE, UK

{ssripada,ereiter,lhawizy}@csd.abdn.ac.uk

Abstract

Computer-generated texts, whether from Natural Language Generation (NLG) or Machine Translation (MT) systems, are often post-edited by humans before being released to users. The frequency and type of post-edits is a measure of how well the system works, and can be used for evaluation. We describe how we have used post-edit data to evaluate **SUMTIME-MOUSAM**, an NLG system that produces weather forecasts.

1 Introduction

In this paper we describe an evaluation technique, which looks at how much humans need to post-edit texts generated by an NLG system before they are released to users. Post-edit evaluations are common in machine translation [Hutchins and Somers, 1992], but we believe that ours is the first large-scale post-edit evaluation of an NLG system. Mitkov and An Ha [2003] reported a small scale post-edit evaluation of their NLG system.

The system being evaluated is **SUMTIME-MOUSAM** [Sripada et al, 2003], an NLG system, which generates weather forecasts from Numerical Weather Prediction (NWP) data. The forecasts are marine forecasts for offshore oilrigs. **SUMTIME-MOUSAM** is operational and is used by Weathernews (UK) Ltd to generate 150 draft forecasts per day, which are post-edited by Weathernews forecasters before being released to clients.

Time	Wind Dir	Wind Spd 10m	Wind Spd 50m	Gust 10m	Gust 50m
06:00	W	10.0	12.0	12.0	16.0
09:00	W	11.0	14.0	14.0	17.0
12:00	WSW	10.0	12.0	12.0	16.0
15:00	SW	7.0	9.0	9.0	11.0
18:00	SSW	8.0	10.0	10.0	12.0
21:00	S	9.0	11.0	11.0	14.0
00:00	S	12.0	15.0	15.0	19.0

Table 1. Weather Data produced by an NWP model for 12-Jun 2002

Table 1 shows a small extract from the NWP data for 12-06-2002, and Table 2 shows part of the textual forecast that SumTime-Mousam generates from the NWP data.

Field	Text
WIND(KTS) 10M	W 8-13 backing SW by mid afternoon and S 10-15 by midnight.
WIND(KTS) 50M	W 10-15 backing SW by mid afternoon and S 13-18 by midnight.

Table 2. Extract from **SUMTIME-MOUSAM** Forecast Produced from NWP data in Table 1.

Weathernews uses SumTime-Mousam to generate draft forecasts; we call them 'Pre-edit Texts'. The forecasters then post-edit them to produce 'Post-edit texts'. When the forecaster is done, the complete forecast is sent to the customer.

2 Post-Edit Evaluation

The evaluation was carried out on 2728 forecasts, collected during period June to August 2003. Each forecast was roughly of 400 words, so there are about one million words in all in the corpus.

For each forecast, we have the following data:

- Data: The final edited NWP data
- Pre-edit text: The draft forecast produced by **SUMTIME-MOUSAM**.
- Post-edit text: The manually post-edited forecast, which was sent to the client.
- Background information: includes date, location, and forecaster

The following procedure is performed automatically by a software tool:

- First, we break sentences up into phrases, where each phrase describes the weather at one point in time.
- The second step is to align phrases from the previous step as a preparation for comparison in the next step. Our alignment procedure first generates an exhaustive

list of possible alignments and uses a scoring scheme to select aligned phrases.

- The third step is to compare aligned phrases and label each pre-edit/post-edit pair as match, replace, add, or delete.

For example, A and B of Figure 1 are analyzed as in Table 3 where phrases are shown separated by a shaded row.

A. Pre-edit Text: SW 20-25 backing SSW 28-33 by midday, then gradually increasing 34-39 by midnight.
B. Post-edit Text: SW 22-27 gradually increasing SSW 34-39.

Figure 1. Example pre-edit and post-edit texts from the post-edit corpus

POS	A	B	label
Direction	SW	SW	match
Speed	20-25	22-27	replace
Conjunction	then	<none>	delete
Adverb	gradually	gradually	match
Verb	increasing	increasing	match
Direction	<none>	SSW	add
Speed	34-39	34-39	match
Time	by midnight	<none>	delete

Table 3. Detailed Edit Analysis

We processed 2728 forecast pairs (pre-edited and post-edited). These were divided into 73041 phrases. Out of these, the alignment procedure failed to align 7608 (10%) phrases. Out of the successfully aligned phrases, 43914 (60%) are perfect matches, and the remaining 21519 (30%) are mismatches. Table 4 summarizes the mismatches and suggests that the major problem is ellipsis. Most (25235 out of 35874, 70%) of these errors are deletions, where the forecaster deletes words SumTime-Mousam’s texts.

S. No.	Mismatch Type	Freq.	%
1.	Ellipses (word additions and deletions)	35874	65
2.	Data Related Replacements (range and direction replacements)	10781	20
3.	Lexical Replacements	8264	15
	Total	54919	

Table 4. Results of the evaluation showing summary categories and their frequencies

3 Discussion of Post-Edit Evaluations

We were attracted to post-edit evaluation because we believed that (A) people would only edit things that were clearly wrong; and (B) post-editing was an important usefulness metric from the perspective of our users. Looking back, (B) was certainly true. The amount of post-editing that generated texts require is a crucial component of

the cost of using **SUMTIME-MOUSAM**, and hence of the attractiveness of the system to users.

(A) however was perhaps less true than we had hoped. Wagner [1998] also described post-edited texts in MT as at times noisy. During the development of **SUMTIME-MOUSAM**, our analysis of manually written forecasts [Reiter and Sripada, 2002] had highlighted a number of “noise” elements that made it more difficult to extract information from such corpora. While collecting the post-edit data, we assumed that people would only post-edit mistakes, where the generated text was wrong or sub-optimal, and hence post-edit data would be better for evaluation purposes than corpus comparisons.

In fact, however, there were many justifications for post-edits. Some post-edits fixed problems in the generated texts (such as overuse of *then*); some post-edits refined/optimized the texts (such as using *for a time*); some post-edits reflected individual preferences (such as *easing* vs *decreasing*); and some post-edits were downstream consequences of earlier changes (such as introducing SSW before ‘34-39’ in B, in the example of Section 2). We wanted to use our post-edit data to improve the system, not just to quantify its performance, and we discovered that we could not do this without attempting to analyze why post-edits were made. Probably the best way of doing this was to discuss post-edits with the forecasters.

4 Conclusion

We have used analysis of post-edits, a popular evaluation technique in machine translation, to evaluate SumTime-Mousam, an NLG system that generates weather forecasts. While we encountered some problems, such as the need to identify why post-edits were made, on the whole we found post-editing to be a useful evaluation technique which gave us valuable insights as to how to improve our system.

References

- [Hutchins and Somers, 1992] John Hutchins and Harold L. Somers. *An Introduction to Machine Translation*, Academic Press, London, 1992.
- [Mitkov and An Ha, 2003] Ruslan Mitkov and Le An Ha. Computer-Aided Generation of Multiple-Choice Tests, In *Proc. of the HLT-NAACL03 Workshop on Building Educational Applications Using NLP*, Edmonton, Canada, pages 17-22, 2003.
- [Reiter and Sripada, 2002] Ehud Reiter and Somayajulu G. Sripada. Human Variation and Lexical Choice. *Computational Linguistics*, 28:545-553, 2002.
- [Sripada et al., 2003] Somayajulu G. Sripada, Ehud Reiter, and Ian Davy. SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator. *Expert Update*, 6(3):4-10, 2003.
- [Wagner, 1998] Simone Wagner. Small Scale Evaluation Methods In *Proc. of the Workshop on Evaluation of the Linguistic Performance of Machine Translation Systems at the KONVENS-98*, Bonn, pages 93-105, 1998.

Online Support System for Mediator Education

Takahiro Tanaka, Yoshiaki Yasumura, Daisuke Katagami, Katsumi Nitta

Tokyo Institute of Technology
Department of Computational Intelligence and Systems Science
4259 Nagatsuta, Midori-ku, Yokohama, Japan 226-8502
takat@ntt.dis.titech.ac.jp

Abstract

To settle the disputation, ADR (Alternative Disputation Resolution) has been becoming popular instead of the trial. However, to educate the mediation skill, much training is needed. In this paper, we introduce the overview of the online mediator education system. This system navigates the users by providing materials for decision making by referring to old cases. Users can communicate with each other by using avatars, and they can see the status of disputation in the form of several diagrams.

1. Introduction

These days, there are various types of troubles related to transactions on the Internet. Especially the number of troubles in online shopping and online auctions are on the increase. Since going through trials to argue is high cost and time-consuming, therefore it is difficult to provide effective remedy for these types of troubles. Prompt and low-cost mediations and arbitrations called the Alternative Disputation Resolution (ADR) are in the spotlight.

Non-legal professions can carry out mediations, however, as the demand increasing, we are lacking mediators. Trainee mediators not only need education on the procedures and skills for actual mediations but also need to practice with example cases [Katsh, 2000; Rule, 2002]. However, when mediators practice, they need constant attendance by instructors. They have to be at the same location at same time which is a great burden.

Given this factor, we have been researching to aim for developing an online mediator education system. This system does not only provide online argument environment, but also assists and educates the mediators by using case example base. It also suggests a new education model using the analyzed data accumulated by the system.

2. Online mediation education assistance system

Online mediations are carried out by connecting to the server through the Internet by the users using the argument interface. Mediator (host) and the two parties are present for the system. Educational assistance is done by the instructor suggesting the next possible remark and judging the mediator's skill by analyzing the accumulated mediation records statistically.

The main features of this system are the following three;

(1) To provide online mediation education environment

Exchanging remarks online, setting index to the remark contents and show them in diagrams and accumulate them in case example base

(2) To navigate mediations using case example base

During mediation practice, similar past mediation cases can be searched and the next remarks candidate are indicated to the student acting as a mediator.

(3) To analyze mediation skill using case example base

By analyzing mediation records in case example base, assess the mediation student's skill.



Figure 1: Argument Interface

Situated mediations are carried out with Argument Interface as shown in Figure 1. Users input "remark texts", "link data" and "avatar expressions".

The case data dealt in this study is natural-language texts representing conversations in mediations. To be able to search for similar situations from the conversations and to analyze statistically, we propose to index the remarks as follows;

Firstly, when instructors set questions, they should know what the point at issue is, for instance, in mediations for auction troubles, the following point at issues are to be included [Ashley, 1991; Aleven, 1997];

- f1: Whether or not the product auctioned was a defect
 - f11: Does it have any flaw?
 - f12: Is it broken?
- f2: Was there enough description for the product?
 - f21: Did the product description picture show the defect part?
 - f22: Did the description explain the defect-status well?

For each question set, we prepare around twenty points at issue for it beforehand. In addition, for each of them, we register which points at issue is on the topic when which term (or groups of term) appear. This extraction of wordlist is done by TF/IDF method by using mediation records in the case example base.

This allows one remark to have two data be indexed; (1) the point at issue in the remark and (2) the relation with the previous remark.

During mediation, we tend to be stuck what remarks to make next. In those cases, it would be helpful if we can refer to other people's remarks in similar situations picked up from the case example base, and by knowing how the mediation proceeded can help them refer to making next remark.

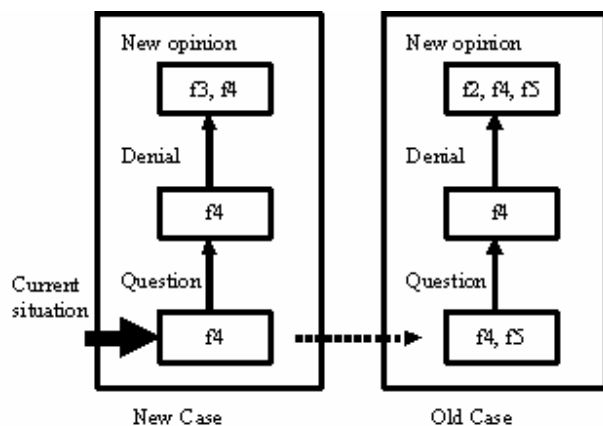


Figure 2: Similar Situation Search

However, searching similar situations are not easy. Similar remarks can always be slightly different from one another, as the expressions are different, and what we understand from them differs regarding the previous remarks.

To make searching for similar situations easier, we propose a searching method of using index of remarks. This method, as shown in Figure 2, goes back to two of the previous remarks from the present subject remark and within these three remarks, the point at issue and the relation (link data) are compared and similar situations are assessed.

3. Utilizing the case base

By analyzing old cases statistically, we can draw various information which is useful for legal education strategy. For example, by observing the number of issue points which the mediator introduced in the case, we can estimate how positively the mediator controls the mediation. This information is used to measure the mediation skill.

Another example is that we can classify the mediation data in the case base by comparing the flow of issue points. The classified data is used to arrange the mediation records and to evaluate the results of mediation training.

4. Conclusion

We introduced the outline of the online system of ADR mediation education assistance. The characteristics of this system are that it uses case example base to compare the mediation navigation and to assist mediator education.

The final object of this study is to develop a mediation agent that carries out mediations instead of human. Mediation agent acts as a host following the Argument models. With using the case example base, they can refer to the remarks from similar situations to use in hosting. Therefore, we need to collect more simulated mediation cases and statistically analyze them and with the results of them, we need to improve the accuracy of the similar situation search.

References

- [Rule, 2002] Colin Rule, Online Dispute Resolution For Business, Jossey-Bass, 2002.
- [Katsh, 2000] Ethan Katsh and Janet Rifkin, Online Dispute Resolution : Resolving Conflicts in Cyberspace, Jossey-Bass, 2000.
- [Ashley, 1991] Ashley, K.D, Reasoning with cases and hypotheticals in HYPO, Int.J.ManMachine Studies 34, pp.753-796, 1991.
- [Aleven, 1997] Vincent Aleven, Teaching Case-Based Argumentation Through a Model and Examples, University of Pittsburgh, 1997.

Appropriate Microplanning Choices for Low-Skilled Readers

Sandra Williams and Ehud Reiter

Department of Computing Science,
University of Aberdeen, Aberdeen AB24 3UE, U.K.
{swilliam,ereiter}@csd.abdn.ac.uk

Abstract

We have developed a set of microplanning choice rules which are intended to enable Natural Language Generation (NLG) systems to generate appropriate texts for readers with below-average literacy, focusing in particular on choices related to how discourse structure is expressed (cue phrases, ordering, sentence structure). Evaluation experiments suggest that our rules do enhance the readability of texts for low-skilled readers, although there is still room for improvement.

1 Introduction

Natural Language Generation (NLG) systems are computer programs that generate written texts in English or other human languages [Reiter and Dale, 2000]. Most existing NLG systems assume that generated texts will be read by proficient readers with good literacy levels. However, many people in the UK and elsewhere are not proficient readers. The goal of our research is to generate texts which low-skill readers will find (relatively) easy to read.

Generating appropriate texts for poor readers is a multifaceted problem, which involves choices at all NLG levels (content, microplanning, realisation). The focus of our research is on microplanning choices, in particular on choices related to the expression of discourse structure.

This work was done in the context of the GIRL and SkillSum¹ projects. These projects worked in the application area of generating feedback reports on assessments of adult basic skills (e.g. literacy). That is, users took a test assessing their basic skills, and GIRL/SkillSum generated for them reports that summarized their performance on the test.

While many previous researchers have looked at tailoring generated texts according to the user's domain expertise (e.g. [Paris, 1988]), less has been done on tailoring texts according to the reader's literacy. Perhaps the best known previous work in this area is PSET [Devlin *et al.*, 1999], which examined choices in texts intended for aphasic readers. Unfortunately most of PSET's rules were not experimentally validated. Scott and de Souza [1990] suggested

some psycholinguistically-motivated rules for expressing discourse relations, but did not evaluate them at all.

2 Microplanning choices investigated

The document (content) planners of our systems produce as output a tree, where core messages are related by discourse relations such as *explanation* or *concession*. Discourse relations are essentially RST relations, and messages are represented using a deep-syntactic representation. An example of an extract from a typical content plan, with messages shown as text glosses instead of deep syntactic structures, is shown in Figure 1.

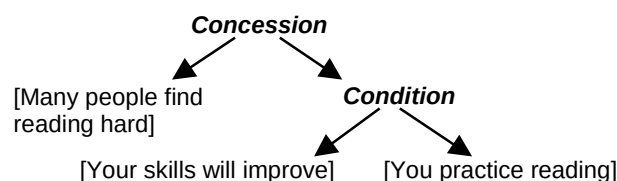


Figure 1 – Extract from a typical content plan

Our work focuses on how discourse relations such as *Concession* in Figure 1 are expressed, in particular:

- *cue phrases*: which cue phrases (if any) should be used to express a discourse relation?
- *ordering*: which order should the constituents related by a discourse relation be expressed in?
- *punctuation (sentence structure)*: should constituents be expressed in separate sentences (paragraphs?). If not, should punctuation be used to separate them?

We developed a set of rules for these choices which we hypothesised were appropriate for low-skill readers; this is our Enhanced Readability (ER) model. We also developed a control model for making these choices, based on the most common choices in the RST-DTC [Carlson *et al.*, 2002].

We created a microplanner that generated texts according to the rules in the ER and control models. We used a constraint-based approach that in general terms is similar to Power [2000]; further details are given in Williams [2004].

¹ Funded by PACCIT-LINK Grant ESRC RES-328-25-0026.

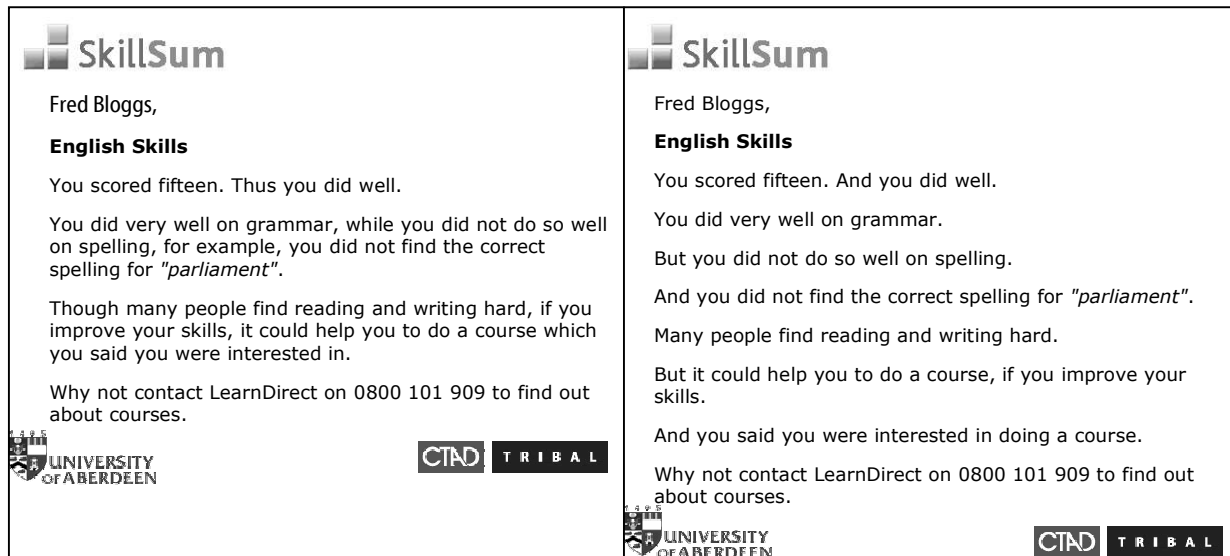


Figure 2: Examples of output from research system SkillSum. Left-hand control text automatically generated using corpus rules (and NOT intended for low-skilled readers), right-hand text generated with ER model.

3 Evaluation

GIRL evaluation experiment: 38 subjects, of varied literacy levels, took the GIRL assessment and were shown GIRL's feedback reports. Subjects were initially shown texts generated with either the ER or control version (randomised), and asked to read the report aloud; we measured reading time and reading errors (reading aloud was preferred over silent reading because pilot tests showed that many poor readers would skim texts when asked to silently read). Subjects were then shown both versions (ER and control) of the report, and asked to state a preference.

The results showed that poor readers on average seemed to read ER texts faster, make fewer reading errors on ER texts, and also preferred ER texts; however none of these results were statistically significant. There was little difference between the two versions for good readers

SkillSum evaluation experiment: 60 subjects were selected by skills experts to be people with moderate but not severe literacy problems, we also removed outliers; hence this group was more homogenous than of the first experiment. After completing the assessment and reading their own report, each subject was asked to read a report generated for someone else (in order to de-personalise the experiment); half read ER and half read control versions. In fact the reports read were those shown in Figure 2. As in the GIRL experiment, we measured reading aloud rate and reading errors (but not preference).

This time our results showed a significant effect on reading rate, subjects read the ER version 9% faster than the control version ($p=0.04$). There was also a weakly significant ($p=0.058$) improvement in reading errors.

4 Conclusion

We have only scratched the surface of the topic of generating appropriate texts for low-skilled readers; much more can

and should be done. In particular we would like to include lexical choice in our models, and also develop different models for people with different skill profiles. Nevertheless, we think our results to date are encouraging, and suggest that good choice rules can make a difference.

References

- [Carlson *et al*, 2002] L. Carlson, D. Marcu, and M. E. Okunowski. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In van Kuppevelt and Smith (eds.), *Current Directions in Discourse and Dialogue*. Kluwer.
- [Devlin *et al*, 1999] S. Devlin, J. Tait, Y. Canning, J. Carroll, G. Minnen and D. Pearce. The Application of Assistive Technology in Facilitating the Comprehension of Newspaper Text by Aphasic People. In Buhler and Knops (eds.) *Assistive Technology on the Threshold of the New Millennium*. IOS Press.
- [Paris, 1988] C. Paris. Tailoring Object Descriptions to a User's Level of Expertise. *Computational Linguistics* 14(3):64-78.
- [Power, 2000] R. Power: Planning texts by constraint satisfaction. *Proc of COLING 2000*, 642-648
- [Reiter and Dale, 2000]. E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Cambridge University Press.
- [Scott and de Souza, 1990] D. Scott and C. de Souza. Getting the Message Across in RST-Based Text Generation. In R Dale, C. Mellish and M. Zock (eds.), *Current Research in Natural Language Generation*. Academic Press.
- [Williams, 2004] S. Williams. Natural language generation (NLG) of discourse relations for different reading levels. PhD Thesis, University of Aberdeen.

POSTER PAPERS

PLANNING

Mixed-Initiative Activity Planning for Mars Rovers

John Bresina, Ari Jónsson*, Paul Morris, Kanna Rajan

NASA Ames Research Center

Mailstop 269-2

Moffett Field, CA 94035

{bresina,jonsson,pmorris,kanna}@email.arc.nasa.gov

Abstract

One of the ground tools used to operate the Mars Exploration Rovers is a mixed-initiative planning system called MAPGEN. The role of the system is to assist operators building daily plans for each of the rovers, maximizing science return, while maintaining rover safety and abiding by science and engineering constraints.

In this paper, we describe the MAPGEN system, focusing on the mixed-initiative planning aspect. We note important challenges, both in terms of human interaction and in terms of automated reasoning requirements. We then describe the approaches taken in MAPGEN, focusing on the novel methods developed by our team.

1 Introduction

Many complex systems are operated with operations plans that are generated offline at different intervals. Spacecraft, in particular planetary rovers, provide some of the most extreme examples of such operations. In the case of the Mars Exploration Rovers, a new plan was generated for each rover, every Martian day (which is slightly longer than an Earth day). Each plan had to satisfy complex safety rules, while achieving as much science as possible.

A mixed-initiative activity planning system called MAPGEN was one of the key tools used to generate the daily plans for the Mars rovers. The complete set of constraints and operations preferences, which varied significantly over time and between days, was impossible to formally specify. This made a fully automatic plan generation approach inapplicable to the problem at hand. However, the limited time available for plan generation and the complexity of the task at hand also made it infeasible to generate highly effective plans manually. Consequently, a mixed-initiative planning approach was adopted, where humans could control the construction of a plan, while automated planning and reasoning capabilities were used to assist the human user in making decisions, actively enforce constraints, and handle mundane task management issues.

While mixed-initiative planning [Burstein and McDermott, 1996] is a fairly established field, the MAPGEN application provided new challenges, both in terms of interactions with the user and in terms of how to apply automated reasoning. In this paper, we outline these challenges and describe the solutions developed.

2 Challenges and requirements

The primary challenge in the deployment of MAPGEN for the MER mission was to overcome skepticism about the use of automated planning. In particular, there were concerns that users would not understand what the automation was doing and that the automation would limit human input. To address this, the tool had to allow the user a broad range of operations, sufficient to arrive at any desired legal plan. At the same time, the automated reasoning behavior had to “feel natural” to the user.

Other specific challenges arose from user requirements. These included adjustable control over planning and scheduling decisions, as well as easy placement of activities in time. The key motivation was to allow users to choose ways to adapt plans to complex limitations such as onboard energy. These decisions required human-level judgment.

3 Mixed-initiative planning in MAPGEN

Like most mixed-initiative planning systems, MAPGEN combines a user interface with a planning services framework. In MAPGEN, the user interface is an established spacecraft operations tool called APGEN [Maldague *et.al.*, 1998], and the planning and automated reasoning services are provided by a constraint-based planning framework called EUROPA [Frank and Jónsson, 2003].

Through the interface, users can edit the plans, with oversight and active constraint enforcement supplied by the automated reasoning component. The core notion is that users can modify a plan by adding, editing, moving, and deleting high-level activities. The automated reasoning component handles management of low-level and support activities, such as ensuring resources like CPU are turned on when required. The automated reasoning also enforces constraints stemming from flight rules and specific daily constraints that define the coordination of science activities.

* Research Institute for Advanced Computer Science (RIACS)

3.1 Interactive plan modification

One of the core issues in mixed-initiative planning is the introduction of external decision-making and plan editing into a carefully designed automated search engine. The intrusion of user choices complicates commonly used approaches such as backtracking search and propagation-based forward checking of consistency. The EUROPA planning framework used in MAPGEN supports non-chronological backtracking, but it cannot propagate information in plans that have constraint violations. To support arbitrary changes by users, MAPGEN included a plan modification strategy that would adjust plans to eliminate inconsistencies.

3.2 Interactive temporal reasoning

The most common way for users to modify plans is to change the placement of activities in time. The EUROPA planning framework tracks a family of temporal instantiations, represented as a simple temporal network. This made it possible for users to quickly modify the temporal placement of activities while staying within a solution family. In MAPGEN, this idea was extended further to allow users to quickly move to a nearby family that differed only in the ordering of a small set of activities, using the previous solution as a heuristic to guide the search.

3.3 Interactive planning and scheduling

As noted above, MAPGEN users wanted a range of automated planning services. The system offers a fully automated “plan everything” operation, a selective “plan this and everything related to it” operation, and a fine-grained “plan this and try to put it here” operation. Users can also un-plan activities and store them in a “hopper,” which holds requested activities that are not yet in the plan.

3.4 Minimizing perturbation

The key to making the automated services feel natural and unobtrusive is for them to respect the existing plan as much as possible. This is accomplished by combining an effective form of temporal placement preference with a heuristic bias. For changes in the temporal placement of activities, the system exploits the underlying temporal flexibility of EUROPA plans. As each plan represents a family, the system chooses an instance to display that is as close as possible to what the user had prior to the changes being made. For more significant changes, such as those involving automated planning, the heuristic bias guides the search to options that restrict changes in temporal placement to those needed to satisfy constraints; thus the plan changes only in accordance with achieving the user’s desires.

3.5 Responsiveness issues

Mixed-initiative planning systems must respond and return control quickly to the user. For simple temporal placement, this is not a difficult issue, since the core operation is propagation in a simple temporal network, which can be done in low-order polynomial time.

For an automated planning operation, which involves a cascading decision process, MAPGEN relaxes completeness in favor of responsiveness. This has to be done carefully to maximize chances of finding near-optimal solutions within limited time. We developed a backtracking algorithm that noted the difficulty of planning activities, and when the effort to plan an activity exceeded an allowance determined by its priority, the activity was rejected from the plan.

4 Open issues and future work

A number of issues that arose in the development and use of MAPGEN could not be resolved in time for the Mars Rover application; among those are the following.

Explanations for user

When activities could not be moved or planned, due to a combination of prior decisions and applicable constraints, users had a hard time understanding why. This made it difficult for them to resolve the problems. To address this, an explanation facility is needed that can provide understandable reasons to the user and respond to questions.

Improved reasoning and planning

While the planning approach used in MAPGEN was quite effective, it left certain aspects of decision-making to the user—in particular, the user was tasked with handling vague preferences and very complex resource availability limitations. Future work should move towards incorporating complex preferences and automatically planning against complex resource limits.

5 Conclusions

The use of MAPGEN as a critical tool in the ground operations for the Mars Exploration Rovers is a major milestone in the application of automated planning in space exploration and other complex domains. Mission operations staff successfully used the tool in an intense, time-pressured environment, leading to an estimated 10 and 40 percent increase in science return, compared to operating without AI assistance. The deployment of MAPGEN has also identified key challenges for the mixed-initiative planning community and for the automated planning community as a whole.

References

- [Burstein and McDermott, 1996] Burstein, M., and McDermott, D., Issues in the development of human-computer mixed-initiative planning. In *Cognitive Technology*, B. Gorayska, and J. L. Mey, editors, pages 285-303. Elsevier, 1996.
- [Frank and Jónsson, 2003] Frank, J., and Jonsson, A., Constraint-Based Interval and Attribute Planning, *Journal of Constraints Special Issue on Constraints and Planning*, 2003.
- [Maldague *et.al.*, 1998] Maldague, P., Ko, A., Page, D., and Starbird, T., APGEN: A multi-mission semi-automated planning tool. First International NASA Workshop on Planning and Scheduling, Oxnard, CA, 1998.

Planning with graded fluents and actions

Marta Cialdea, Carla Limongelli, Andrea Orlandini, Valentina Poggioni

Dipartimento Informatica e Automazione

Università degli Studi Roma Tre

Via della Vasca Navale, 79 - 00146 Roma

{cialdea,limongel,orlandin,poggioni}@dia.uniroma3.it

Abstract

This work can be seen as a first approach to a new planning model that takes into account the possibility to express actions and fluents with non-boolean values. According to this model, a planning problem is defined using both graded (multi-valued) and classical (boolean) fluents. Moreover, actions that can have different application degrees can be defined. In this work a PDDL extension allowing to describe such new problems is proposed and a planning algorithm for such problems is presented.

1 Introduction

In the last years, extensions of the classical planning model have been investigated, such as temporal models, conditional and contingent models, probabilistic models and other mixed models (for example [Hoffmann, 2002; Petrick and Bacchus, 2002; Bonet and Geffner, 2003; Chien and al., 2000]). But, to the best of our knowledge, a feature of the classical model has never been modified: the use of boolean expressions to describe fluents and actions in the domains. This is often too restrictive in order to represent realistic domains because the world is not black and white (i.e. true or false) but it has a lot of colors (i.e. intermediate truth-values, or “degrees of truth”). A different approach to planning that takes into account a numerical “state” associated to a fluent is proposed in the probabilistic planning model, but, in this case, “numbers” represent our knowledge or uncertainty about the state or the success of an action, while the real world is always two-valued.

This work can be seen as a first approach to a new planning model that takes into account the possibility to express both actions and fluents with non-boolean values. According to this model, a planning problem is defined using both graded (multi-valued) and classical (boolean) fluents; moreover actions having different application degrees can be defined (i.e. actions having adjustable intensity and that can affect fluents proportionally to how much they are applied). In such a way, also the efficiency of actions can be easily represented.

The work defines an extension of the planning language PDDL that allows us to define planning domains and planning problems having graded fluents and graded actions, and a solving algorithm for such problems, where first a candidate plan with partially instantiated actions is constructed, then the

plan applicability and correctness is verified by means of a translation into a MIP (Mixed Integer Programming) problem and finally, if a solution of the MIP problem exists, a complete instantiation is made and the solution plan is found.

2 The Planning Language

The planning language proposed in this paper is based on standard PDDL. It provides two kinds of actions and fluents, representing both classical *boolean* fluents and actions, and *graded* fluents and actions. Both actions and fluents have an additional argument denoting their “degree of truth”: in a fluent it means “how much” the predicate is true (0 means that it is false, 1 that it is true) and in an action it means “how much” the action is applied (0 means that it is not applied at all, 1 means that it is applied with the maximum efficiency).

The type of such terms is declared as a new type *degree* in the PDDL domain definition. If the fluent is *boolean* then its truth-value is a natural number in $\{0, 1\}$ (as in the classical model), otherwise, if the fluent is *graded*, its truth-value is a real number in $[0, 1]$. The difference between the two kinds of fluents is declared in the domain description. There are different declaration sections for boolean and graded predicates.

The additional parameter in an action denotes the degree of application of the action. Again, the value of this parameter in *boolean actions* belongs to $\{0, 1\}$, while it ranges in $[0, 1]$ in *graded actions*. Actions may also have other additional parameters referring to fluent degrees. A specific section in action declaration allows one to associate fluents to degree parameters. Such variables maybe used both in preconditions and effects: action preconditions may contain inequality constraints over fluent degrees and effects can be described by means of expressions that define the new “degree of truth” of a fluent as a linear combination of previous fluent values and the application degree of the action.

The choice to include the degrees among the action parameters does not increase the complexity of the operators because, as explained in Section 3, such parameters are not instantiated during the solution search phase.

The definition of a graded planning problem may contain an *objective function*, that is a linear function of the action application degrees that must be minimized or maximized when looking for action degree values satisfying a given partially instantiated plan. For example, it can be used in or-

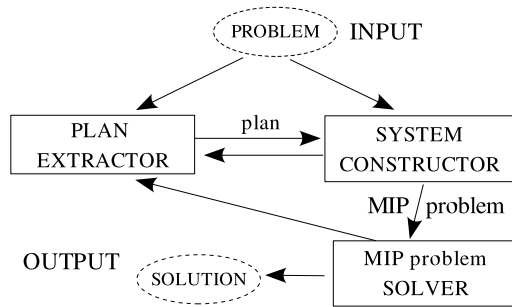


Figure 1: The system architecture

der to minimize the cost of the extracted candidate plan giving different operator costs. It is defined in a new section : `objective_function(...)` in the problem description.

3 System Architecture

The system takes in input a graded planning problem $(I, G, f, \mathcal{O})$ and if a solution exists it returns a graded solution plan P , otherwise, if it terminates it returns a *NoSolution* message. The system architecture is presented in Fig.1: it is composed by three modules, the *plan extractor*, the *system constructor*, the *MIP solver*.

The *plan extractor* synthesizes a “candidate plan” $P = (A_1(x_1), A_2(x_2), \dots, A_m(x_m))$ where actions are partially instantiated, i.e. where all application and fluent degrees are unbound but only the variables x_i standing for the degree of application of A_i are in evidence. It implements a simple backward algorithm with heuristic functions that solves relaxed problems. Then the *system constructor* computes the world evolution using these actions (the resulting states depend on the action application degrees) and reduces the verification and full instantiation of the plan to a MIP problem; it builds the MIP problem corresponding to a candidate plan and passes it to the *MIP solver*. During the system construction phase some conditions in the action preconditions are directly checked and if they are not satisfied the module fails and another candidate plan must be extracted. In this case the information about which action causes failure is used and another candidate plan is constructed replanning from this point. Finally the *MIP solver* computes a solution of the generated problem; if a solution exists it is a set of real and/or integer values $\vec{g} = (g_1, \dots, g_m)$ and the plan $P = (A_1(g_1), \dots, A_m(g_m))$ is the graded solution plan of the given problem, otherwise a new candidate plan is extracted and a new MIP problem is generated.

4 Conclusions and related works

In this work a language and a model of planning with graded fluents and actions are presented. A prototype of the system has been developed.

To the best of our knowledge this is the first system able to manage non boolean fluents and actions. Recent works have proposed languages [Fox and Long, 2003; Giunchiglia *et al.*, 2004; Lee and Lifschitz, 2003] and systems (for example [Koehler, 1998; Baiocchi *et al.*, 2003; Hoffmann, 2002; Haslum and Geffner, 2000]) that can handle numerical values. Degrees can be represented in PDDL 2.1, but only if

their values range on finite sets. Graded fluents could be represented in PDDL 2.1 by means of numerical fluents (functions) but respecting some more restriction w.r.t. what is done in this work. However, graded actions are not representable at all.

At the moment we are working in two main directions. First of all an algorithm for intelligent backtracking when the *MIP solver* fails and an improvement of the backtracking phase when the *system constructor* fails are under investigation. Moreover a set of graded domains and graded problems is under construction in order to carry out a wide set of experiments. The second research direction is theoretical: an extension of the algorithm to planning under uncertainty on the initial state is straightforward, introducing variables for fluent degrees in the world construction. Moreover, we are investigating the possibility of representing and treating vague (fuzzy) fluents, using intervals or sets to represent fluent degrees.

References

- [Baiocchi *et al.*, 2003] M. Baiocchi, A. Milani, and V. Poggioni. Planning with fuzzy resources. In *AI*IA 2003: Advances in Artificial Intelligence. In LNAI 2829*, 336-348, 2003.
- [Bonet and Geffner, 2003] B. Bonet and H. Geffner. Faster heuristic search algorithms for planning with uncertainty and full feedback. In *Proc. of IJCAI 2003*, 2003.
- [Chien and al., 2000] S. Chien and al. ASPEN: Automated planning and scheduling for space mission operation. In *Proc. of SpaceOps 2000, Colorado (USA)*, 2000.
- [Fox and Long, 2003] M. Fox and D. Long. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *JAIR*, 20:61–124, 2003.
- [Giunchiglia *et al.*, 2004] E. Giunchiglia, J. Lee, V. Lifschitz, McCain N., and H. Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153:49–104, 2004.
- [Haslum and Geffner, 2000] P. Haslum and H. Geffner. Heuristic planning with resources. In *Proc. of ECAI-00*, 2000.
- [Hoffmann, 2002] J. Hoffmann. Extending FF to numerical state variables. In *Proc. of ECAI 2002*, 2002.
- [Koehler, 1998] J. Koehler. Planning under resource constraints. In *Proc. of ECAI-98*, 1998.
- [Lee and Lifschitz, 2003] J. Lee and V. Lifschitz. Describing additive fluents in action language $\mathcal{J}^+$. In *Proc. of IJCAI 2003*, 2003.
- [Petrick and Bacchus, 2002] R.P.A. Petrick and F. Bacchus. A knowledge-based approach to planning with incomplete information and sensing. In *Proc. of AIPS 2002*, 2002.

Automated Adaptive Support for Task and Information Prioritizing

Tjerk de Greef[†] and Peter-Paul van Maanen^{†‡}

[†] Department of Information Processing, TNO Human Factors

PO BOX 32, 3769 ZG Soesterberg, the Netherlands

[‡] Department of Artificial Intelligence, Vrije Universiteit Amsterdam

De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

Email: degreef@tm.tno.nl, peter-paul.vanmaanen@tno.nl

Abstract

This paper discusses a means for automatically supporting humans in information and task prioritizing. A new generic method based on the Competitive Task Model is described. Its implementation is able to calculate priorities of competing information entities, which provides a way of allocating tasks to the process operator. Due to a combination of dynamic states and information entities, it is usable as an adaptive control mechanism for attention and task allocation.

1 Introduction

The objective of an intelligent information system is to support the operator with the decision making process by means of offering the right information at the right time and place. This appears not an easy job as experience and research in the field of adaptive decision support learns [e.g., Weiland et al., 1998]. Information with a high interruption level is intended to focus the attention of the operator on important information. A popular scenario in literature describes a process operator facing an emergency situation where the operator is overwhelmed by a huge amount of information, which increases the operator workload and is making it almost impossible to deal with the emergency [Sheridan, 1981]. The mentioned scenario typically occurs when time is limited. It is therefore crucial to use valuable time carefully and it seems obvious to support the operator in managing information flow after (or before) a problematic event.

A reduction of information seems to be a logical first approach, but unfortunately several studies show that the reduction of information not necessarily results in improvement of efficiency [e.g., Baker et al., 1985]. The removed information apparently is relevant for the operator. This may be because this information provides insight into the state of the process [Stanton, 2000]. In the present research a combination of research suggested by Freed [1998], Nugent et al. [2000] and Covery [1990] is further explored and used for a description of a model called the

Competitive Task Model (CTM). After this a first description of an algorithm based on CTM is shown.

2 The Competitive Task Model

In the CTM information entities (IEs) are prioritized by means of estimating their *relevance*, *urgency*, and *importance*, in the context of the *goal profile*, *user*, and *task knowledge* (Figure 1). Relevance describes which user tasks have a relation with an offered IE and urgency describes the temporal (response) aspect of the IE for a particular task. Relevance and urgency estimation is done by means of two interpreters that require knowledge of the user and its tasks. User and task knowledge is used to determine if the offered IE reduces uncertainty [Shannon et al., 1946] and to provide an indication of the relation between an IE and a specific task, respectively. The act of combining these interpretations per specific task yields a *delta score*, indicating the priority of the IE with respect to a certain user

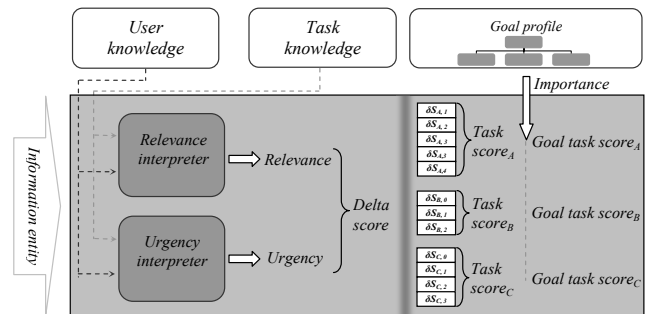


Figure 1 – The Competitive Task Model

task. For each user task the delta scores are combined for all offered IEs. This results in a *task score*, indicating the priority of a certain task given a set of IEs. Finally, the task scores are related to the specific goals using a two-layered goal profile (GP). The first layer describes the goals and every goal implies one or more tasks. These tasks populate the second layer. The GP yields an indication of the *importance* of a specific task with respect to the user's

higher goals. The combination of importance of a certain user task and its related task score results in a *goal task score*, indicating the eventual prioritization of user tasks.

As an example of applying the CTM, imagine a navy officer on a frigate. The task profile consists of three high level goals, i.e. to fight, to sail and to float. Suppose that an IE concerning a fire in the engine room reaches the officer. This IE is highly urgent and relevant for the sailing task, but to a lesser extent for the floating task. Hence the delta score of the sailing task is higher than that of the floating task. But according to the GP (domain knowledge) floating is more important than sailing if the floating capability is in danger, and therefore the CTM indicates that the floating task should receive attention instead of the sailing task.

3 The Basic Algorithm

According to the CTM there are three terms needed to prioritize IEs: *relevance* (R), *urgency* (U) and *importance* (I). The first two terms describe task τ and state σ dependent information for which $R_i(\tau, \sigma)$ and $U_i(\tau, \sigma)$ are numbers in the interval $[0,1]$, where i is an IE. These two terms are quantified via relevance and urgency interpreters (Figure 1) and are combined as a quantitative expression of the priority of the IE on the task and state using a function called the delta score (δS):

$$\delta S_i(\tau, \sigma) = R_i(\tau, \sigma) \oplus U_i(\tau, \sigma) \quad (1)$$

where $\oplus$ is an operator combining R and U . In order to calculate the priorities of the competing tasks, a task score (TS) is calculated by means of the summation of all δS 's per task, given a state:

$$TS(\tau, \sigma) = \sum_{i \in IEs} \delta S_i(\tau, \sigma) \quad (2)$$

The above task score is not taking into account the related higher level importance. Therefore TS and I are combined:

$$TS_{imp}(\tau, \sigma) = I(\tau, \sigma) \otimes TS(\tau, \sigma) \quad (3)$$

where $\otimes$ is a operator combining TS and I . The combining operators $\otimes$ and $\oplus$ can be for instance normalized linear functions with its form possibly dependent on the given task τ , state σ , and IE i .

4 Discussion

The CTM model described in this paper is able to calculate the priorities of competing tasks based on a set of IEs. The IEs are analysed in the context of the current activities of the operator, the importance of the IE regarding the main goals, and the urgency and relevance of the IE regarding the operator's tasks. Because of the combined input of IEs and dynamic states it can be used as an adaptive control mechanism for operator attention or as task allocation

strategies, that are crucial in (automating) adaptive decision support. This is different from traditional models of preference choice from game theory, where preferences are given *a priori*. Though, further research is needed to optimize the theoretical framework. For instance, Freed [1998] mentions that switching between hierarchies can be an efficient approach when information is applicable for a number of subtasks. The CTM does not elaborate on this. Furthermore, the automatic relevance and urgency interpreters, and the combining operators in the basic algorithm need further experimentation and specification.

Acknowledgments

This research is funded by the Dutch Ministry of Defense under programme numbers V055 and V206.

References

- [Baker et al., 1985] S Baker, David Gertman, Erik Hollnagel, C. Holstrom, E Marshall and F Øwre. *An experimental comparison of three computer-based alarm systems: results and conclusions*. Halden Reactor Project (HWR-142), Norway.
- [Coverly, 1990] Stephen R. Coverly. *The 7 habits of highly effective people*. New York: Simon & Schuster.
- [Freed, 1998] Michael Freed. Managing Multiple Tasks in Complex, Dynamic Environments. In *Proceedings of National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference*, pages 921-927, Madison, Wisconsin.
- [Nugent et al., 2000] William A. Nugent and Richard W. Obermayer, *Human Computer Interaction Design Guidelines for an Alert Warning and Attention Allocation System. I. Literature Review*. San Diego, CA: Space and Naval Warfare Systems Center, Code D44209, 17 February 2000.
- [Shannon et al, 1946] C Shannon & W Weaver, *The mathematical theory of communications*. Urbana: University of Illinois Press.
- [Sheridan, 1981] Thomas B. Sheridan. Understanding human error and aiding human diagnostic behavior in nuclear power plants. In J. Rasmussen and William B. Rouse (Eds), *Human detection and diagnosis of system failures*. New York: Plenum Press.
- [Stanton et al., 2000] N Stanton, David Harrison, Karin Taylor- Burge, L. Porter, Sorting the Wheat from the Chaff, A study on the detection of alarms. *Cognition, Technology and Work*, 2(3):134-141, August 2000.
- [Weiland et al., 1998] Monica Weiland, Gwendolyn E. Campbell, Wayne Zachary and Janis A. Cannon-Bowers. Applications of cognitive models in a combat information center. Paper presented at the *1998 Command and Control Research and Technology Symposium*. Monterey, CA.

Planning for Weakly-Coupled Partially Observable Stochastic Games

AnYuan Guo Victor Lesser

University of Massachusetts Amherst
Department of Computer Science
140 Governor's Drive, Amherst, MA 01003
{anyuan, lesser}@cs.umass.edu

1 Introduction

Partially observable stochastic games (POSGs) provide a powerful framework for modeling multi-agent interactions. While elegant and expressive, the framework has been shown to be computationally intractable [Bernstein *et al.*, 2002]. An exact dynamic programming algorithm for POSGs has been developed recently, but due to high computational demands, it has only been demonstrated to work on extremely small problems. Several approximate approaches have been developed [Emery-Montemerlo *et al.*, 2004; Nair *et al.*, 2003], but they lack strong theoretical guarantees. In light of these theoretical and practical limitations, there is a need to identify special classes of POSGs that can be solved tractably.

One dimension along which computational gain can be leveraged is by exploiting the independence present in the problem dynamics. In this paper, we examine a class of POSGs where the agents only interact loosely by restricting one another's available actions. Specifically, rather than having fixed action sets, each agent's action set is a function of the global state. The agents are independent from one another otherwise, i.e. they have separate transition and reward functions that do not interact. This class of problems arises frequently in practice. Many real world domains are inhabited by self-interested agents that act to achieve their individual goals. They may not affect each other in any way, except for occasionally putting restrictions on what each other can do.

The best-known solution concepts in game theory are that of computing Nash equilibrium and performing strategy elimination. Our work addresses both of these solution concepts. First, we show that finding a Nash equilibrium where each agent achieves reward at least k is NP-hard. Another contribution of this work is that we present an exact algorithm for performing iterated elimination of dominated strategies. It is able to solve much larger problems than exact algorithms for the general class by exploiting problem structure.

2 POSGs with State-Dependent Action Sets

In this section, we will present a formal definition of our model. An n -agent POSG with state-dependent action sets can be defined as $\langle \{S_i\}, \{s_i^0\}, \{A_i\}, \{B_i\}, \{P_i\}, \{R_i\} \rangle$, where,

- S_i is the state space of agent i . $S = S_1 \times S_2 \times \dots \times S_n$ denotes the joint state set.

- $s_i^0 \in \Delta(S_i)$ is the initial state distribution of agent i .
- A_i is the action set of agent i . $A = A_1 \times A_2 \times \dots \times A_n$ denotes the joint action set.
- $B_i : S \rightarrow 2^{A_i}$ is the available action function that maps a joint state to the set of available actions for agent i . $B_i(s) \subset A_i$ for all $s \in S_i$.
- $P_i : S_i \times A_i \times S_i \rightarrow [0, 1]$ is the transition function of agent i . The joint transition function is completely factored, where $P((s'_1 \dots s'_n) | (s_1 \dots s_n), (a_1 \dots a_n)) = \prod_{i=1}^n P_i(s'_i | s_i, a_i)$.
- $R_i : S_i \times A_i \rightarrow \mathbb{R}$ is the reward function for agent i . This states that the rewards of the agents depend only on the local state and the action taken by agent i .

In our model, each agent has a complete view of its own local state. A local policy for each agent is a mapping from the local state and the available action set to an action in that set. A joint policy is a tuple of local policies, one for each agent.

Definition 1 A local policy π_i , for agent i of an n -agent POSG with state-dependent action sets, is a mapping from pairs of local states and local action sets $\langle s_i, A_i(s) \rangle$ to actions in the current local action set $A_i(s)$, where $s = \langle s_1, \dots, s_i, \dots, s_n \rangle$.

Here is an example of a POSG with state-dependent action sets. Two autonomous rovers are exploring an unknown terrain and collecting rock samples. There is a river with a narrow bridge on it that only allows one rover to pass at a time. To simplify the illustration, we will assume the rovers have two states $\{\text{on land}, \text{on bridge}\}$, and three actions each $\{\text{move}, \text{get on bridge}, \text{pick up rock}\}$. The rovers receive reward for the number of rocks picked up. The state-dependent action set is used to constrain the rover's actions such that only one is allowed on the bridge at a time. For example, the available actions for rover 1 are specified as follows, $B_1(\langle s_1 = \text{on land}, s_2 = \text{on land} \rangle) = \{\text{move}, \text{get on bridge}, \text{pick up rock}\}$, $B_1(\langle s_1 = \text{on land}, s_2 = \text{on bridge} \rangle) = \{\text{move}, \text{pick up rock}\}$, and analogously for rover 2.

3 Complexity

We state our decision problem, denoted as POSG-NE, as follows: given a POSG with state-dependent action sets,

$G = \langle \{S_i\}, \{s_i^0\}, \{A_i\}, \{B_i\}, \{P_i\}, \{R_i\} \rangle$, does there exist a Nash equilibrium where all agents have expected utility at least k ?

Theorem 1 *POSG-NE is NP-hard even in problems involving 2 agents and a horizon of 2.*

4 Iterated Action Elimination Algorithm

We present an algorithm that performs iterated elimination of dominated strategies. The algorithm is able to work directly with the compact representation of a POSG without first converting it to the double exponentially large normal form representation. In this algorithm, first, we first fix the action sets of each agent at each state to the optimistic and pessimistic action sets. The idea behind this is that although an agent cannot predict exactly what actions will be available at each state, it can find out what actions would be available in the best and worst case scenario. In the best case, none of the actions that can be restricted by the state of the other agents are, and in the worst case, all of the actions that can be restricted are in fact unavailable.

For each agent i :

1. For each state s_k , compute the optimistic and pessimistic action sets. Here $s = \langle s^1, s^2, \dots, s^n \rangle$ and $s^i = s_k^i$.

$$A_{opt}(s_k^i) = \bigcup_s B_i(s)$$

$$A_{pes}(s_k^i) = \bigcap_s B_i(s)$$

2. Compute the value functions of the 2 MDPs that correspond to alternately fixing the available action sets to the A_{opt} and A_{pes} .

$$U(s) = \max_{a \in A_{opt}(s)} \left[R(s, a) + \sum_{s'} P(s'|s, a) U(s') \right]$$

$$L(s) = \max_{a \in A_{pes}(s)} \left[R(s, a) + \sum_{s'} P(s'|s, a) L(s') \right]$$

3. For each action in each state, derive upper and lower bounds on the action value from the value function bounds $U(s)$ and $L(s)$.

$$Q_U(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) U(s')$$

$$Q_L(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) L(s')$$

4. At each state, eliminate all actions a' whose action value is dominated by another action a , $Q_L(s, a) \geq Q_U(s, a')$.
5. Repeat steps 1 to 4 until no more actions can be eliminated at any state.

Table 1: The iterated action elimination algorithm.

# states	before pruning	after pruning
6	972,000	13
8	34,992,000	168
9	2.1×10^8	69
12	4.5×10^{10}	774
16	5.9×10^{13}	2,198

Table 2: Average size of the policy space before and after action elimination for problems with 3 constrained states.

Once we fix the available action sets of each agent at all the states, the agents no longer depend on each other in any way. Our model decomposes to a set of MDPs. For each agent, we solve two MDPs, one using the optimistic action sets and the other using the pessimistic action sets. The solutions will provide us with an upper and a lower bound on the actual value function of each agent. With these upper and lower bounds on the expected values of each state, we can now derive bounds on the expected action values. At each state, dominated actions are pruned. We iterate the action elimination procedure until no more actions can be pruned. Since the number of actions at each state is finite, the procedure will eventually converge.

Theorem 2 *The iterated action elimination algorithm corresponds to the iterated elimination of very weakly dominated strategies.*

5 Experiments

We tested the algorithm on a simplified 2-agent autonomous rover exploration scenario. The size of each agent's local state space varies from 6 to 16. We introduced up to three constrained states in each agent's state space. 100 trials were run for each size of state space and number of constraints. Table 2 shows detailed results for problems with three constrained states. For problems with one and two constrained states, the final policy space ranges from 3 to 106 and 8 to 252 respectively. In all three cases, the iterated action elimination algorithm is able to reduce the size of the policy space by several orders of magnitude.

References

- [Bernstein *et al.*, 2002] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27:4, November 2002.
- [Emery-Montemerlo *et al.*, 2004] R. Emery-Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the Third Joint Conference on Autonomous Agents and Multiagent Systems*, 2004.
- [Nair *et al.*, 2003] R. Nair, M. Tambe, M. Yokoo, D. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.

Multi-Agent Assumption-Based Planning

Damien Pellier, Humbert Fiorino

Laboratoire Leibniz, MAGMA

46, avenue Félix Viallet, 38000 Grenoble, France

Damien.Pellier@imag.fr, Humbert.Fiorino@imag.fr

Abstract

The purpose of this poster is to introduce a dialectical theory for plan synthesis based on a multi-agent approach. This approach is a promising way to devise systems able to take into account partial knowledge and heterogeneous skills of agents. We propose to consider the planning problem as a defeasible reasoning where agents exchange proposals and counter-proposals and are able to conjecture *i.e.*, formulate plan steps based on hypothetical states of the world.

1 Introduction

The problem of plan synthesis achieved by autonomous agents in order to solve complex and collaborative tasks is still an open challenge. Increasingly new application areas can benefit from this research domain. From our point of view, multi-agent planning can be likened to the process used in automatic theorem proving. In a sense, a plan can be considered to be a particular proof based on specific rules, called actions. In this poster, we draw our inspiration from the proof theory described by [Lakatos, 1976]. The plan synthesis problem is viewed as a dialectical and collaborative goal directed reasoning about actions. Each agent can refine, refute or repair the ongoing team plan. If the repairing of a previously refuted plan succeeds, it becomes more robust but it can still be refuted later. If the reparation of the refuted plan fails, agents leave this part of the reasoning and explore another possibility: finally “bad” sub-plans are ruled out because no agent is able to push the investigation process further. As in an argumentation with opponents and proponents, the current plan is considered as an acceptable solution when the proposal/counter-proposal cycles end and no more objection remains.

The originality of this approach relies on the agent capabilities to elaborate plans under partial knowledge and/or to produce plans that partially contradict its knowledge. In other words, in order to reach a goal, such an agent is able to provide a plan *which could be executed if certain conditions were met*. Unlike “classical” planners, the planning process does not fail if some conditions are not asserted in the knowledge base, but rather proposes an Assumption-Based Plan or *conjecture*. Obviously, this conjecture must be *reasonable*: the

goal cannot be considered “achieved” and the assumptions must be as few as possible because they become new goals for the other agents. For instance, suppose that a door is locked: if the agent seeks to get into the room behind the door and the key is not in the lock, the planning procedure fails even though the agent is able to fulfill 100% of its objectives behind the door. Another possibility is to suppose for the moment that the key is available and then plan how to open the door etc. whereas finding the key might become a new goal to be delegated. To that end, we designed a planner that relaxes some restrictions regarding the applicability of planning operators.

2 Assumption-Based Planning Model

In order to produce conjectures, we propose an assumption-based planning model based on a domain independent planning mechanism, HTN. In HTN planning model, the objective is not to achieve a set of goals but to perform some sets of *tasks*. The agent’s input includes a set of operators and also a set of *methods*, each of which is a prescription on how to decompose some tasks into some sets of subtasks. The agent proceeds by decomposing *non-primitive tasks* recursively into smaller and smaller subtasks, until *primitive tasks*, that can be performed directly by planning operators, are reached.

Assumption-Based Planning is defined as $\langle \mathcal{G}, \mathcal{E}, \mathcal{A} \rangle$: $\mathcal{G}$ is a goal description (*i.e.*, a set of world states), $\mathcal{E}$ is a partial description of the world (*i.e.*, the agent’s beliefs) and $\mathcal{A}$ is a description of the actions that an agent can execute. $\mathcal{E}$ and $\mathcal{G}$ are described in propositional logic.

A *primitive operator* α is a tuple $\langle name_\alpha, Pre_\alpha, Del_\alpha, Add_\alpha \rangle$ where $name_\alpha$ is the name of the primitive action, Pre_α is the preconditions set needed to execute α , Del_α and Add_α define respectively the set of effects to delete and to add to the agent’s beliefs.

A *compound operator* α is described by a method $\langle n_\alpha, Pre_\alpha, Act_\alpha \rangle$ where n_α is the name of α , Pre_α is the preconditions set needed to apply α , Act_α defines a list of actions to execute (*i.e.*, the method body).

A *planning problem* is defined by a tuple $\langle \mathcal{E}, \mathcal{O}, \mathcal{G} \rangle$: $\mathcal{E}$ defines the agent’s beliefs; $\mathcal{O} = \{ \langle Pre_\alpha, Del_\alpha, Add_\alpha \rangle \mid \alpha \in \mathcal{A} \}$ defines the description of the operators that an agent can execute; $\mathcal{G}$ defines the goal of an agent.

A *conjecture* χ is defined as an ordered list of couples $\chi = (\langle \mathcal{H}_{\alpha_1}, \alpha_1 \rangle, \dots, \langle \mathcal{H}_{\alpha_n}, \alpha_n \rangle)$ where α_i is an action in $\mathcal{A}$; $\mathcal{H}_{\alpha_i}$ describes the assumptions that must hold before executing α_i . If $\mathcal{H}_{\alpha_i}$ is an empty set, no assumption is needed to apply α_i .

Given a planning problem $\langle \mathcal{E}, \mathcal{O}, \mathcal{G} \rangle$, a conjecture χ is equivalent to an ordered list of $n + 1$ world states $\chi = \mathcal{E}_0, \mathcal{E}_1, \dots, \mathcal{E}_n$ with $\mathcal{E}_0 = \mathcal{E} \cup \mathcal{H}_{\alpha_1}$ and $\mathcal{E}_i = (\{\mathcal{E}_{i-1} \cup \mathcal{H}_{\alpha_{i-1}}\} - Del_{\alpha_i}) \cup Add_{\alpha_i}$ ($1 \leq i \leq n$). The set of solution conjectures $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G})$ of a planning problem can be recursively defined:

- If $\mathcal{G}$ is an empty set, the empty conjecture is returned.
- Otherwise, let α be the first task or goal of $\mathcal{G}$, and $\mathcal{R}$ be the remaining goals: i) If α is a primitive action and there is a conjecture χ_1 to reach α then $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G}) = \{\text{append}(\chi_1, \chi_2) \mid \chi_2 \in \mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{R})\}$; ii) If α is a primitive action and there is no conjecture χ to reach α then $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G}) = \emptyset$; and finally, iii) If α is a composed action then $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G}) = \mathcal{C}(\mathcal{E}, \mathcal{O}, \text{append}(\text{Act}_\alpha, \mathcal{R}))$ where Act_α defines the actions list to be executed in order to realize α .

To reach its goal, an agent must check all assumptions made by the conjecture. It can rely on its teammates competences to make those assumptions become true. In other words, assumptions made by one agent become additional goals to be recursively satisfied. The assumption computation distinguishes two kinds of assumptions: i) *Hypothesis* represents literals that do not belong to the current agent's beliefs. This means that expressions missing from the beliefs are not considered as false but rather as unknown; ii) *Fact negation* represents literals that are negations of facts in the current agent's beliefs: this fact is withdrawn and replaced by its negation.

3 Dialectical Controller

In a multi-agent context, the plan synthesis relies on dialectical exchanges between agents as expected in a debate. This is convenient for dialogue where participants have no predetermined role (*i.e.*, proponent, opponent). Agents interact collaboratively in the dialogue so as to construct a plan without assumption fulfilling the assigned goals. Agents have a dialogue based on speech acts. At the informational layer, speech acts can contain conjectures or propositions needed to elaborate the global shared plan. The conjectures and the refutations suggested by the agents during the dialogue are stored in a structure called *proof board*. Each agent has its own proof board initialized at the dialogue beginning. The proof board defines the public part of the agents' interactions and records the agents' reasoning. Each time an agent uses a speech act to refine, refute or repair a conjecture, the proof board is updated. Through agents' dialogue, only one specific part of the shared plan is considered. We call this particular conjecture, the *current conjecture*. This conjecture defines the dialogue context. In order to structure the dialectical plan synthesis, we propose to define a set of dialogue conventions described according to the states of the proof board. These conventions are organized in two layers:

- an *informational layer* that defines the rules to exchange refinements, refutations and repairings about the current

conjecture. Each new conjecture suggested by an agent produces new goals to achieve for the other agents.

- a *contextualization layer* in which agents decide to start, when a set of goals is assigned to the system, or stop interacting when they believe that a solution was found or is not reachable. Finally, agents can decide to change the dialogue context by modifying the current conjecture if it has been refuted or none of the agents can refine its assumptions.

The proof board of each agent can be viewed as a proof tree. Each node defines a conjecture suggested by an agent. The edges of the tree express refinement relations. Each child node defines a sub-conjecture proving an assumption of the conjecture contained in the father node. In order to test if a solution plan *i.e.*, a plan without assumption is contained in the proof board, agents try to extract recursively a sub-tree of the proof board from the root node. For each assumption in the root node, a sub-tree which nodes are not refuted and do not contain assumptions must be extracted. This test is recursively triggered on the alternative refinements proposed for each assumption. The performatives list used are shown in Tab. 3. We consider two performatives levels. The performatives used to refine, refute or repair conjectures that have an informational content and the performatives used to modify the dialogue context. Each time a performative is received or sent, the proof board is modified

Levels	Performatives
Informational	refine, refute, repair
Contextualization	prop.enter, prop.leave, prop.change, ack.enter, ack.leave, ack.change, rej.enter, rej.leave, rej.change

Table 1: Performatives list ordered by level.

4 Conclusion

The advantage of the dialectical plan synthesis is to merge in the collaborative plan generation, the composition and the coordination steps. It also includes the notion of uncertainty in the agents' reasoning and allows agents to make conjectures and to compose their heterogeneous competences. Moreover, we apply conjecture/refutation to structure the multi-agent reasoning as a collaborative investigation process. However, former works on synchronization, coordination and conflict resolution are integrated through the notions of refutation/repairing. From our point of view, this approach is suitable for applications in which agents share a common goal and in which the splitting of the planning and the coordination steps (when agents have independent goals, they locally generate plans and then solve their conflicts) becomes difficult due to the agents strong interdependence.

References

- [Lakatos, 1976] I. Lakatos. *Proofs and Refutations: The Logic of Mathematical Discovery*. Cambridge University Press, Cambridge, England, 1976.

Open-World Planning for Story Generation

Mark O. Riedl

Institute for Creative Technologies
University of Southern California
13274 Fiji Way, Marina Del Rey, CA 90292
riedl@ict.usc.edu

R. Michael Young

Department of Computer Science
North Carolina State University
Raleigh, NC 27695
young@csc.ncsu.edu

Abstract

The ability to generate narrative is of importance to computer systems that wish to use story effectively for entertainment, training, or education. One way to generate narrative is to use planning. However, story planners are limited by the fact that they can only operate on the story world provided, which impacts the ability of the planner to find a solution story plan and the quality and structure of the story plan if one is found. We present a planning algorithm for story generation that can non-deterministically make decisions about the description of the initial story world state in a least-commitment fashion.

1 Introduction

The standard approach to incorporating storytelling into a computer system is to script a story at design time. That is, a system's designers determine ahead of time what the story should be and hard-code the story into the system. An alternative approach is to generate stories either dynamically or on a per-session basis using planning.

Young [1999] suggests that planning has many benefits as a model of narrative.

- Plans are comprised of partially ordered steps. If steps are character actions, then a plan is a good model of a story *fabula* – a chronological enumeration of events that occur in the story world.
- Planning algorithms construct plans based on causal dependencies between steps. This ensures that all events are part of causal chains, resulting in coherent story structure.

Story planners such as Universe [Lebowitz, 1985] and IPOCL [Riedl and Young, 2004] require a predefined world description as input. The task of creating the world in which a story is set is already accomplished by the “human author” – the user – and a story planner only has to find a plot that can be performed in the story world. It is possible for the human author to describe a story world initial state that cannot be transformed through character actions into the desired outcome. In this case, the responsibility for a planner's failure lies with the human author. Another

possibility is that the story planner can find a solution plan, but the solution plan is strangely structured because the planner must work around limitations inherent in the initial state description.

It is our belief that a story planner that has discretion over the description of the initial story world state will:

- Be able to avoid failure conditions that are attributed to the human author's definition of the initial world state.
- Find better quality solutions because the planner does not need to work around limitations inherent in the human author's description of the initial world state.

2 Initial State Revision

The use of the closed world assumption (CWA) presupposes complete and correct knowledge about the world. Since the universe of discourse that describes any story world is potentially infinite, the practical implementation of the CWA is to enumerate everything about the world that is known to be true so that everything that is not explicitly declared to be true is presumed false. Since story worlds are often fictional, the CWA places a burden on the human author to completely specify the story world or risk limiting the stories that can be constructed by the planner. However, there may be many aspects of the story world that the human author may not have a preference over.

The Initial State Revision (ISR) algorithm partitions knowledge about the initial world state into *three* sets: known true ($\mathcal{T}$), false ($\mathcal{F}$), and undetermined ($\mathcal{U}$). $\mathcal{T}$ is the set of atomic ground sentences that are known to be true. $\mathcal{F}$ is the set of atomic ground sentences that are known to be false or are assumed to be false because their true value is not specified by the human author. $\mathcal{U}$ is the set of atomic (but not necessarily ground) sentences whose truth values are undetermined but not assumed to be false. $\mathcal{U}$ is the set of knowledge about the state of the world that the human author has explicitly designated as having no preference over. Consequently the possible values of $\mathcal{U}$ define *possible worlds* in which the story can be set. Initial state revision reduces the set of possible worlds by committing to values of sentences in the undetermined set as necessary to produce the best possible story plot.

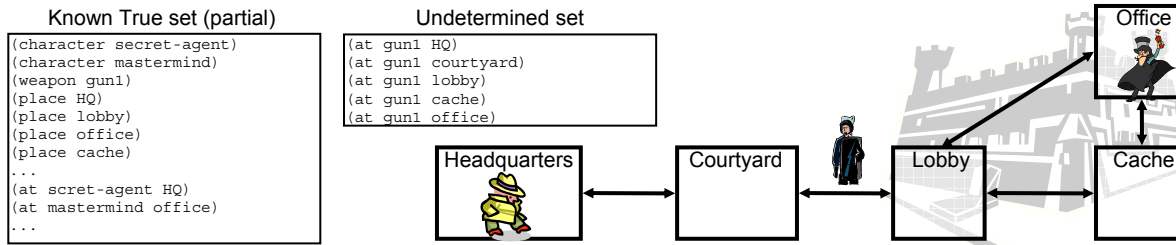


Figure 1: Diagram of the secret agent world.

2.1 Motivating Example

Figure 1 shows an example domain in which a secret agent must assassinate a terrorist mastermind. However, the terrorist mastermind lives in a fortress which is heavily guarded such that no one can enter the fortress while carrying a weapon. The description of the story world disallows any story that involves the secret agent entering the fortress and assassinating the terrorist, for the secret agent will be unable to carry a weapon past the guards. Furthermore, suppose the story planner is constrained such that the terrorist mastermind never leaves the fortress. In this case the story planner can not find a complete story plan for this story world unless the planner is able to revise the initial state of the world. Figure 1 shows the known true and undetermined portions of the initial world state. Properties of the gun have been defined such that its location is undetermined. Through initial state revision, the story planner can declare the gun to have always been inside the fortress, thus making it possible to find a complete plan.

2.2 The Algorithm

ISR planning is an extension of the conventional partial order causal link (POCL) planning algorithm [Penberthy and Weld, 1992]. Plan steps have preconditions and effects. In every complete plan, preconditions of a step must be satisfied by a causal link from the effect of a preceding step or by the initial world state. An open (unsatisfied) condition on a plan step can unify with sentences in any set in the initial step, including the undetermined set. When an open condition p on some plan step unifies with a sentence q in the undetermined set, a causal link is established as in typical POCL planning and, additionally, sentence q becomes determined. If p is not negated, q is determined to be true and moved into set $\mathcal{T}$, otherwise q is determined to be false and moved into set $\mathcal{F}$. By using q to satisfy an open condition, the planner has in effect committed to a set of possible worlds in which the value of q is determined. Note that it is not necessary for the undetermined set to be empty for planning to be complete. That is, the story plan can be in a set of possible worlds if the distinguishing details are not integral to the plot.

It is often the case that propositions that describe the state of the world are mutually exclusive, i.e., if one is true then the other cannot be true. For example, an object cannot be in more than one place at a time. The undetermined set $\mathcal{U}$ can contain atomic sentences that should never be

simultaneously true in the initial state of the world. ISR utilizes *mutual exclusion sets* (also called *mutex sets*). A mutex set is a set of sentences such that no two sentences can both be true in the initial state. If a sentence in a mutex set is determined to be true, then all other sentences in the mutex set are determined to be false. In the example, the sentences in $\mathcal{U}$ are all mutually exclusive to ensure that the gun cannot be in more than one place.

3 Conclusions

Planners are constrained to operate in the world as provided by their users. For an agent operating in a limited domain, it is possible for the world to be completely specified. For infinitely large domains such as a fictional story world, it is impossible to encode every fact about the world. The closed world assumption places a burden on the human author to describe a world that supports story generation. It is possible, however, that despite the human author's intentions, the story world description does not adequately suit storytelling, resulting in planner failure or in solution story plans that are oddly structured to overcome limitations of the world description. The ISR planning algorithm assumes creative control over aspects of the story world description. With control over the initial state of the story world, a story planner can find better story plans because it is not limited by assumptions made by the human author.

Acknowledgements

The work presented here has been supported by the National Science Foundation CAREER Award #0092586.

References

- [Lebowitz, 1985] Michael Lebowitz. Story-telling as planning and learning. *Poetics*, 14, 1985.
- [Penberthy and Weld, 1992] J. Scott Penberthy and Daniel S. Weld. UCPOP: A sound, complete, partial order planner for ADL. In *Proc. of the 3rd Int. Conf. on Knowledge Representation and Reasoning*, 1992.
- [Riedl and Young, 2004] Mark O. Riedl and R. Michael Young. An intent-driven planner for multi-agent story generation. In *Proceedings of AAMAS*, 2004.
- [Young, 1999] R. Michael Young. Notes on the use of planning structures in the creation of interactive plot. In *Proc. of the 1999 AAAI Fall Symposium on Narrative Intelligence*, 1999.

Disjunctive Temporal Planning with Uncertainty

K. Brent Venable
Dept. of Mathematics
University of Padova
Padova, Italy
kvenable@math.unipd.it

Neil Yorke-Smith
Artificial Intelligence Center
SRI International
Menlo Park, CA 94025, USA
nysmith@ai.sri.com

Abstract

Driven by planning problems with both disjunctive constraints and contingency, we define the Disjunctive Temporal Problem with Uncertainty (DTPU), an extension of the DTP that includes contingent events. Generalizing existing work on Simple Temporal Problems with Uncertainty, we divide the time-points into controllable and uncontrollable classes, and propose varying notions of controllability to replace the notion of consistency.

1 Introduction

For modelling and solving real-world planning and scheduling problems, quantitative temporal constraint networks in the form of the Simple Temporal Problem (STP) [Dechter *et al.*, 1991] are widely known. The STP has been independently extended in two orthogonal directions.

First, since the STP formalism is unable to express concepts such as “event A occurs before or after event B”, the Disjunctive Temporal Problem (DTP) [Stergiou and Koubarakis, 2000] admits more than one disjunct in a constraint. Thus DTP constraints are disjuncts of the form: $Y_1 - X_1 \in [a_1, b_1] \vee \dots \vee Y_n - X_n \in [a_n, b_n]$. Note a variable may appear in more than one disjunct. A simple way to solve a DTP is to consider the *component* STPs obtained by selecting one disjunct from each constraint. The computational complexity of solving each component STP (deciding consistency and deriving the minimal network) is polynomial, while the worst-case complexity of solving a DTP is NP-hard [Stergiou and Koubarakis, 2000]; but in practice, efficient solving techniques have been developed (e.g. [Tsamardinos and Pollack, 2003]), and tractability results are known for some classes of DTPs [Satish Kumar, 2005].

Second, since the STP formalism assumes all time-points are under the complete control of the execution agent, the *Simple Temporal Problem with Uncertainty* (STPU) [Vidal and Fargier, 1999; Morris *et al.*, 2001] divides the time-points into two classes, *executable* and *contingent*. Contingent time-points are controlled by exogenous factors, often referred to as ‘Nature’, and consistency of an STP is replaced by varying notions of *controllability*.

Driven by planning problems that feature both disjunctive constraints and contingent events, in this poster we propose

the *Disjunctive Temporal Problem with Uncertainty* (DTPU). Consider, for instance, planning the schedule of an elderly patient [Peintner and Pollack, 2004]. On the one hand, her schedule includes events, such as exercising and a relative’s visit, that are not totally ordered by the problem requirements; on the other hand, while the patient has some idea when the relative will arrive, the precise time is not under her control.

2 Extending the DTP with Uncertainty

Following the STPU formalism [Vidal and Fargier, 1999], in a DTPU we divide the time-point variables into two classes: the controllable *decision variables* V_d and the uncontrollable *parameters* V_u . In an STPU, this induces a partition of the temporal relation constraints C into two types. A constraint $Y - X \in [a, b]$, where $Y \in V_u$, is a *contingent* link. Intuitively, contingent links model tasks with uncertain durations. If a constraint is not contingent, it is an *executable* link; thus all other constraints (i.e. where $Y \in V_d$) are executable links.

In a DTPU, each constraint is a disjunction of links:

Definition 1 A DTPU is a tuple $\langle V_d, V_u, R_d, R_u, C \rangle$, where V_d, V_u are as above, R_d is the set of all executable links, and R_u is the set of all contingent links, and C is a finite set of disjunctive temporal constraints.

Thus in a DTPU, three types of constraints arise according to the contingent status of the included disjuncts. First, only R_d disjuncts; analogously with STPUs, the constraint is said to be *executable*. Second, only R_u disjuncts; analogously with STPUs, the constraint is said to be *contingent*. Third, both R_d and R_u disjuncts. Here, a situation which has no analogue in an STPU, the constraint can be satisfied by the satisfaction of either some executable link (a possibility under the agent’s control), or of some contingent link (which is not); we say that it is *semi-executable*.

A constraint is satisfied if at least one of its disjuncts is satisfied. A solution to a DTPU is a complete assignment to all variables $V = V_d \cup V_u$ that satisfies all the constraints in C . A solution s can be partitioned into $s_d \cup s_u$, where s_d and s_u are the set of temporal values assigned to elements of V_d and V_u respectively. s_d is called a *decision*, and s_u is a *realisation* (or possible world, or scenario). We say a *component* STPU of a DTPU is the problem obtained by selecting one disjunct from each constraint.

Example 1 Consider further the example of an elderly patient’s schedule [Peintner and Pollack, 2004]. The time-points in a DTP model of the schedule include E_s and E_e , and V_s and V_e , which describe the start and end of exercising and a visit respectively. One of the constraints is:

$$V_s - E_e \in [5, \text{inf}] \vee E_s - V_e \in [0, \text{inf}] \quad (1)$$

stating that “Exercise must finish at least 5 minutes before the visitor arrives or start anytime after the visitor leaves.”

While the visitor has a scheduled arrival time of 4:30 p.m., his precise arrival is not under the control of the patient; thus V_s is a parameter. This means that the first disjunct of (1) is a contingent link. The second disjunct is an executable link, and so constraint (1) is semi-executable.

2.1 Controllability

For space reasons, we present informal definitions extending the notions of controllability to DTPUs. The important new possibility is of a semi-executable DTPU constraint.

Strong Controllability. A DTPU is *strongly controllable* (SC) if there exists a decision that, combined with any realisation, satisfies the constraints. In other words, there is a way to assign values to the decision variables such that, given any values for the parameters, at least one disjunct on each constraint is satisfied. Note this means that a DTPU is SC if some component STPU is SC. The converse is false, however, since in a SC decision, $s_d \cup s_u$ may satisfy different disjuncts in different realisations.

Dynamic Controllability. A DTPU is *dynamically controllable* (DC) if there exists an incremental, online decision that, given knowledge of the values of all parameters that have occurred until now, can be dynamically extended to a decision in the future, whatever the values of the parameters that have not yet occurred. Note this means that a DTPU is DC if some component STPU is DC, provided the decision for the component STPU can be consistently extended to a decision to all decision variables in the complete DTPU. (A component STPU may involve a strict subset of V_d .) Again, the converse is false in general.

Weak Controllability. A DTPU is *weakly controllable* (WC) if for any realisation there exists a decision that, combined with that realisation, satisfies the constraints. That is, every realisation can be consistently extended to a decision. Once more, a DTPU is WC if some component STPU is WC, but the converse is false.

Trivial Controllability. The situation where every constraint is semi-executable brings the possibility that the agent could satisfy the DTPU without regard to the contingent links, by satisfying at least one executable link in each constraint. We say a component STPU of a DTPU is a *component STP* if it contains only executable links. A DTPU is *trivially controllable* (TC) if one or more of its component STPUs is a consistent component STP. Consistency of a component STP is required to prove that: every TC DTPU is SC.

Since $SC \implies DC \implies WC$ (inherited from the STPU case), by testing whether a DTPU is trivially controllable, we have a sufficient but not necessary condition for all other types of controllability. Testing for this condition requires, in the worst case, checking the consistency of every component STP. While there may be exponentially-many such STPs to check, the consistency of each can be determined relatively quickly, compared to other inference tasks for DTPUs.

The other constraints of Example 1, not shown here, are all single-disjunct executable constraints. Thus, since a component STP arising with the second disjunct of (1) is consistent, the whole problem is trivially controllable. While so for this particularly simple example of a DTPU, more interesting DTPUs are not TC in general.

3 Future Work

As a step towards modelling real-world planning problems with disjunctive temporal constraints and contingent events, we have proposed the Disjunctive Temporal Problem with Uncertainty. While the DTPU extends previous work on the STPU, much future work is needed. On the theory side, our main task is to formally state and prove properties of DTPUs; on the practical side, our task is to develop algorithms to check the various controllability notions, and prove their complexity. Beyond this, we are interested to combine the DTPU framework with user preferences [Peintner and Pollack, 2004], in a similar manner to [Rossi *et al.*, 2004].

Acknowledgments. We thank M. Pollack and T. Uribe for useful discussions, and the reviewers for their comments. This material is based upon work supported by the Defense Advanced Research Projects Agency, through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

References

- [Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49(1–3):61–95, 1991.
- [Morris *et al.*, 2001] P. Morris, N. Muscettola, and T. Vidal. Dynamic control of plans with temporal uncertainty. In *Proc. of IJCAI’01*, pages 494–502, Seattle, WA, 2001.
- [Peintner and Pollack, 2004] B. Peintner and M. E. Pollack. Low-cost addition of preferences to DTPs and TCSPs. In *Proc. of AAAI-2004*, pages 723–728, San Jose, CA, 2004.
- [Rossi *et al.*, 2004] F. Rossi, K. B. Venable, and N. Yorke-Smith. Controllability of soft temporal constraint problems. In *Proc. of CP’04*, LNCS 3258, pages 588–603, Toronto, Canada, 2004.
- [Satish Kumar, 2005] T. K. Satish Kumar. On the tractability of restricted disjunctive temporal problems. In *Proc. of ICAPS’05*, pages 110–119, Monterey, CA, 2005.
- [Stergiou and Koubarakis, 2000] K. Stergiou and M. Koubarakis. Backtracking algorithms for disjunctions of temporal constraints. *Artificial Intelligence*, 120(1):81–117, 2000.
- [Tsamardinos and Pollack, 2003] I. Tsamardinos and M. E. Pollack. Efficient solution techniques for disjunctive temporal reasoning problems. *Artificial Intelligence*, 151(1–2):43–89, 2003.
- [Vidal and Fargier, 1999] T. Vidal and H. Fargier. Handling contingency in temporal constraint networks: From consistency to controllabilities. *JETAI*, 11(1):23–45, 1999.

POSTER PAPERS

ROBOTICS AND PERCEPTION

Talking Robots: a Fully Autonomous Implementation of the Talking Heads

Jean-Christophe Baillie & Matthieu Nottale

Laboratory of Electrical and Computer Engineering, ENSTA

32 Bd Victor 75015 Paris France

jean-christophe.baillie@ensta.fr matthieu.nottale@ensta.fr

Abstract

The “Talking Robots” experiment, inspired by the “Talking Heads” experiment from Sony, explores possibilities on how to ground symbols into perception using language, with two autonomous Aibo robots in an unconstrained environment. We present here the first results of this experiment and outline in the conclusion a planned extension to social behaviors grounding.

1 Introduction

The “Talking Robots” experiment, inspired by the “Talking Heads” experiment from Sony, has been started for six months in our laboratory. The purpose of this experiment is double: first, to show that the original “Talking Heads” experiment can be reproduced in a more demanding context: autonomous robots (Aibos), unconstrained vision and speech, and autonomous synchronization. Second, to serve as a ground to build a more complex experiment. Making a full use of the unconstrained context, we will in future work investigate how the robots could autonomously develop their own language games.

We will present briefly in this poster the structure of the existing “Talking Robots” experiment and the underlying scientific questions that we address. We will also comment on our first results.

2 Grounding symbols into perception

Grounding symbolic representations into perception is a key and difficult issue for artificial intelligence [Harnad, 1990; Steels and Baillie, 2003]. Including social interactions and, more specifically, language acquisition and development in this context has proven to be a fruitful orientation. One of the recent and successful attempts in this direction is the “Talking Heads” experiment [Steels and Kaplan, 2002]. This experiment involves two cameras interacting in a simplified visual environment made of colored shapes on a white board. The agents inside the cameras are developing a *shared grounded lexicon* of words related to visual meanings like “red”, “large”, “above”, etc.

The “Talking Heads” experiment has proven that it was possible to design a computer-based mechanism for language

acquisition that shares many properties in common with what can be observed in human language acquisition. It can be seen as an attempt in the direction of a computational model of language acquisition and symbol grounding, outlining possible key structural elements that might be essential for such a development.

However, the “Talking Heads” had two main limitations, which were good and necessary simplifications to start with, but that we will consider carefully here:

- The environment was limited and simplified (only colored shapes on a white background, fixed cameras).
- The interaction protocol (language game) was predetermined and hardcoded.

With the recent development of relatively cheap and powerful robotic platforms (see Sony, Honda or Fujitsu, among others) research on symbol grounding can move from simulation or simple environments to complex natural environments and embodied systems. Following this trend, we have started the “Talking Robots” experiment which reproduces the initial Talking Heads experiment with autonomous robots (Aibo ERS7) evolving in an unconstrained visual environment instead of simple cameras.

The second point, regarding hardcoded interactions, is not handled in the Talking Robots, but constitutes our future research goal. The visual routines, robust perception mechanisms and the general experience we drew from the Talking Robots will be a key component and a basis for an extended experiment involving the dynamical creation of language games between two robots.

3 Results

3.1 Technical issues

The autonomous and unconstrained implementation of the language game called “Guessing Game” involves the following technical and practical issues:

- Automatically determine the leader (the “speaker”) and the follower (the “hearer”) in the game: we use beep signals with specific tones to perform synchronization.
- Have the two robots standing one next to the other, starting from a random position: we use movement detectors and a waving head motion to locate an Aibo head in the

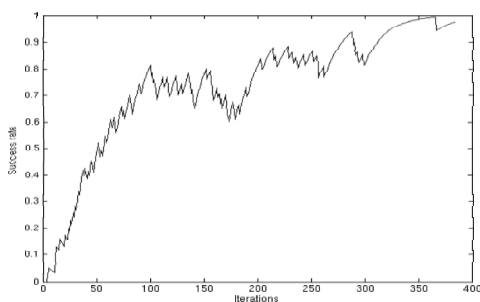
image and pattern recognition based on a FMI-SPOMF transform [sheng Chen *et al.*, 1994] to identify the orientation of this head. Together with a scale measure, this gives an accurate enough estimation of the position and orientation of the other robot.

- Use a stable segmentation algorithm which will give comparable results from one robot to the other, if they look in approximately the same direction: the CSC algorithm [Rehrmann and Prieze, 1998] has proven to be the most stable after a series of tests (the measure of stability is the entropy of a normalized correlation matrix between two segmentations of two slightly different images, which should ideally be zero).
- Provide an accurate pointing device for the robots to designate the objects they are talking about: we use a blinking laser pointer fixed on the robot's head and a simple red spot detector which gives excellent results.

Putting all these elements together, we successfully obtained converging results for the "Discrimination Game" (comparable to the original experiment) and we have run several successful "Guessing Games".

3.2 Discrimination Game

The discrimination game [Steels and Kaplan, 2002] is played by one robot only and is designed to dynamically create a set of visual categories grounded in the surrounding visual environment of the robot: for example, with objects of different sizes in the image, a category for "big" and "small" will be developed but no category corresponding to "colors" will arise in a black and white environment. Our games successfully converge with 250 iterations to a set of approximately 80 to 90 categories in the unconstrained environment of the lab, and a success rate above 80%:



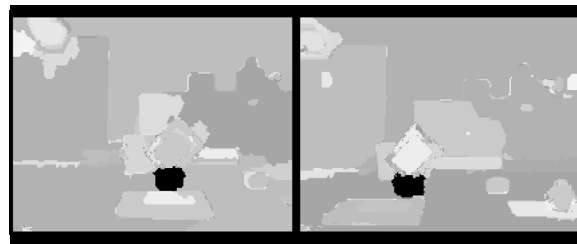
Categories are mainly developed for position and area. If we turn the vision of the robot to black and white, the discrimination tree corresponding to "saturation" shrinks in a few iterations, as expected.

3.3 Guessing Game

The "Guessing Game" has been run on a limited set of experiments. Although we have still no statistical results yet, we have observed in detail several games to compare them to the original Talking Heads.

The following figure shows the result of a guessing game played on a topic (in black) whose meaning has been misunderstood by the hearer but correctly identified after the

speaker has pointed it (both robots succeed in sharing the same topic):



This kind of failure is currently happening most of the time, and we have to improve the constancy of segmentation and context to increase the probability of having the speaker and the hearer both categorizing a topic in the same way. Apart from this, the original mechanisms of the Talking Heads have been successfully reimplemented and run. Further results will be presented in the poster.

4 Conclusion

The Talking Robot experiment has started to successfully reproduce in a complex environment the results obtained in the simplified environment of the Talking Heads. Building on our experience on this embedded version, we plan to start a new set of experiments involving not only a predetermined *language game* interaction protocol, but where the robots will dynamically create a protocol and use it to play a language game which will not be known in advance. This ambitious research project will require to state precisely the core structural elements of the Talking Robots and propose an architecture to generalize them and have a mechanism to let them evolve. On the practical side, we will make use of the robust vision and localization algorithms we developed for the Talking Robots.

References

- [Harnad, 1990] Harnad. The symbol grounding problem. *Physica D* 42, pages 335–346, 1990.
- [Rehrmann and Prieze, 1998] Rehrmann and Prieze. Fast and robust segmentation of natural color scenes. In *ACCV (1)*, pages 598–606, 1998.
- [sheng Chen *et al.*, 1994] Qin sheng Chen, Michel Defrise, and F. Deconinck. Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(12), 1994.
- [Steels and Baillie, 2003] Luc Steels and Jean-Christophe Baillie. Shared grounding of event descriptions by autonomous robots. *Robotics and Autonomous Systems*, 43(2-3):163–173, 2003.
- [Steels and Kaplan, 2002] Steels and Kaplan. Bootstrapping grounded word semantics. In T. Briscoe, editor, *Linguistic evolution through language acquisition: formal and computational models*, chapter 3, pages 53–73. Cambridge University Press, Cambridge, 2002.

An On-Line Time Warping Algorithm for Tracking Musical Performances*

Simon Dixon

Austrian Research Institute for Artificial Intelligence
Freyung 6/6, Vienna 1010, Austria
simon@oefai.at

Abstract

Dynamic time warping is not suitable for on-line applications because it requires complete knowledge of both series before the alignment of the first elements can be computed. We present a novel on-line time warping algorithm which has linear time and space costs, and performs incremental alignment of two series as one is received in real time. This algorithm is applied to the alignment of audio signals in order to track musical performances.

1 On-Line Time Warping

Although efficiency and real-time concerns of dynamic time warping (DTW) have been addressed in the literature, we do not know of any work in which the real-time constraint involves a streamed sequence, so that the alignment must be calculated incrementally, in the forward direction, while one of the sequences is not known in entirety. In this work we present an on-line time warping (OLTW) algorithm which is able to perform incremental alignment of arbitrarily long sequences in real time.

DTW aligns time series $U = u_1, \dots, u_m$ and $V = v_1, \dots, v_n$ by finding a minimum cost path $W = W_1, \dots, W_l$, where each W_k is an ordered pair (i_k, j_k) , such that $(i, j) \in W$ means that the points u_i and v_j are aligned. The alignment is assessed with respect to a local cost function $d(i, j)$, usually represented as an $m \times n$ matrix, which assigns a match cost for aligning each pair (u_i, v_j) . The path cost is the sum of the local match costs along the path. Several constraints are placed on W , namely that the path is bounded by the ends of both sequences, and it is monotonic and continuous. The minimum cost path can be calculated in quadratic time by dynamic programming, using the recursion:

$$D(i, j) = d(i, j) + \min \left\{ \begin{array}{l} D(i, j-1) \\ D(i-1, j) \\ D(i-1, j-1) \end{array} \right\}$$

where $D(i, j)$ is the cost of the minimum cost path from $(1, 1)$ to (i, j) , and $D(1, 1) = d(1, 1)$. The path itself is obtained by tracing the recursion backwards from $D(m, n)$.

In the on-line case, the length of the incoming sequence is unknown, so one of the boundary conditions for the search

must be estimated along with the optimal path. Also, in order to run in real time with arbitrarily long series, the complete algorithm must be linear in the length of the series, so that the incremental step is bounded by a constant. If U is the partially unknown sequence, then we seek at each time t the best alignment of the sequence $u_1, \dots, u_t$ to some initial subsequence of V . This is performed with the OLTW algorithm (Figure 1), which we now explain.

The variables t and j are pointers to the current positions in series U and V respectively. The main loop of the algorithm calculates a partial row or column of the path cost matrix. The calculation of the path cost uses the standard DTW recursion formula, restricted to use only the matrix entries which have already been calculated. The path cost is normalised by the path length, so that paths of varying lengths can be compared in the function *GetInc*. The number of cells calculated is given by the search width parameter, c ; e.g. for a new row, the row number is incremented, and the cells in the last c columns up to and including the current column are calculated.

The function *GetInc* selects whether to calculate a row, column, or both. If less than c elements of each series have been processed, new rows and columns are alternately calculated. If one sequence has been incremented successively *MaxRunCount* times, the other sequence is incremented. Otherwise the minimum path cost for each cell in the current row and column is found. If this occurs in the current position (t, j) , then both the row and column counts are incremented; if it occurs elsewhere in row j , then the row count is incremented, otherwise the column count t is incremented.

For each incoming data point u_t , the minimum cost path calculated at time t is the same as that calculated by DTW, assuming the same path constraints, but the number of calculations performed by OLTW is bounded by a constant. A further advantage of OLTW is that the centre of the search band is adaptively adjusted to follow the best match.

2 Tracking of Musical Performances

In music performance, high level information such as structure and emotion is communicated by the performer through parameters such as tempo, dynamics, articulation and vibrato. These parameters vary within a musical piece, between musical pieces and between performers. We use OLTW to extract this information directly from audio signals by aligning different performances of the same piece of music, enabling live tracking and visualisation of expressive parameters during a performance. This could be used to complement the listening

*This work was supported by the Vienna Science and Technology Fund project C1010 *Interfaces to Music* and the EU-FP6-IST-507142 project *SIMAC*. ÖFAI acknowledges the financial support of the Austrian federal ministries BMBWK and BMVIT.

ALGORITHM On-Line Time Warping

```

t := 1; j := 1
previous := None
INPUT u(t)
EvaluatePathCost(t, j)
LOOP
  IF GetInc(t, j) != Column
    t := t + 1
    INPUT u(t)
    FOR k := j - c + 1 TO j
      IF k > 0
        EvaluatePathCost(t, k)
  IF GetInc(t, j) != Row
    j := j + 1
    FOR k := t - c + 1 TO t
      IF k > 0
        EvaluatePathCost(k, j)
  IF GetInc(t, j) == previous
    runCount := runCount + 1
  ELSE
    runCount := 1
  IF GetInc(t, j) != Both
    previous := GetInc(t, j)
END LOOP

FUNCTION GetInc(t, j)
IF (t < c)
  return Both
IF runCount > MaxRunCount
  IF previous == Row
    return Column
  ELSE
    return Row
(x, y) := argmin(pathCost(k, l)), where
  (k == t) or (l == j)
IF x < t
  return Row
ELSE IF y < j
  return Column
ELSE
  return Both

```

Figure 1: The on-line time warping (OLTW) algorithm.

experience of concert-goers, to provide feedback to teachers and students, or to enable interactive performance and automatic accompaniment systems.

The audio data is represented by the positive spectral difference between successive 20ms frames, that is, the increase in energy (if any) in each frequency bin. This emphasises the onsets of tones, the most important indicators of musical timing. The cost of matching two frames is given by the Euclidean distance between their spectral difference vectors.

Since the slope of the path represents the relative tempo, it is reasonable to constrain it to a range between $\frac{1}{3}$ and 3 ($MaxRunCount = 3$), to allow for moderate but not arbitrary differences in tempo. In off-line tests with a search width of 10s ($c = 500$ frames) the program aligned each minute of music in about 10 seconds (3GHz PC). It also runs comfortably in real time with these parameters.

Quantitative testing was performed off-line with recordings of 22 pianists playing 2 excerpts of solo piano music

Error $\leq$		Percentage of notes			
Frames	Seconds	On-line		Off-line	
		Etude	Ballade	Etude	Ballade
0	0.00	13.9%	13.1%	46.5%	36.6%
1	0.02	36.1%	34.5%	84.5%	77.1%
2	0.04	52.8%	50.7%	91.1%	88.9%
3	0.06	64.0%	62.2%	93.4%	92.5%
5	0.10	78.4%	76.7%	96.0%	95.1%
10	0.20	92.7%	91.4%	98.8%	97.4%
25	0.50	98.7%	98.5%	99.8%	99.1%
50	1.00	99.9%	99.7%	100.0%	99.8%

Table 1: Alignment results shown as cumulative percentages of notes with an error up to the given value (see text).

by Chopin (Etude in E Major, Op.10, no.3, bars 1–21; and Ballade Op.38, bars 1–45), played on a computer-monitored grand piano. This provided precise measurements of the times and velocities of all notes, so that we had both the audio recordings and discrete measurements of each note.

After aligning a pair of files, the error for each note or chord was calculated, using the Manhattan distance between the point representing the onsets of the corresponding notes in the two performances and the nearest point on the time warping path. In Table 1, we show the percentages of notes with errors less than or equal to 0,1,2,3,5,10,25 and 50 frames across the 231 ($= \frac{22 \times 21}{2}$) pairs of performances of each piece. The results using the off-line version of the algorithm are also included in the table. The average error was 84ms for the on-line algorithm and 30ms for the off-line version; the worst errors were 2.62s (on-line) and 3.64s (off-line). The off-line algorithm performs better because it has the advantage of knowing the future evolution of the signal when calculating the alignment at a given point.

3 Conclusion

We presented a new on-line time warping algorithm, which aligns a sequence arriving in real time with a stored sequence of arbitrary length. At each time frame, the calculated path is optimal with respect to the data computed up to that time, but this might not correspond to the optimal path calculated by an off-line algorithm with full knowledge of both sequences.

OLTW was used in implementing a musical performance alignment system, which was tested on several hundred pairs of performances with an average error of 84ms. The system can be used off-line for comparisons of performance interpretation, using unlabelled audio recordings, e.g. in a media player plug-in which, given a position in one audio file, automatically finds the corresponding position in other audio files of the same piece of music. An on-line application is automatic accompaniment: given an audio file with the soloist and accompaniment on separate tracks, alignment could be performed on the solo track while the accompaniment track is played back with corresponding accommodation of the dynamic and timing changes of the soloist. Currently planned extensions of this work include a score following system and a range of visualisation tools for use in concerts, teaching and private rehearsal. At the conference we will demonstrate the system tracking tempo and dynamics in a live performance and displaying the data with an animation designed by musicologists for off-line performance visualisation.

Path-Planning for Autonomous Training on Robot Manipulators in Space

Froduald Kabanza

Université de Sherbrooke
kabanza@usherbrooke.ca

Roger Nkambou

U. du Québec à Montréal
nkambou.roger@uqam.ca

Khaled Belghith

Université de Sherbrooke
khaled.belghith@usherbrooke.ca

Abstract

This paper describes the integration of robot path-planning and spatial task modeling into a software system that teaches the operation of a robot manipulator deployed on International Space Station (ISS). The system addresses the complexity of the manipulator, the limited direct view of the ISS exterior and the unpredictability of lighting conditions in the workspace. Robot path planning is used not for controlling the manipulator, but for automatically checking errors of a student learning to operate the manipulator and for automatically producing illustrations of good and bad motions in training.

1 Introduction

Designing software that teaches requires, in advanced cases, the implementation of “intelligence” capabilities. After all, the best human teachers are those mastering the subject they teach, having communication skills and understanding the student’s problem solving process in order to help him. With the aim of furthering intelligent software-based education systems, we have been developing a software simulator, called *RomanTutor*, that can be used to train astronauts to operate a robotic manipulation system (the Mobile Servicing System, MSS), on the International Space Station (ISS, Figure 1).

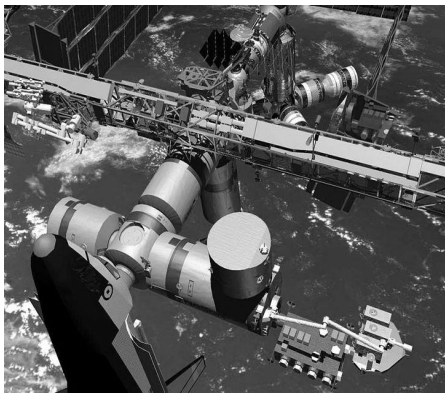


Figure 1. ISS with the SSRMS

The MSS consists of a Space Station Remote Manipulator System (SSRMS), a Mobile Base System (MBS), a Mobile Transporter (MT), and a Special Purpose Dexterous Manipulator (SPDM). The SSRMS is a 17-meter long articulated robot manipulator with seven rotational joints and two latching end-effectors which can grapple special fixtures, giving it the capability to “walk” from one grapppling fixture to next on the exterior of the ISS. The SPDM is a dexterous manipulator with two symmetrical six-joint arms and can be operated from the end of the SSRMS. The MT is a platform that serves to move SSRMS along the main truss of ISS.

The MSS is operated from a robotic workstation located inside one of the ISS modules and is equipped with three video monitors, each displaying a view from one of the 14 cameras mounted on the ISS exterior and the SSRMS. Crewmembers operating the MSS have no direct view of the ISS exterior other than the three monitors. In fact, choosing the right camera views to display is one of the tasks for operating the SSRMS.

RomanTutor is a system still under development; here we describe the integration of robot path-planning and spatial task modeling in an MSS simulator to provide useful feedback to a student operating the SSRMS. To illustrate, when a student is learning to move a payload, RomanTutor invokes the path-planner periodically to check whether there is a path from the current configuration to the target and provides feedback accordingly. The path-planner not only computes collision free paths but is also capable of taking into account the limited direct view of the ISS, the lighting conditions and other safety constraints about operating the SSRMS.

2 Architecture and Basic Functionalities

RomanTutor works with any robot manipulator provided a 3D model of the robot and its workspace are specified. The system includes the following components among others (Figure 2): a graphic user interface, a feedback generator, a path planner, a movie generator, and third-party libraries (PQP [Larsen *et al.*, 2000], Open Inventor from Silicon Graphics and MPK [Sanchez and Latombe, 2001]).

A snapshot of the *user interface* is shown on Figure 3. It emulates the Robot workstation using three screens (for the three monitors). The keyboard is used to operate the robot. In *command mode*, one controls the joints directly; in *automatic mode*, one moves the end-effector, small increments at a time, relying on inverse kinematics to calculate the joint rotations.

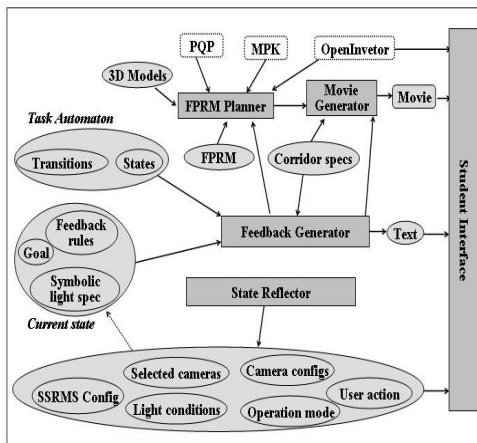


Figure 2. RomanTutor Architecture

The robot free workspace is segmented into zones with each zone having an associated degree of desirability, that is, a real number in the interval $[0, 1]$, depending on the task, visual cue positions, camera positions, and lighting conditions. The closer the *dd* is to 1, the more the zone is desired. *Safe corridors* are zones with *dd* near to 1, whereas *unsafe corridors* are those with *dd* in the neighborhood of 0. We extend the definition of *dd* to robot configurations and paths in a straightforward way.

Students carry out robot operation tasks that involve moving the manipulator (avoiding collision and singularities, using the appropriate speed, switching cameras as appropriate, and using the right operation mode at each stage), berthing, or mating. These tasks require the student to be able to identify a corridor in a free workspace for a safe operation of the robot and follow it. The student must do this based on the task, the location of cameras and visual cues, and the current lighting conditions. Therefore localization and navigation are important in robot operations.

The *feedback generator* periodically checks the current state to trigger feedback to the student, using rules that are preconditioned on the current state information and the current goal. These are “teaching” expert rules and can be as efficient as the available teaching expertise allows. The feedback generator also changes the lighting conditions based upon specification rules in the current state. Feedback rules, lighting rules and goals are structured into a *task automaton*.

A *task automaton* is a state transition system that abstracts the evolution of the simulated system

configuration under the student’s actions. In any given state of the task automaton the feedback generator is monitoring the student’s actions, focusing on one subgoal, and using a set of feedback rules to help the student. A transition to a new state (e.g., the payload has reached a predefined milestone) switches the subgoal and the feedback rules to those of the new state.

Feedback rules are production rules preconditioned on the state variables and the state goal; the consequents are normally multimedia content to be displayed to the student, but rules can also update user-defined variables (e.g., to keep a record of previous state variables or to update a performance score for the student).

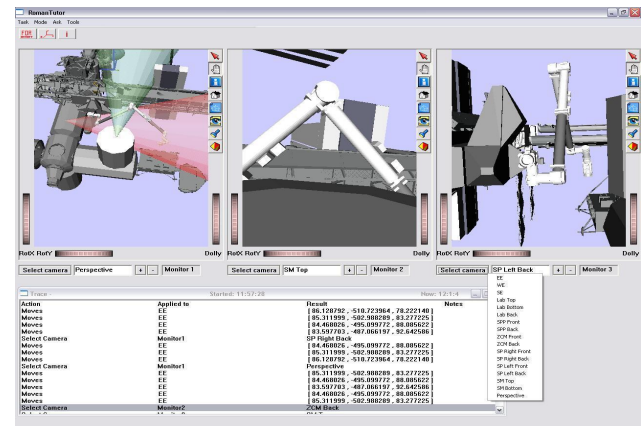


Figure 3. RomanTutor User Interface

Feedback rules can invoke automated movie generation using templates of the form “illustrate correct move from <current-configuration> to <goal-configuration>”. Such a template uses safe corridor specifications and the path-planner to generate a path with a high degree of desirability (i.e., in a safe corridor); then, it determines the best sequence of camera views for the different parts of the zone, and uses them as virtual cameras to generate a movie showing the path.

3 The Path Planner

For efficient path planning, we pre-process the robot workspace into a roadmap of collision-free robot motions in regions with highest desirability degree. More precisely, the roadmap is a graph such that every node n in the graph is labeled with its corresponding robot configuration $n.q$ and its degree of desirability $n.dd$, which is the average of *dd* of zones overlapping with $n.q$. An edge (n,n') connecting two nodes is also assigned a *dd* equal to the average of *dd* of configurations in the path-segment $(n.q,n'.q)$. The *dd* of a path (i.e., a sequence of nodes) is an average of *dd* of its edges.

Following probabilistic roadmap methods (PRM) [Sanchez and Latombe, 2001], we build the roadmap by picking robot configurations probabilistically, with a probability that is biased by the density of obstacles. A path is then a sequence of collision free edges in the

roadmap, connecting the initial and goal configuration. Following the Anytime Dynamic A* (AD*) approach [Likhachev *et al*, 2005], to get new paths when the conditions defining safe zones have dynamically changed, we can quickly re-plan by exploiting the previous roadmap. On the other hand, paths are computed through incremental improvements so the planner can be stopped at anytime to provide a collision-free path and the more time it is given, the better the path optimizes moves through desirable zones. Therefore, our planner is a combination of the traditional PRM approach [Sanchez and Latombe, 2001] and AD* [Likhachev *et al*, 2005] and it is flexible in that it can take into account zones with degrees of desirability. We call it Flexible Anytime Dynamic PRM (FADPRM).

More precisely, FADPRM works as follows. The input is: an initial configuration, a goal configuration, a 3D model of obstacles in the workspace, a 3D specification of zones with corresponding *dd*, and a 3D model of the robot. Given this input:

- To find a path connecting the input and goal configuration, we search backward from the goal towards the initial (current) robot configuration. Backward instead of forward search is done because the robot moves, hence its current configuration, is not necessarily the initial configuration; we want to re-compute a path to the same goal when the environment changes before the goal is reached.
- A probabilistic queue OPEN contains nodes of the frontier of the current roadmap (i.e., nodes are expanded because they are new or because they have previously been expanded but are no longer up to date w.r.t. to the desired path) and a list CLOSED contains non frontier nodes (i.e., nodes already expanded)
- Search consists of repeatedly picking a node from OPEN, generating its predecessors and putting the new ones or out of date ones in OPEN.
- The density of a node is the number of nodes in the roadmap with configurations that are a short distance away (proximity being an empirically set parameter, taking into account the obstacles in an application domain). The distance estimate to the goal takes into account the node's *dd* and the Euclidean distance to the goal.

A node *n* in OPEN is selected for expansion with probability proportional to

$$(1-\beta) / \text{density}(n) + \beta * \text{goal-distance-estimate}(n)$$

with $0 \leq \beta \leq 1$.

This equation implements a balance between fast-solution search and best-solution search by choosing different values for β . With β near to 0, the choice of a node to be expanded from OPEN depends only on the density around it. That is, nodes with lower density will be chosen first,

which is the heuristic used in traditional PRM approaches to guarantee the diffusion of nodes and to accelerate the search for a path [Sanchez and Latombe, 2001]. As β approaches 1, the choice of a node to be expanded from OPEN will rather depend on its estimated distance to the goal. In this case we are seeking optimality rather than speed.

- To increase the resolution of the roadmap, a new predecessor is randomly generated within a small neighborhood radius (that is, the radius is fixed empirically based on the density of obstacles in the workspace) and added to the list of successors in the roadmap generated so far. The entire list of predecessors is returned.
- Collision is delayed: detection of collisions on the edges between the current node and its predecessors is delayed until a candidate solution is found; if there is a collision, we backtrack. Collisions that have already been detected are stored in the roadmap to avoid doing them again.
- The robot may start executing the first path found.
- Concurrently, the path continues being improved by re-planning with an increased value of β .
- Changes in the environment (moving obstacles or changes in *dd* for zones) cause updates of the roadmap and re-planning.

4 Conclusion

RomanTutor's potential benefits to future training strategies are (1) the simulation of complex tasks at a low cost (e.g., using inexpensive simulation equipment and with no risk of injuries or equipment damage) and (2) the installation anywhere and anytime to provide "just in time" training. Crewmembers would be able to use it onboard of the ISS, for example, to study complex maintenance or repair operations. For very long missions, they would be able to use it to train regularly in order to maintain their skills.

References

- [Sanchez and Latombe, 2001] G. Sanchez and J-C. Latombe. A single-query bi-directional probabilistic roadmap planner with lazy collision checking. In *Proc. of 9th International Symposium on Robotics Research*, pages 403-417, 2001.
- [Larsen *et al.*, 2000] E. Larsen, S. Gottschalk, M. Lin, and D. Manocha. Fast Distance Queries using Rectangular Swept Sphere Volumes. In *Proc. of IEEE International Conference on Robotics and Automation*, 4:24-28, 2000.
- [Likhachev *et al*, 2005] M. Likhachev, D. Ferguson, A. Stentz, and S. Thrun. Anytime Dynamic A* : An Anytime Replanning Algorithm. In *Proc. of International Conference on Automated Planning and Scheduling*, June 2005.

Growth of Motor Coordination in Early Robot Learning

M.H. Lee and Q. Meng

Department of Computer Science
University of Wales, Aberystwyth
Ceredigion, SY23 3DB, Wales, UK
{mhl, qqm}@aber.ac.uk

Abstract

We present an implementation of a model of very early sensory-motor development, guided by results from developmental psychology. Behavioral acquisition and growth is demonstrated through constraint-lifting mechanisms initiated by global state variables. The results show how staged competence can be shaped by qualitative behavior changes produced by anatomical, computational and maturational constraints.

1 Introduction: developmental learning

In this paper we describe an approach to sensory-motor learning and coordination that draws from psychology rather than neuroscience. There have been many models of sensory-motor coordination but most of these have been based on specific, usually connectionist, architectures and tend to focus on a single behavioral task. We are interested in exploring mechanisms that can support not only the growth of behavior but also the transitions that are observed as behavior moves through distinct stages of competence.

Inspired by developmental psychology, we are investigating mechanisms for development in terms of stages (periods of similar behavior) and transitions (phases where new behavior patterns emerge) in very early development, in particular, the control of the limbs during the first three months of life.

Figure 1 shows our experimental robot system with two manipulator arms and a camera which is mounted on a computer-controlled pan and tilt head. In the present experiments only two joints of the arms are used, the others being held fixed. We fitted one arm with a simple probe consisting of a 10mm rod containing a small proximity sensor to detect any objects passed underneath.

2 The Motor Coordination Problem

Even before any cross-modal spatial integration can begin it is necessary to first discover the structure of the local spaces within each modality. Various stages in behavior can be discerned and during these stages the local egocentric limb space becomes assimilated into the infant's awareness and forms a substrate for future cross-modal skilled behaviors. The essential correlation between proprioceptive space and motor space



Figure 1: The laboratory robot system used in experiments

seems to be a foundation stone for development, and occurs at many levels [Pfeifer and Scheier, 1997]. Sensory-motor growth in the limbs appears to precede visual development (it may begin in the womb) and even when it can continue concurrently with visual development. For this reason, in the experiments reported here we do not involve the eye system.

A two-section limb requires a motor system that can drive each section independently. The actual biological mechanisms of proprioceptive feedback are not entirely known, and there are several encoding schemes: joints encoding, shoulder encoding, body-centered encoding, and Cartesian frame.

2.1 Mappings as a Computational Substrate

We have developed a computational framework for investigating sensory-motor coordination problem based on a two-dimensional mapping scheme. Each element in the map is represented by a patch of receptive area known as a field. The fields are circular, regularly spaced, and are overlapping. We assume that basic uniform map structures are produced by prior growth processes but they are not pre-wired for any spatial system. Our arm system has to learn the correlations between its sensory and motor signals and the mapping structure is the mechanism that supports this.

Every field in a map has a set of associated variables that can record state and other properties during operation: $F\{s, e, f, m\}$. These variables represent the value experienced, the current degree of stimulation of a field as a result of excitation or inhibition, the frequency the field has been accessed or visited, and the motor parameters that were in force when this field was stimulated, respectively.

The excitation level held in a map's fields is set to 1.0 for the first stimulation, but repeated stimulations are reduced by

a habituation function that recovers when stimulation ceases. Also a very slow decay function causes all excitation levels to fall over time. Global variable **Global excitation**, is a measure of total excitation of all field excitations above a nominal lower threshold; **Global familiarity**, is a normalized sum of field access frequency. Such global indicators can be used to signal when changes have effectively ceased and the map has become **saturated**.

2.2 Constraint lifting and reflexes

Distinct stages of competence development can be achieved by lifting constraints when high competence at a level has been reached [Rutkowska, 1994]. We have several possible constraints: the availability of contact sensing, the resolution of the proprioception sense, and the parameters of the motor system. Of course, another constraint could be not having a visual sense but this very early stage of infant growth does not rely on vision. We use global state indicators to lift constraints in two ways: finer resolution sensory maps are used when global familiarity is high, and the degree of motor randomness increases with very low global excitation.

Novelty is the motivational driver and the motor system attempts to repeat actions that cause stimulation. We provide two basic reflexes to initiate the system: go to “mouth” and return to “rest”(arm being in the lateral position).

3 Experiments and results

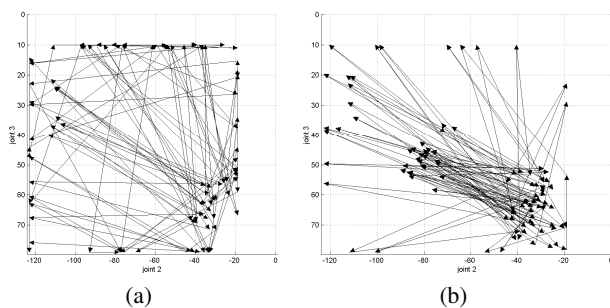


Figure 2: (a): Arm movements with no contact sensing. (b): Arm movements with active contact sensing.

The first trials used no contact sensing and objects on the table were either ignored or pushed out of range. Figure 2(a) illustrates behavior as traces of movements. As the stimulation levels of the mouth area fall due to the habituation function so random motor signals are introduced, which produce hand sweeps to points on the extreme boundary. When contact sensing is active, figure 2(b) then shows rest/mouth moves being interrupted by contact with an object on the path.

From these figures we see that the arm moved between mouth and rest areas first, but as these became less stimulated so random moves were introduced and fields on the boundary for the local body space were explored. Then, when contact sensing was allowed (a constraint lifted), internal fields and their neighbors were stimulated by object contact. Figure 3(a) shows map growth in terms of four “types” of fields. The

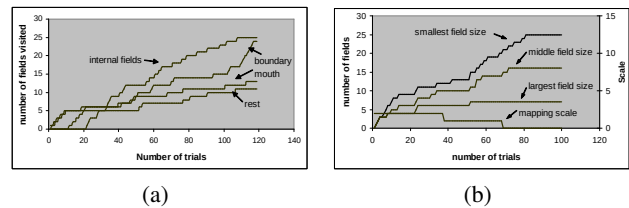


Figure 3: (a): Growth of S-M map. (b): Transitions between three maps of different scale. Repeated visits are ignored.

observed behavior is seen as series of stages: first a “blind groping” mainly directed at the mouth area, then more groping but at the boundary, these are accompanied with unaware pushing of objects, then follows more directed and repeated “touching” of detected objects. All these behaviors, including motor babbling and the rather ballistic approach to motor action, are widely reported in young infants [Piek and Carman, 1994].

Regarding proprioception, we did not observe any clear advantage in any one encoding scheme. It is likely that the encoding scheme will matter much more when hand/eye coordination is to be learned.

From the field size experiments we see a trade off: speed of exploration versus accuracy. Figure 3(b) shows how the system started on a coarse map and progressively transitioned to a finer scale map as the global familiarity variable reached a steady plateau.

Regarding the excitation parameters, the main effects are to vary the persistent actions or number of repetitions performed on a stimulus and to alter the order in which attention is given to different objects.

4 Discussion and conclusions

Very few of sensory-motor coordination studies follow the psychological literature on development and even less deal with transitions between more than one behavioral skill pattern.

The system described here records sensory-motor schemas in topological mappings of sensory-motor events, pays attention to novel or recent stimuli, repeats successful behavior, and detects when reasonable competence at a level has been achieved. The behavior observed from the experiments displays initially spontaneous movements of the limbs, followed by more “exploratory” movements, and then directed action towards contact with objects.

References

- [Pfeifer and Scheier, 1997] R. Pfeifer and C. Scheier. Sensory-motor coordination: the metaphor and beyond. *Robotics and Autonomous Systems*, 20(2):157–178, 1997.
- [Piek and Carman, 1994] J.P. Piek and R. Carman. Developmental profiles of spontaneous movements in infants. *Early Human Development*, 39:109–126, 1994.
- [Rutkowska, 1994] J. C. Rutkowska. Scaling up sensorimotor systems: Constraints from human infancy. *Adaptive Behaviour*, 2:349–373, 1994.

Measuring the Cost of Robotic Communication

Avi Rosenfeld, Gal A Kaminka, Sarit Kraus*

Department of Computer Science

Bar Ilan University, Israel

{rosenfa, galk, sarit}@cs.biu.ac.il

1 Introduction

Groups of robots are likely to accomplish certain tasks more quickly and robustly than single robots, [Jager and Nebel, 2001]. Many robotic domains such as robotic search and rescue, vacuuming, and waste cleanup are characterized by limited operating spaces where robots are likely to collide. In order to maintain group cohesion under such conditions, some type of information transfer is likely to be useful between members of the team. This is especially true as robotic domains are typically fraught with dynamics and uncertainty such as hardware failures, changing environmental conditions, and noisy sensors.

Questions such as what to communicate and to whom have been the subject of recent study [Sen *et al.*, 1994], [Jager and Nebel, 2001], [Tews, July 2001]. At times, forms of implicit coordination have been shown to allow agents better adaptability, robustness and scalability qualities [Sen *et al.*, 1994]. In theory, the lack of communication allows such methods to be implemented on simpler robots. A second series of approaches attempt to improve group performance by having robots locally communicate information [Jager and Nebel, 2001]. A third type of approach involves the use of some type of central planner [Tews, July 2001]. We believe that each type of communication framework is best suited for different environmental conditions. A mechanism is needed to match the proper system to the given environment. This paper attempts to provide such a framework with its use of a coordination cost measure.

We measure all coordination costs including the time and energy spent on coordination. We use this measure to evaluate the cost of communication. This result also allows us to create robots that alter their communication scheme when faced with dynamic domain conditions.

2 Talking the Right Amount

We model every robot's coordination cost C_i , as a factor that impacts the entire group's productivity. This cost includes the time and energy used during communication and also proactive and / or reactive collision resolution behaviors. We found that the use of communication, or lack thereof, can impact the time or energy used in the second type of behaviors.

*This material is based upon work supported in part by NSF grant #IIS-0222914, BSF grant #2002401, and the Israeli Ministry of Science and Technology. Sarit Kraus is also affiliated with UMIACS.

We also are able to create adaptive coordination methods that switch between different communication methods based on each robot's estimate of its coordination cost. When robots detect no resource conflicts, it decreases an estimate of this cost, V , by a certain amount W_{down} . When a robot senses a conflict is occurring, the value of V is increased by a certain amount W_{up} . Thus, the value V is constantly in flux based on the robot's perception of its environment. The values for V are used to switch between a set of communication schemes ranging from those with little cost overhead such as those with no communication, to more robust methods with higher overheads such as the localized and centralized methods. This approach is novel in that robots individually measure these values online during task execution and can thus adapt their communication method as needed in response to unforeseen changes such as group or environment changes.

We proceeded to implement variations of three types of communication schemes – implicit, localized, and centralized methods. In our implementation, all communication types are similar in that they resolve collisions by mutually repelling once they sensed a teammate within a certain safe distance ϵ , which we set to one robot diameter. Once robots came within this distance, they acted as they were in danger of colliding and used one of the resolution behaviors involving various types of communication schemes. Different types of communication were used to determine the length of time a robot would repel after this point.

We used the Teambots [Balch, 2000] simulator to implement these types of communication within groups of Nomad N150 robots. We measured how many pucks (out of 60) were delivered to the goal region within 9 minutes by groups of 2 – 25 robots using each communication type. We repeated each trial 100 times for statistical significance.

Within the implicit method, no time or energy was lost due to communication. This method assumed that robots were able to autonomously compute their distance from the home base. Once robots detected an impending collision, they used a function of this distance (scalar distance * 5) as the time to repel from its teammate(s). For these robots, the coordination cost was equal to the time or the energy spent in these repulsion behaviors. In our first set of experiments we measured the time spent in these behaviors. In our second set of experiments, we allocated each robot 500 units of fuel. We assumed most of the fuel was used by the robots to move, with a

smaller amount (1 unit per 100 seconds) used to maintain basic sensors and processing capabilities. Our coordination cost involved the amount of fuel used for resolving coordination issues while robots engaged in repulsion behaviors.

Our localized method initiated communication only between the robots within the ε distance. After this event, these group members would exchange information above their relative distances from their typical target, their home base. The closer robot then moved forward, while the other robot repelled for a fixed period of 20 seconds. In our first set of experiments, we measured the time these robots engaged in communication and repulsion behaviors. We assumed that 1/5 of a second was needed for this localized information transfer. In our second group of experiments, we measured our coordination cost as a function of energy spent on coordination. We again calculated this amount as a function of the distance traveled during repulsion activities with a smaller amount on sensors. In addition, we assumed each robot spent 0.3 units of fuel per localized communication exchange.

Our final method, *Latency*, used a centralized server with a database of the location of all robots. Within this method, one of two events triggered communication. First, as with the localized method, robots dropping within the ε distance initiated communication by reporting its position, only in this case with a centralized server. The server then reported back a repel value based on its relative position to other teammates. However, in order for the server to store a good estimate of the positions of all robots, a second, often more frequent type of communication was needed where each robot reported its position to the server. We used the same time and energy usage parameters from the localized experiment to measure the cost of coordination.

3 Analyzing the Cost of Communication

Our results support our claim that the best method of communication does change with domain conditions. Figure 1 contains the results from the time based coordination cost trials. The X-axis represents the group size, and the Y-axis the number of pucks successfully retrieved within each group. The implicit approach worked best in small groups where collisions were less likely. In medium sized groups, the localized approach worked better. As collisions became frequent, the large amount of communication inherent in the centralized method became justified, and this group performed significantly better. In these experiments, we found that a latency time of 1 second yield the highest productivity from various latency variations we studied.

When comparing equally sized groups, the team's productivity strongly negatively correlated to its time and energy coordination costs. On average over groups of 2 – 25 robots, we found a high statistical correlation of -0.96 between these groups' productivity and their total coordination cost. In our equivalent energy experiment, we found a correlation value of -0.95. This supports our hypothesis that coordination costs measure based on time or energy can effectively model the most effective communication method in groups.

Our measure was useful for communication adaptation as well. In our implementation, adaptation was triggered once a robot's autonomously measured coordination cost threshold,

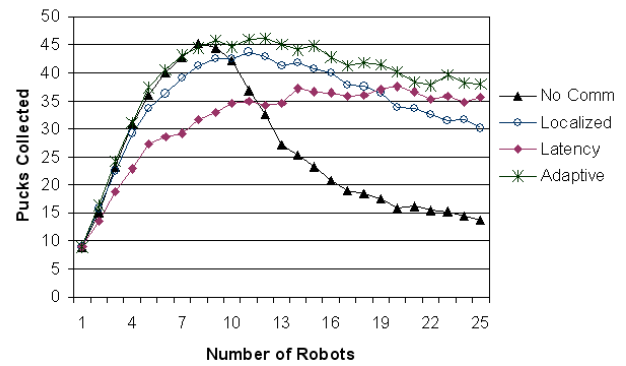


Figure 1: Adaptation Using Coordination Costs

V exceeded a certain threshold. After this point, that robot broadcasted which method it was switching to and all group members would change in kind and reinitialize their cost estimates V to this new value. Figure 1 shows the results from our time based adaptive method. We conducted the two tailed t-test to confirm the statistical significance of for our adaptive time and energy methods (not shown). In both cases we found p-values well below the 0.05 confidence interval. We believe the success of these approaches was based on the ability of these method to switch their basic communication strategy as the probability of collisions vacillated. This enabled these robots to improve their productivity in a significant fashion regardless of the group size.

4 Conclusion and Future Work

This work demonstrates how coordination costs can model the relative effective of robotic communication methods. Our measure focuses on the time and fuel spent communicating and resolving collisions. It facilitates effective comparison between implicit, localized and centralized communication methods. Using this information we are able to select the most effective communication scheme for a given domain for a group of robots. We are even able to dynamically switch between these methods in a quick, online fashion. For future work, we believe that additional expansions to our measure will facilitate comparison and adaptation even within heterogeneous groups of robots with diverse qualities.

References

- [Balch, 2000] Tucker Balch. www.teambots.org, 2000.
- [Jager and Nebel, 2001] Markus Jager and Bernhard Nebel. Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. In *IROS*, pages 1213–1219, 2001.
- [Sen et al., 1994] Sandip Sen, Mahendra Sekaran, and John Hale. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 426–431, Seattle, WA, 1994.
- [Tews, July 2001] A. Tews. Adaptive multi-robot coordination for highly dynamic environments. In *CIMCA 2001*, July 2001.

An Heuristic Search based Approach for Moving Objects Tracking

¹Elena Sánchez-Nielsen, ²Mario Hernández-Tejera

¹Department of S.O.R. and Computer Science
University of La Laguna, 38271 La Laguna, Spain
enielsen@ull.es

²Institute of Intelligent Systems and Numerical Applications in Engineering,
University Campus of Tafira, 35017 Gran Canaria, Spain
mhernandez@isusiani.ulpgc.es

Abstract

Fast and accurate tracking of moving objects in video streams is a critical process in computer vision. This problem can be formulated as exploration problems and thus can be expressed as a search into a state space based representation approach. However, these search problems are hard to solve because they involve search through a high dimensional space. In this paper, we describe an A* heuristic search for computing efficient search through a space of transformations corresponding to the 2D motion of the object, where most promising search alternatives are computed by means of integrating target dynamics into the search process, and ideas from information theory are used to guide the search. The paper includes evaluations with video streams that illustrate the efficiency and suitability for real-time vision tasks on general purpose hardware. Moreover, the computational cost to carry out the tracking task is smaller than real time requirements (40 ms).

1 Problem Formulation

Let $T = \{t_1, \dots, t_r\} \subseteq \mathbb{R}^2$ be a set of points that represent a template, let $I = \{i_1, \dots, i_s\} \subseteq \mathbb{R}^2$ be another set of points that denote an input image, let a bounded set of translational transformations $\mathbf{G}(\cdot)$ be a set of transformations $G: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ consisting of translations, which are parameterized by a vector $(\Delta x, \Delta y)^T \in [\Delta_{\min}, \Delta_{\max}] \times [\Delta_{\min}, \Delta_{\max}] \subseteq \mathbb{R}^2$ and let a bounded error notion of quality of match $Q(G; T, I, \varepsilon)$ be a measurement for computing the degree of match between a template T and a current input image I . The quality of match assigned to a transformation G is represented by the allowed error bound ε , when template points are brought to point's image using a transformation G :

$$Q(G) = \sum_{t \in T} \max_{i \in I} \|G(t) - i\| < \varepsilon \quad (1)$$

Given a template T , an input image I and an error bound ε , template position matching problem can be viewed as a search process in the space of transformations in order

to find the transformation, $G_{\max}$, that maximizes the quality of match $Q(G)$:

$$G_{\max}(T, I, \varepsilon) = \arg \max_{G \in \mathbf{G}} Q(G; T, I, \varepsilon) \quad (2)$$

2 A* Search Algorithm

Problem solving through the use of heuristic search strategies is expressed through a *state space based representation approach* [Pearl, 1984]. According to the heuristic search framework, template position matching problem is formulated as: *the search process oriented to find the transformation parameters $G_{\max} \in \mathbf{G}(\cdot)$ that maximizes the quality of function match $Q(G)$ between the transformed template $G(T)$ and the current image I* . Next, each one of the elements of the problem is described:

- *State*: each state n is associated with a subset $G_k \subseteq \mathbf{G}(\cdot)$, which is represented by the transformation corresponding to the centre that is referred to as G_c .
- *Initial state*: is represented by a bounded set of translational transformations $\mathbf{G}(\cdot)$, which allowing matching the current template into the current scene.
- *Final state*: is the transformation that best matches the current template points into the current scene, according to the quality of function $Q(G)$. The quality of function match assigned to a transformation G is expressed in terms of the partial directed Hausdorff distance between the transformed template T and the current image I :

$$Q(G) = h_k(G(T), I) < \varepsilon \quad (3)$$

That is, the final state corresponds to the state with the translation $G_{\max}$ which verifies that the distance between 80% of translated template points and current image points are less than 2.0.

- *Operators*: for each current state n , the operators A and B are computed:
 - Each partial set of transformations from the current state is partitioned into four regions by vertical and horizontal bisections (that is, four new states).
 - The quality of function match (equation 3) is computed for each one of the four new states.

Splitting operation generates an expanded quaternary tree where with each node a $2^i \times 2^j$ region is associated. The splitting operation is finished when the quadrisection process computes a translational motion according to the quality of function match $Q(G)$. The best node to expand is computed through the use of an A* approach [Pearl, 1984]. The corresponding cost value assigned to each state n is defined as:

$$f(n) = c(n) + h^*(n) \quad (4)$$

$h^*(n)$ is formulated in terms of expanding the best state of the search space based on measuring the similarity among the distribution functions that characterize the current state n and the objective state, denoted respectively by P and Q . The distribution function P is represented by a histogram distances $\{H_{G_c}\}_{i=1..r}$, which contains the number of template points T at distance d_j with respect to the points of the input image I , when the transformation G_c of the current state n is applied on the current template T . The distribution function Q is modelled by approximating $\{H_{G_c}\}_{i=1..r}$ by an exponential distribution function $f(n) = ke^{-an}$.

The similarity between the distributions P and Q is evaluated through the relative entropy or Kullback-Leibler distance measurement from information theory [Cover and Thomas, 1991]:

$$D(P \parallel Q) = \sum_{i=1}^R p_i \log \frac{p_i}{q_i} \quad (5)$$

An estimated cost function $c(n)$ is employed to lead the search process towards promising solutions. With this aim, a depth term is added to the $f(n)$ function (expression 4). This term is based on the number of operators of type A applied from the initial search node to the current node n .

An alpha-beta predictive filtering [Shalom and Rong, 1993] is incorporated into the search algorithm with the purpose of predicting target position and reducing the initial size of the search space of transformations, focused on the assumption that there is a relationship between the size of the search area and the resulting uncertainty of the alpha-beta predictive filtering. Figure 1 illustrates the integration of the stages that compose the alpha-beta filtering with the computation of the dimension $M \times N$ of $G(\cdot)$ and their respective equations.

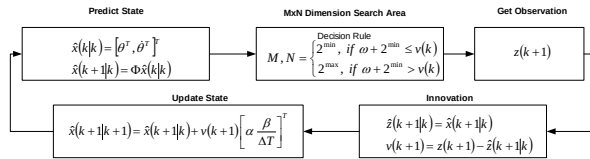


Figure 1. Computing adjustable-size based search area.

The algorithm for template position matching in our approach based on an A* search is as follows.

Input

$G(\cdot)$: initial set of transformations.

$\mathcal{E}$: distance error bound allowed when template points are brought to point's image using a transformation G .

$v(k)$: uncertainty measurement computed by alpha-beta filtering approach.

ω : estimate which determines from what value should be selected $2^{\max}$ or $2^{\min}$ bound value for computing $M \times N$ dimension of $G(\cdot)$.

$2^{\max}$ and $2^{\min}$: upper and lower nearest values to the innovation factor $v(k)$.

η : number of operators of type A applied from the initial search state to the current state n .

Output

$M \times N$ dimension of initial state set $G(\cdot)$

$$G_{\max} = [g_x^{\max}, g_y^{\max}]$$

Algorithm

Step 1) Compute adjustable $M \times N$ dimension of initial state set $G(\cdot)$:

$$M, N = \begin{cases} 2^{\min}, & \text{if } \omega + 2^{\min} \leq v(k) \\ 2^{\max}, & \text{if } \omega + 2^{\min} > v(k) \end{cases}$$

Step 2) Find $G_{\max}$ such the quality of function match $Q(G) = h_k(G(T), I) < \mathcal{E}$ is verified:

While ($Q(G) > \mathcal{E}$) **Do**

2.1) Split current state n into four new states $\{n_i\}_{i=1..4}$

2.2) Compute $Q(G_c) \leftarrow h_k(G_c(T), I)$ for each new n_i

2.3) Expand the best state n_i according the evaluation function $f(n) = c(n) + h^*(n)$:

2.3.1) $h^*(n) \leftarrow D(P \parallel Q)$

2.3.2) $c(n) \leftarrow c(n-1) + \eta$

End While

Step 3) Output $G_{\max} = [g_x^{\max}, g_y^{\max}]$

3 Comparative Analysis and Runtime

Diverse experiments have been carried out with 24 different sequences, achieving the same behaviour for all of them on a P-IV 2.4 GHz. Comparing the results achieved from the proposed A* heuristic search that uses an adjustable-size search area approach with the blind search strategy that does not use information to guide the search process and does not estimate the dimension appropriated for initial search state such as the conventional search strategy described by [Rucklidge, 1996], demonstrates that the proposed A* heuristic search framework is computationally lighter and as consequence, faster than the blind search strategy, in an average rate of three times better. Average time measured in seconds for processing each frame for the 24 different sequences is 0.01, confirming the adaptation of the search strategy proposed to real-time restrictions under unrestricted environments for arbitrary shapes.

References

- [Cover and Thomas, 1991] T. Cover and J. Thomas. Elements of Information Theory. John Wiley & Sons Incs, 1991.
- [Pearl, 1984] J. Pearl. Heuristics. Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley Series in Artificial Intelligence, 1984.
- [Rucklidge, 1996] W. Rucklidge. Efficient computation of the minimum Hausdorff distance for visual recognition. Lecture Notes in Computer Science, 1173, Springer-Verlag, 1996.
- [Shalom and Rong, 1993] Estimation and Tracking: Principles, Techniques, and Software. Artech House, Boston, 1993.

3-D Interpretation of Single Line Drawings

Kenji Shoji, Fubito Toyama, and Juichi Miyamichi

Utsunomiya University

Department of Information Science

7-1-2 Yoto, Utsunomiya 321-8585, Japan

shoji@is.utsunomiya-u.ac.jp

Abstract

The human visual system can interpret two dimensional (2-D) line drawings like the Necker cube as three dimensional (3-D) wire frames. We focus attention on a principle to minimize the entropy of angle distribution between line segments in a 3-D wire frame as a concrete definition of the law of pragnanz in Gestalt Psychology. And we implement the principle with the perceptual preference of planarity to the loops of wire frames using a genetic algorithm. Experimental results show the good coincidence with human perception.

1 Introduction

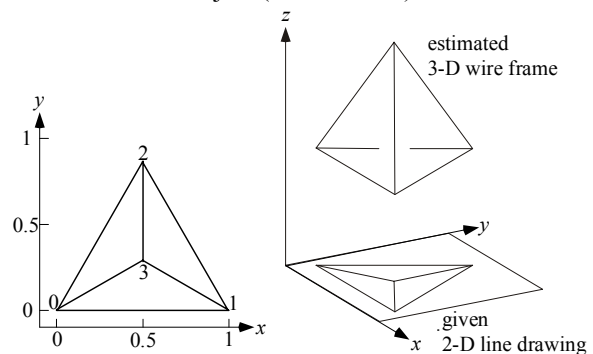
Line drawings are more often used in documents which illustrate the functions and shapes of products, buildings, and so on rather than the photo-realistic images of them created by computer graphics, because line drawings are more suitable to convey the shapes of 3-D objects than the photo-realistic images.

Single line drawings have many possibilities in 3-D interpretation, that is, there are infinite 3-D shapes which produce a same line drawing. Human vision selects one or two of them based on some internal criteria. Previously, Shoji et. al. [2001] proposed a principle to minimize the entropy of angle distribution (MEAD) between line segments in a 3-D wire frame as a concrete definition of the law of pragnanz in Gestalt Psychology. The MEAD principle seems to be appropriate, but the results of their simulation are not sufficient to support the principle.

In this work, we focus attention on the MEAD principle and other perceptual tendencies. And we propose a 3-D interpretation model based on the MEAD principle and the perceptual preference of planarity to the loops of wire frames. In the implementation of the model for simulation we use a genetic algorithm (GA) based on the minimal generation gap (MGG) model as a generation alternation model and the unimodal normal distribution crossover (UNDX) as a crossover [Ono et. al., 1998].

2 Definition of Problem

Line drawings as the input of a simulation program are represented by the x and y coordinates of vertices and the pairs of vertex numbers which are corresponding to the both ends of line segments. For example, the line drawing in figura 1 (a) is represented by the list of x - y coordinates of vertices #0 to #3 and the list of pairs of vertex numbers such as (#0, #1), (#1, #2), (#2, #0), (#0, #3), and so on. We suppose the type of projection to be approximately orthographic. Then, the task of the simulation program is to estimate the z coordinate of each vertex. Estimated z coordinates and given x and y coordinates produce a 3-D wire frame object. Figura 1 (b) shows the x - y - z coordinate system, an input line drawing, and an estimated 3-D object (a wire frame).



(a) An example of line drawing. (b) Relationship between a given 2-D line drawing and an estimated 3-D wire frame.

Figure 1: A given 2-D line drawing and an estimated 3-D wire frame.

3 Model and its implementation

A proposed model for 3-D interpretation is to search an optimal solution that maximizes the weighted sum F of evaluation values F_e and F_p based on the MEAD principle and the perceptual preference of planarity to the loops of wire frames, respectively. That is,

$$F = F_e + w_p F_p$$

where w_p is a weight and its value was 0.5 in simulation.

3.1 MEAD principle

The evaluation value F_e based on the MEAD principle is,

$$F_e = 1/H$$

where H is the entropy of angle distribution as follows:

$$H = -\int_0^{180} p(\theta) \log p(\theta) d\theta$$

where $p(\theta)$ is a probability density function of angle between two 3-D line segments sharing an end point which is a vertex of a 3-D wire frame. Figure 2 shows an example of angle distribution.

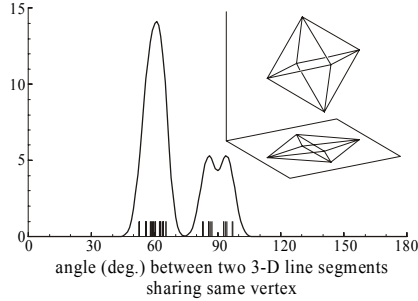


Figure 2: An example of angle distribution.

3.2 Planarity to loops

Loop is a cycle of graph of vertices connected by line segments of a wire frame as shown in figure 3. Perception tends to interpret such loops as flat faces of polyhedron. The evaluation value F_p represents this perceptual preference as

$$F_p = 1 - \frac{\sum_{i \in L} \lambda_i}{\sum_{i \in L} S_i}$$

where L is the set of the indices of loops having 4 or more vertices. λ_i is the minimal eigen value of the covariance matrix of x , y , and z coordinates of the set of vertices on the i -th loop. S_i is the area of i -th loop on xy plane.

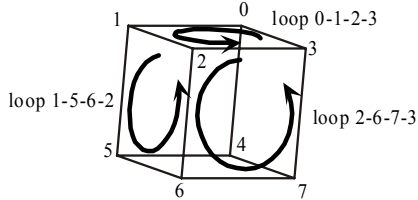


Figure 3: Loops of a wire frame.

3.3 GA with MGG and UNDX

For the optimization the set of real numbers, a real-coded GA is often used. We also use it with MMG and UNDX [Ono et. al., 1998]. The chromosome consists of the sequence of z coordinates. We set a population size to be 200, the number of crossovers to be 200, the number of generations to be 4,000. That is, we randomly pick up two sequences from the 200 sequences of z coordinates (the population), and generate 400 sequences based on the UNDX referring the two sequences and a sequence randomly selected from the rest of the population. The 400 plus two sequences are evaluated and sorted by F value, and the best and a sequence selected by roulette based on its rank are return to the population. The

above procedure is repeated 4,000 times and the best in the last population is the simulation result.

4 Simulation

We used the data of 18 line drawings for simulation that were used by Shoji et. al. [2001]. The data can be obtained from the URL <http://www.ccs.neu.edu/home/feneric/msdsm.html>. All of these line drawings are regular. So, we also prepare irregular ones. From all of the regular 18 line drawing the proposed model makes 3-D wire frames coincident with perception, while Shoji's model using only the MEAD fails in 3 of the 18 line drawings. On the irregular line drawings which are made from a quadrangular pyramid truncated two non-parallel planes, the proposed model also succeeds. Only the characteristic results are shown in figure 4.

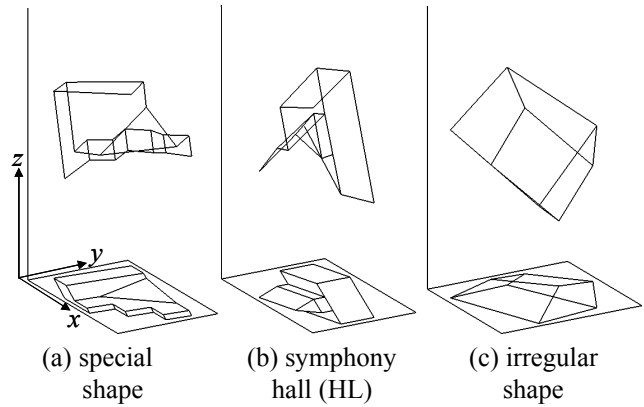


Figure 4: Examples of the results of the simulation. 2-D line drawings are shown on xy planes.

5 Conclusions

We implemented the proposed model based on the MEAD principle and the perceptual preference of planarity to the loops of wire frames using a genetic algorithm. Simulation results show the good coincidence with human perception. As a future work, we plan to use the proposed model for the evaluation engine of the e-learning system for illustrators. That is, a learner draws a line drawing observing an actual wire frame object, and then the system evaluates the drawing comparing the 3-D wire frame made from it with the 3-D data of the original one.

References

- [Ono et. al., 1998] Isao Ono, Yoshihiro Tatsuzawa and Shigenobu Kobayashi. Evolutionary design of lens system, In *Proceedings of the 3rd International Symposium on Artificial Life and Robotics*, pages 81-84, Beppu, Japan, Jan 1998.
- [Shoji et. al., 2001] Kenji Shoji, Kazunori Kato, and Fubito Toyama. 3-D Interpretation of Single Line Drawings Based on Entropy Minimization Principle, In *2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 90-95, Kauai, USA, Dec 2001.

SVM-based Obstacles Recognition for Road Vehicle Applications

M. A. Sotelo, J. Nuevo, D. Fernandez, I. Parra, L. M. Bergasa, M. Ocana, R. Flores

Department of Electronics

University of Alcala

Campus Universitario s/n Alcala de Henares, Madrid, Spain

michael@depeca.uah.es

Abstract

This paper describes an obstacle Recognition System based on SVM and vision. The basic components of the detected objects are first located in the image and then combined with a SVM-based classifier. A distributed learning approach is proposed in order to better deal with objects variability, illumination conditions, partial occlusions and rotations. A large database containing thousands of object examples extracted from real road images has been created for learning purposes. We present and discuss the results achieved up to date.

1 Introduction

This paper describes an SVM-based object recognition system that can recognise both vehicles and pedestrians using vision. In our approach, the basic components of the objects are first located in the image and then combined with a SVM-based classifier. Our object detection technique is characterised by example-based learning algorithms. The salient features of a class are learnt by the system based on a set of examples. Example-based techniques have been previously used in natural, cluttered environments for pedestrian detection [Shashua, 2004]. In general, these techniques are easy to use with objects composed of distinct identifiable parts arranged in a well-defined configuration. This is the case of road vehicles, where a distributed learning approach based on components [Mohan, 2001] is more efficient for object recognition in real cluttered environments than holistic approaches [Papageorgiou, 2000]. Distributed learning techniques can deal with partial occlusions and are less sensitive to object rotations. The use of SVMs is a viable option as long as we intend to discriminate between two classes: car and non-car.

2 System Description

The system is divided in two modular subsystems. The first subsystem is responsible for vehicle detection and tracking. The second subsystem provides pedestrians detection using the information obtained by the vehicle detection module. In this paper, we focus on the vehicle recognition system alone, working with 320x240 monochrome images. The objects searching space is reduced by using the limits established by

the estimated lane markings. This helps reduce the rate of false positive detections. In case that no lane markings are detected, a basic area of interest is used instead covering the front part ahead of the host-vehicle. In a first stage, an attention mechanism has been devised with the intention of filtering out inappropriate candidate windows based on the lack of distinctive features, such as horizontal edges and symmetrical structures, which are essential characteristics of road vehicles. This has the positive effect of decreasing both the total computation time and the rate of false positive detections. Each road lane is sequentially scanned, from the bottom to the horizon line of the image, looking for collections of horizontal edges that might represent a potential obstacle. We use a distributed learning approach in which each individual part of the vehicle is independently learnt by a specialized classifier in a first learning stage. The local parts are then integrated by another classifier in a second learning stage. We have considered a total of 3 different sub-regions for each candidate region, covering the most characteristic parts of the vehicle. Two small sub-regions have been located in the area of the region where the wheels are supposed to be. A third sub-region is located in the central part of the region, covering the area where car plates and rear windshield are usually placed. The locations of the three sub-regions have been chosen in an attempt to detect coherent and structural car features. A set of features must be extracted from each sub-region and fed to the classifier. Before doing that, the entire candidate region of interest is pre-processed using a Canny operator in order to enhance the differential information contained in it (edges). The Canny image provides a good representation of the discriminating features of the car class. On the one hand, edges, both horizontal and vertical, are clearly visible and distinguishable. On the other hand, the vertical symmetry of a car remains unchanged. In addition, edges are not affected by colours or intensity. This property makes the use of edges robust enough to different car models of the same type. The pre-processed sub-region is directly applied to the input of the classifier. The dimensions of the entire region of interest are normalized before being fed to the classifier. A size of 70x80 pixels has been chosen, as it is adequate for detecting vehicles at long distances (up to 80 meters).

The global training strategy is carried out in two stages. In a first stage, separate SVM-based classifiers are trained using individual training sets that represent a subset of a sub-region.

This stage provides classification of individual parts of the candidate sub-regions. In a second step, the outputs of all classifiers are merged in a single SVM classifier in order to provide the final classification result.

3 Results

The system was implemented on a Apple PC at 2.0 GHz running the Debian GNU/Linux Operating System. The complete algorithm runs at 25 frames/s. We created a preliminary database containing 2000 samples of road vehicles. The samples were extracted from recorded images acquired in real experiments onboard a road vehicle in real traffic conditions in Madrid. Two different training sets were built for the same sub-region in different conditions in order to decrease the complexity of the training process. This yields a total of 6 training sets (2x3). All training sets were created at day time conditions using the TsetBuilder [Nuevo, 2005] tool, specifically developed in this work for this purpose. By using the TsetBuilder tool different candidate regions are manually selected in the image on a frame-by-frame basis. This allows to select candidate regions containing vehicles of different size, from different manufacturers, and so on. The number of non-vehicle samples in the training sets was chosen to be similar to the number of vehicle samples. We obtained a detection rate of 85% in a test set containing 1000 images, and a false detection rate of 5%. No image from the training set was used in the test set. As an example, figure 1 shows a sequence of images in which a vehicle is detected and tracked along the lane of the host vehicle. A blue box is overprinted over the detected vehicle indicating the estimated distance measured from the host vehicle. Other vehicles appearing along the adjoining lane are marked with a horizontal red line.

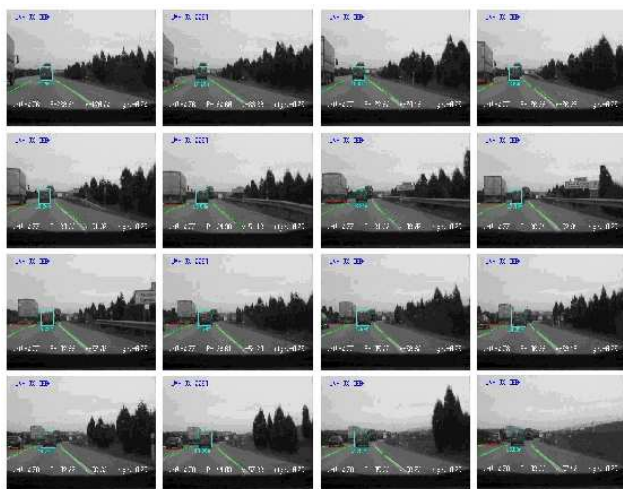


Figure 1: Vehicle detection and tracking in a sequence.

4 Conclusions and Future Work

We have developed a visual multi-frame two-stage object classification system based on Support Vector Machines

(SVM) [Boser, 1992]. The learning process has been simplified by decomposing the candidate regions into 3 local sub-regions that are easily learned by individual SVM classifiers. Several training sets have been built for each sub-region in order to cope with different weather and illumination conditions. The results achieved up to date using a set of 2000 samples are encouraging. Nevertheless they still need to be improved before being safely used as an assistance driving system onboard road vehicles in real conditions. For this purpose, the content of the training sets will be largely increased by including new and more complex samples that will boost the classifier performance. In addition, the attention mechanism will be refined in order to provide more candidates around the original candidate region. This will reduce the number of candidate regions that only contain a part of the vehicle, i.e., those cases in which the entire vehicle is not completely visible in the candidate region due to a misdetection of the attention mechanism.

Acknowledgment

This work has been funded by Research Projects CICYT DPI2002-04064-05-04 and FOM2002-002 (Ministerio de Fomento, Spain).

References

- [Shashua, 2004] A. Shashua, Y. Gdalyahu, and G. Hayu. Pedestrian detection for driving assistance systems: single-frame classification and system level performance. *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 1–6, Parma, Italy, June 2004.
- [Mohan, 2001] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23, April 2001.
- [Papageorgiou, 2000] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38,1:15–33, 2000.
- [Boser, 1992] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning*, 1992.
- [Nuevo, 2005] J. Nuevo. TsetBuilder User Manual. *Technical Report. University of Alcala*. <ftp://www.depeca.uah.es/pub/vision/SVM/manual.pdf>, 2005.

Detecting and locating faults in the control software of autonomous mobile robots *

Gerald Steinbauer and Franz Wotawa

Institute for Software Technology

Technische Universität Graz

8010 Graz, Inffeldgasse 16b/2, Austria

{steinbauer,wotawa}@ist.tugraz.at

1 Introduction

Control software of autonomous mobile robots is characterized by its fairly high complexity which is in conflict with stability. Complexity is caused by the software components implementing the basic functionality like planning, sensor fusion, actuator interfaces, and their interactions. Because of the high complexity but also the instability of hardware components and connections, and the underlying operating system, complete stability is very unlikely. But if we want to build a robot that is truly autonomous it has to deal with failures during its runtime without degrading the desired behavior or even worse failing to fulfill its mission. Hence, a diagnosis system on top of the control system which does monitoring the current behavior, locating the cause of a detected failure, and taking the appropriate actions is a necessity.

To achieve this we introduce a model-based solution. This includes a model of software components and their relationships that are specified in the software architecture of the robot's control system. This model is then used to derive root causes of a detected failure. The failure detection is based on observations. For this we use the concept of observers. If the monitored value exceeds its pre-specified boundaries, the observer raises a conflict which causes the diagnosis engine to compute the root causes. Once the root cause has been identified, the diagnosis system takes appropriate actions in order to retain the system's correct behavior. The fault detection, localization, and correction procedures are all based on declarative models of the control software.

2 Modeling software architectures

Software architectures comprise software components and their connections. Components represent a collection of classes which implement a certain behavior. The connections between components represent dependency relations like client-server relationships and data flow. During the execution of a program the components might spawn processes and interact using method calls, or other means of communication, like events. Figure 1 depicts parts of the software architecture of our mobile robot.

The formalization of the structural properties of the software architecture considers the software components, their

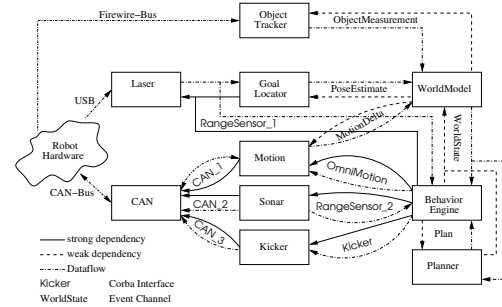


Figure 1: Dependencies between software modules.

connections in terms of identifiers representing events or procedure calls, and a classification of dependency relations which are used to repair the software during runtime. We distinguish two different dependencies between components: weak and strong. Two components are weakly dependent if killing one component directly affects the other component. Otherwise, the relationship is a strong dependency.

A software architecture model (SAM) comprises a set of software components CO , a set of connections C , a function $out : CO \mapsto 2^C$ returning the output connections for a given component, a function $in : CO \times C \mapsto 2^C$ returning the input connections for a given component and output connection, a set of weak dependencies $WDC \subseteq 2^{CO \times CO}$ and a set of strong dependencies $SDC \subseteq 2^{CO \times CO}$.

The concrete behavior of the software at runtime is determined by the implemented behavior of its software components. An abstraction of the concrete behavior is necessary in order to get a declarative model of the system's behavior. The idea behind the abstract behavior model of software components is similar to models which are based on dependencies like the one described by [Friedrich *et al.*, 1999]. If all inputs to the model are correct, a software component should produce a correct output. This conversion has to be performed for all components and their output connections.

For example, the rule that represents the abstract behavior of the OT (Object Tracker) component is: $\neg AB(OT) \wedge ok(Firewire) \rightarrow ok(ObjectMeasurement)$ where the predicate AB stands for abnormal, and ok indicates a correct event or method call.

In order to locate root causes, i.e., the components which

*This research has been funded in part by the Austrian Science Fund (FWF) under grant P17963-N04.

cause a detected misbehavior, we have to introduce a notation of observations. The easiest way of doing this is to use the same *ok* predicate for the purpose. If we detect a misbehavior at *OM* (*ObjectMeasurement*), we could represent this by the literal $\neg ok(OM)$. We introduce a distinguished predicate *correct* for observations to distinguish observations and computed values. *correct*(*x*) for a connection *x* is true whenever the observed connection shows the correct behavior. Otherwise *correct*(*x*) is false. The appropriate model for *OT* and *OM* is: $correct(OM) \rightarrow ok(OM)$, $\neg correct(OM) \rightarrow \neg ok(OM)$, $correct(OM) \rightarrow \neg AB(OT)$

3 Monitoring, Diagnosis and Repair

Coupling the running program with its software architecture model requires an abstraction step. The running program changes its state via changing variable values which is caused by inputs from the environment. But SAM only represents the software components and their communication means. Therefore, we require to map changes to communication patterns. For this purpose we introduce observers. An observer monitors a certain part of the program's behavior during the execution. If an observer detects a behavior that contradicts its specification, it computes the appropriate observations in terms of setting the observation predicates $\neg correct(x)$ for the connection *x*, and invokes the diagnosis engine. In the current implementation we used the following observers: *Periodic event production* checks whether an event *e* is produced at least every *n* ms, *Conditional event production* checks whether an event *e*₁ is produced *n* ms after the occurrence of an event *e*₂, *Spawn processes* checks whether a component spawns a number *n* of processes and *Periodic method calls* checks whether a component calls a method *m* at least every *n* ms.

The observers are used to monitor the state of the system. For this purpose the observers check their rules on a regular basis. In cases of failure the diagnosis procedure is invoked.

The diagnosis task is based on the model-based diagnosis (MBD) paradigm [Reiter, 1987]. In particular we use Reiter's hitting set algorithm [Reiter, 1987]. In order to minimize diagnosis time we only search for minimal cardinality diagnoses which can be easily obtained when using Reiter's algorithm. We only construct the hitting set graph until a level where the first diagnosis is computed. In most practical cases single fault diagnoses can be found.

After diagnosis those components that are responsible for a detected failure have to be killed and restarted. We have to take care of the fact that restarting one component might require restarting another component. This can be done by using the information about strong dependencies between components. The components that have a strong dependency relationship with each other have to be restarted. Hence, the steps for repair would be: (1) compute the diagnoses. (2) compute a set of components that have to be restarted. In this step we compute all components that strongly depend on components of a diagnosis. (3) Maximize the chance of repair by using a larger set of components to be restarted.

The proposed diagnosis system has been tested on our mobile robot system. For the evaluation of the diagnosis we introduced artificial faults into the robot control system and an-

alyzed if the diagnosis system detected and located the fault and recovered the control system.

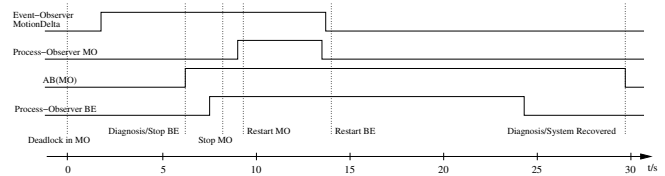


Figure 2: Diagnosis and repair of a fault in the motion service.

Figure 2 shows the results for an introduced deadlock in the motion service (MO). After introducing the deadlock in MO the Event Observer for the event *MotionDelta* perceives that no more events are produced. The diagnosis kernel derived that MO is malfunctioning. Instantly the repair process starts. The repair action comprises a stop of the Behavior Engine (BE), a stop of MO, and a restart of MO and BE. The restart of BE is necessary because BE is strongly coupled with MO. After repair the diagnosis kernel derives the diagnosis that all components work properly now. The relatively long time for the recovery could be explained by the fact that a repair of services could take a while because of the required starting, stopping and re-configuration of hardware components.

4 Related research and conclusion

Williams and colleagues [Williams et al., 1998] used model-based reasoning to detect and recover failures of the hardware of a space probe. Verma and colleagues [Verma et al., 2004] used particle filter to estimate the state of the robot and its environment. These estimations together with a model of the robot were used to detect faults. Previous research has dealt either with hardware diagnosis or diagnosis of software as part of the software engineering cycle. However, diagnosis of software and repair at runtime has never been an issue. The paper described a model-based diagnosis for detecting, locating and repairing faulty software at runtime. For this purpose a modeling technique for representing software architectures which include components, control and data flow, and dependencies has been introduced. Moreover, the concept of observers, have been described in the paper. Finally, the paper presented a repair algorithm and first empirical results of our implementation. These results show that software failures, e.g., deadlocks, can be detected and corrected at runtime.

References

- [Friedrich et al., 1999] Gerhard Friedrich, Markus Stumptner, and Franz Wotawa. Model-based diagnosis of hardware designs. *Artificial Intelligence*, 111(2):3–39, 1999.
- [Reiter, 1987] Raymond Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95, 1987.
- [Verma et al., 2004] V. Verma, G. Gordon, R. Simmons, and S. Thrun. Real-time fault diagnosis. *IEEE Robotics & Automation Magazine*, 11(2):56 – 66, 2004.
- [Williams et al., 1998] B. C. Williams et al. Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1-2):5–48, August 1998.

Learning discontinuities for switching between local models *

Marc Toussaint and Sethu Vijayakumar

Institute of Perception, Action and Behavior

School of Informatics, University of Edinburgh

The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK.

mtoussai@inf.ed.ac.uk, sethu.vijayakumar@ed.ac.uk

1 Introduction

Locally weighted learning techniques, in particular LWPR [Vijayakumar *et al.*, 2002], have successfully been used for high-dimensional regression problems. Their robustness and efficient online versions are crucial in robotic domains where, for instance, an inverse model of an articulated dynamic robot has to be learned in real-time. Such models map a high-dimensional state (e.g., joint angles and velocities) and a desired change of state to the required motor signals (torques).

While typically such mappings are assumed to be smooth, in real world scenarios, there are many interesting cases where the functions of interest are truly discontinuous. Some examples include contacts with other objects (in particular the ground), with other parts of the body, or with “joint limits”. In fact, many interesting interactions with the environment manifest themselves through discontinuities in the sensorimotor data.

In this paper, we show how discontinuous switching between local regression models can be learned. The general topic of switching models has been discussed before, e.g., in the context of state space models [Ghahramani and Hinton, 1998; Pavlovic *et al.*, 2000] or multiple inverse models [Wolpert and Kawato, 1998]. Generally, the question of which particular model receives responsibilities for a given input can be modeled as a hidden variable i in a generative mixture model. In our case, we assume that the responsibility index i can be predicted from the input (the robot state). Thus, inferring a model for i corresponds to classifying the input domain into regions for each sub-model.

Since in robotic domains, local learning is crucial to prevent interference and allow for online adaptation techniques, we propose a model of the responsibility index i which is itself a composition of local classifiers. Multiple pairwise classifiers are concatenated to construct a complete model in the form of a product-of-sigmoids, which is capable of learning complex, sharply bounded domains for each local model in lieu of the typical Gaussian kernels.

2 Learning a family of models

Given training data $\{(\mathbf{x}_k, y_k)\}_{k=1}^M$ with inputs $\mathbf{x}_k$ and outputs y_k , the first level goal of our algorithm is to learn a *family* of models $\{\phi_1, \dots, \phi_N\}$ such that every datum can be explained

by at least one model. The problem of predicting which particular model ϕ_i is responsible for a given input $\mathbf{x}$ is solved on a higher level as explained in the next section. In formal notations, we assume a mixture model

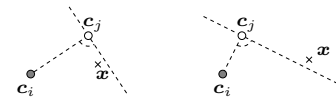
$$P(y|\mathbf{x}) = (1 - \epsilon) \sum_{i=1}^N P_{\text{loc}}(i|\mathbf{x}) P(y|i, \mathbf{x}) + \epsilon \mathcal{U}(y),$$

$$P(y|i, \mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{|y - \phi_i(\mathbf{x})|^2}{2\sigma^2} \right\},$$

where $\mathcal{U}(y)$ is a uniform distribution accounting for background noise (e.g. outliers) and i is the hidden variable specifying the particular model that generates a datum. Since we aim for localized models, we impose a locality constraint already at this level as follows: Let $\mathbf{c}_i$ denote the mean input (*center*) on which model ϕ_i has been trained on. For a given input $\mathbf{x}$, the i th model is eligible if and only if there does not exist a j th model which has its center “between” $\mathbf{c}_i$ and $\mathbf{x}$. More precisely,

$$P_{\text{loc}}(i|\mathbf{x}) = 0 \iff \exists j : \langle \mathbf{x} - \mathbf{c}_j, \mathbf{c}_i - \mathbf{c}_j \rangle < 0,$$

where $\langle \cdot, \cdot \rangle$ is the scalar product in input space:



Further, $P_{\text{loc}}(i|\mathbf{x})$ is uniform over all eligible i 's. Given a current family of models, we can infer a posterior on the responsibility index i for a given datum $(\mathbf{x}, y)$, using Bayes rule:

$$P(i|y, \mathbf{x}) = \frac{1}{Z} P(y|i, \mathbf{x}) P_{\text{loc}}(i|\mathbf{x}).$$

Calculating the MAP assignment $\hat{i}$ allows us to associate every training datum with the most likely model. Using this, the sufficient statistics of each local model ϕ_i is updated. Further, the data that is labeled as “yet unmodeled” (which is inferred to have been generated by $\mathcal{U}(y)$) is used to generate a new family member by the following heuristic (compare RANSAC): A random datum $(\mathbf{x}, y)$ is selected from the unmodeled data; the K closest neighbors of $(\mathbf{x}, y)$ (w.r.t., Euclidean input distance) are chosen as initial training data for the new model, where K is a random Poisson number with mean $3d$ (here, d is the input dimensionality). Finally, models that receive too few MAP responsibilities (less than 10 d in our experiments) are discarded. This iterative process can be repeated until no new models are generated.

*The first author acknowledges support by the German Research Foundation (DFG), Emmy Noether fellowship TO 409/1-1.

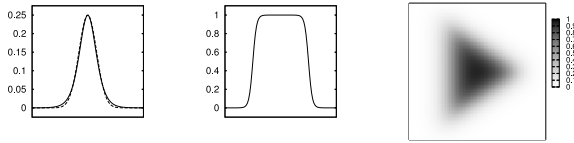


Figure 1: Kernels that can be represented as a product of sigmoids in 1D and 2D.

This general scheme of family learning can be realized with any type of models ϕ_i . In the experiments, we will choose ϕ_i to be linear functions, learned with Partial Least Squares (PLS) regression. PLS, involving an intermediate lower-dimensional projection, has been proven efficient for high-dimensional problems [Vijayakumar *et al.*, 2002].

3 Products of sigmoids for switching

On the second level of our algorithm, the goal is to learn a predictive model $P(i|\mathbf{x})$ of the latent responsibility index i that is more precise than the uninformed prior $P_{\text{loc}}(i|\mathbf{x})$. Given some data, it is easy to decide whether two models are “potentially neighbored”—namely whether there exists data for which both models are eligible—based on their centers. For each pair (ij) of neighbored models, we learn a sigmoidal function $\psi_{ij}(\mathbf{x})$, where $\psi_{ij} \equiv 1 - \psi_{ji}$. The product of such sigmoids around a submodel i defines a coefficient $\beta_i(\mathbf{x})$ that we associate with the submodel for a given input $\mathbf{x}$,

$$\beta_i(\mathbf{x}) = \frac{1}{Z'} \prod_j \psi_{ij}(\mathbf{x}), \quad \psi_{ij}(\mathbf{x}) = \frac{1}{1 + \exp[-\phi_{ij}(\mathbf{x})]}.$$

Here, Z' normalizes β_i over i . As indicated, we represent sigmoids ψ_{ij} with a scalar function ϕ_{ij} . Fig. 1 illustrates the kind of kernels can be represented as products of sigmoids.

The sigmoids $\psi_{ij}(\mathbf{x})$ are meant to represent the likelihood that a model i rather than j is responsible for an input $\mathbf{x}$, conditioned on that either i or j is responsible. The product combination is comparable to an AND voting. The MAP labeling $\hat{i}$ we introduced in the previous section is now used to train these sigmoids. In the experiments, we consider ϕ_{ij} to be linear functions, again learned with PLS.

4 Experiments

We tested the algorithm on piecewise linear, discontinuous test functions. A test function has 3 parameters: the input dimension d , the number L of linear pieces it is composed of and the output noise σ . The localities, slopes and boundaries of the linear pieces are sampled randomly. Fig. 2(a,b) display learning results from a 1D example in comparison to LWPR. Fig. 2(c,d) display two error curves on 10-dimensional test functions over the rather large input domain $[-1, 1]^{10}$. The *family error* is the MSE of the best fitting eligible model $\phi_{\hat{i}}$ (averaged over an independent test data set); the *classification error* counts how often the product of sigmoids correctly predicts $\phi_{\hat{i}}$ to be the best fitting model for a given input (i.e., $\arg\max_i \beta_i = \hat{i}$). In the experiments we find that the algorithm reliably generates a family with optimal family error at the noise level ($\sigma^2 = 0.01$). In 5 dimensions (not displayed here) the classification error rapidly converges to zero while in 10 dimensions, the classification error converges to around 4%. For more results see homepages.inf.ed.ac.uk/mtoussai/projects/05-ijcai.

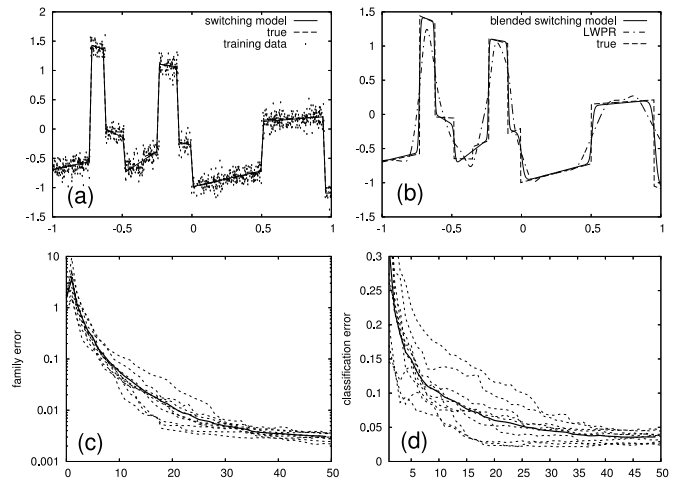


Figure 2: (a) A 1D test function with $d=1$, $L=10$, $\sigma=0.1$. Learned switching model after 20 iterations on $M=1000$ training data points. (b) The blended switching model: $y(\mathbf{x}) = \sum_i \beta_i(\mathbf{x}) \phi_i(\mathbf{x})$ compared to LWPR. (c) Family error (cf. Sec. 4) and (d) classification error for 10 runs on random test functions with $d=10$, $L=10$, $\sigma=0.1$, and $M=10\,000$. The bold line is the average over all curves.

5 Discussion

The presented model addresses the problem of handling the discontinuities that naturally arise, e.g., in sensorimotor data during interaction with a structured environment. Our model extends earlier local learning approaches in several ways: The responsibility region associated with each local model (learned with the product of sigmoids) has a much more versatile boundary shape compared to typical Gaussian kernels. Problems associated with initialization of kernel shapes or widths and the heuristic choice of an ad hoc number of submodels are circumvented by the robust incremental allocation of new models. Although we consistently used PLS as the underlying regression machinery, the general model allows to utilize any efficient single model learner to represent the local models ϕ_i as well as the classifier functions ϕ_{ij} . Future work will in particular investigate non-linear learners for the local models as well as the boundary classifiers.

References

- [Ghahramani and Hinton, 1998] Z. Ghahramani and G.E. Hinton. Variational learning for switching state-space models. *Neural Computation*, 12:963–996, 1998.
- [Pavlovic *et al.*, 2000] Vladimir Pavlovic, James M. Rehg, and John MacCormick. Learning switching linear models of human motion. In *NIPS*, pages 981–987, 2000.
- [Vijayakumar *et al.*, 2002] Sethu Vijayakumar, Aaron D’Souza, Tomohiro Shibata, Jorg Conradt, and Stefan Schaal. Statistical learning for humanoid robots. *Autonomous Robot*, 12:55–69, 2002.
- [Wolpert and Kawato, 1998] D.M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329, 1998.

POSTER PAPERS

UNCERTAINTY

Model minimization by linear PSR

Masoumeh T. Izadi

McGill University
School of Computer Science
mtabae@cs.mcgill.ca

Doina Precup

McGill University
School of Computer Science
dprecup@cs.mcgill.ca

Abstract

Predictive state representation (PSR), proposed by [Littman *et al.*, 2002; Singh *et al.*, 2004], are a general representation for controlled dynamical systems. We present a sufficient condition under which a linear PSR compresses a POMDP representation.

1 Introduction

Efficient decision making under uncertainty requires ignoring irrelevant details of a complex dynamical system and focusing on useful abstractions. A traditional representation for stochastic dynamical systems is provided by Partially Observable Markov Decision Processes (POMDPs). An attractive alternative to POMDPs is the Predictive State Representation (PSR), introduced in [Littman *et al.*, 2002] and further developed in [Singh *et al.*, 2004]. PSRs can be used to represent a larger class of dynamical systems than POMDPs. Even for systems that can be represented as POMDPs, PSRs hold the promise of a more compact representation. In particular, Littman *et al.* (2002) show that a PSR representation is no larger than the number of states in a POMDP. We point out a special case of their linear PSR representation, in which a strict reduction in the number of states is obtained. We believe this special case is interesting because it relies only on the state dynamics, without taking into account the observations. This can be potentially attractive, especially for robotics applications, in which states can have similar dynamics but sensor reading will often be different.

2 Preliminaries

A POMDP representation of a dynamical system includes the following components: a finite unobservable state space S ; a finite action space A ; a finite observation space O ; a transition function $T : S \times A \times S \rightarrow \mathcal{R}$, $T(s, a, s') = P(s_{t+1} = s' | s_t = s, a_t = a)$; an observation function $O : S \times A \times O \rightarrow \mathcal{R}$, $O(o, s, a) = P(o_{t+1} = o | s_{t+1} = s, a_t = a)$; an initial belief state b_0 , which is a vector of size $|S|$, giving the initial probability of the system being in each underlying state; and a reward function $R : S \times A \rightarrow \mathcal{R}$, where $R(s, a)$ is the immediate expected reward. We will not be concerned here with rewards, we will consider just actions and observations.

We denote by T^a an $|S| \times |S|$ matrix containing state transition probabilities for action a . We denote by O^{ao} a diagonal

matrix of size $|S| \times |S|$, in which the diagonal elements correspond to the probabilities of emitting o from each state, given that the state is reached by action a . A test is defined as an ordered sequence of action-observation pairs $t = a_0 o_1 \dots a_{n-1} o_n$, which can happen from the current time step into the future. The conditional probability of test t given a prior history h of action-observation pairs is $p(t|h) = p(o_1 \dots o_n | h, a_0 \dots a_{n-1})$.

A set of tests $Q = \{q_1 \dots q_m\}$ constitutes a linear PSR if, for any history, the probability of any other test t can be computed as a linear combination of the predictions for the tests in Q . In other words, for any test t there exists a vector m_t of size $|Q|$, such that $p(t|h) = p(Q|h)^T m_t, \forall h$.

Littman *et al.* also define an *outcome* function u mapping tests into n -dimensional vectors defined recursively by: $u(\epsilon) = 1_{|S|}$ and $u(aot) = (T^a O^{ao} u(t))$ where ϵ represents a null test and e_n is the $(1 \times n)$ vector of all 1s. Each component $u_i(t)$ indicates the probability of the test t when its sequence of actions is applied from state i . A set of tests Q is called linearly independent if the outcome vectors of its tests $u(q_1), u(q_2), \dots, u(q_m)$ are linearly independent.

PSRs are related to POMDPs through the state-test prediction matrix U [Littman *et al.*, 2002]. The rows of U correspond to states in S and columns correspond to all possible tests in order of increasing length. The entry U_{ij} is the conditional probability of the j th test given that the state of the system is i . A linear PSR can be derived from the matrix U by searching for a maximal set of linearly independent columns of U . Following the definition of outcome function,

$$U_{ij} = u(t_j | s_i) = (T^{a_0} O^{a_0 o_1} \dots T^{a_{m-1}} O^{a_{m-1} o_m})_i$$

where $t_j = a_0 o_1 \dots a_{n-1} o_n$. The maximum number of linearly independent columns is the rank of U . Therefore the size of Q is upper-bounded by the number of states. Next section presents a special case in which $|S|$ is strictly greater than $|Q|$.

3 Linearly dependent states

We say that a state i is linearly dependent on a subset of states $S' \subset S$ if and only if its transition probabilities under *any* action a are a linear combination of the transition probabilities of states in S' under the same action:

$$T_s^a = \sum_{k \in S'} c_k T_k^a \forall a \in A,$$

where $\sum_{k \in S'} c_k = 1$ and there exists k such that $c_k \neq 0$. Note that the coefficients c_k will be used to weigh the transitions

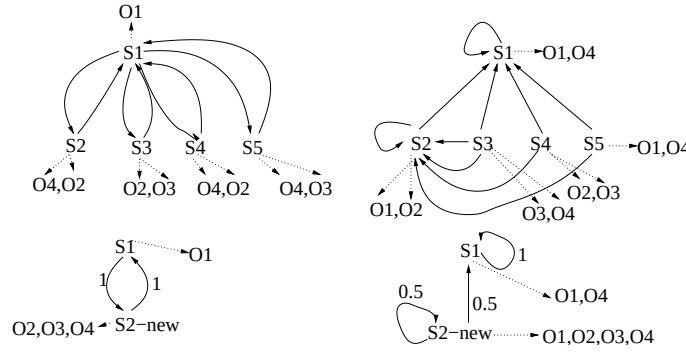


Figure 1: Example of a POMDP with linearly dependent states, and the model reduced by a linear PSR

corresponding to state k for all of the actions. Hence, these coefficients have to be shared across the actions.

Theorem: If the underlying MDP of a given POMDP has a linearly dependent states, then a linear PSR will provide a compression of the state space.

Proof: Suppose there exists a state $i \in S$ which is linearly dependent on a set of states S' , so:

$$\forall a \in A : T_i^a = \sum_k c_k T_k^a$$

To prove that the i th row of U is a linear combination of the other rows for all possible tests, we proceed by induction. Consider first one-step tests. Let O^a be a matrix of size $|S| \times |O|$, giving the probabilities of different observations being emitted from each state, after action a is taken. Then, by taking transposes and multiplying the above equation, we get:

$$(T_i^a)^T O^a = (\sum_k c_k T_k^a)^T O^a = \sum_k c_k (T_k^a)^T O^a$$

Note that this corresponds to the part of the i th row in the U matrix which contains the observations for all one-step tests for action a .

Now suppose that we have established for all tests t of length l that the outcome of t in state i can be written as a linear combination of the outcomes of states k :

$$u_i(t) = \sum_{k \in S'} c_k u_k(t)$$

Consider a test aot of length $l+1$. We have:

$$\begin{aligned} u_i(aot) &= T_i^a O^{ao} u(t) = (\sum_{k \in S'} c_k T_k^a) O^{ao} u(t) \\ &= \sum_{k \in S'} c_k (T_k^a O^{ao} u(t)) = \sum_{k \in S'} c_k u_k(aot) \end{aligned}$$

Hence, the i th row of the U matrix is a linear combination of the rows corresponding to the states in S' , with the same mixing coefficients as those from the transition matrix. Therefore, the rank of U is strictly less than $|S|$. Since the dimension of the linear PSR is given by the rank of U , in this case the linear PSR representation will be smaller than the size of the state space. $\diamond$

An example of a system with 5 states, 4 observations and two actions, in which a simple linear dependence can be seen, is provided in Figure 1. The model corresponding to action

a is in the upper left while the model for action b is in the upper right. The solid lines represent state transitions and the dashed lines represent emission probabilities. For action a , from state $S1$ the system transitions to one of the other states with equal probability. These states return deterministically to $S1$. Under action b , from all the bottom states the system transitions with probability 0.5 to state $S1$. The observations can be assumed to have all the same probability, although this does not really matter for the example. In this case, states $S3$, $S4$ and $S5$ are all linearly dependent on $S2$ (they have the same transition probabilities). However, note that they do not have the same observation models. The bottom row presents a simplified system, which has been reduced to two states by eliminating the linearly dependent states. The models for action a and b are in the left and right column. The emission probabilities are all equal when more than one observation is emitted from a state. In general, these probabilities would be computed by averaging the emission probabilities for the states in the original model.

Note that this theorem relates only linearly dependent state transitions to PSR compression without considering the observations. In fact, more compression can be obtained if the observations are taken into account. A good example of this sort is the POMDP coffee domain used in [Poupart and Boutilier, 2003]. This problem has 32 states, 2 actions and 3 observations. Four of these states are linearly dependent on the other ones, which means that the dimensionality can be reduced to 28. However, by running the linear PSR construction algorithm, a more significant reduction is possible: the problem can be represented with just two tests. The value-directed compression method of [Poupart and Boutilier, 2003] takes advantage of the same regularities as linear PSRs, but also of regularities in the reward function.

References

- [Littman *et al.*, 2002] Michael Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *NIPS 14*, pages 1555–1561, 2002.
- [Poupart and Boutilier, 2003] Pascal Poupart and Craig Boutilier. Value-directed compression of POMDPs. In *NIPS 15*, pages 1547–1554, 2003.
- [Singh *et al.*, 2004] Satinder Singh, Michael R. James, and Matthew R. Ruddary. Predictive state representations: A new theory for modeling dynamical systems. In *Proceedings of UAI*, pages 512–519, 2004.

Using core beliefs for point-based value iteration

Masoumeh T. Izadi

McGill University
School of Computer Science
mtabae@cs.mcgill.ca

Ajit V. Rajwade

University of Florida
CISE Department
avr@cise.ufl.edu

Doina Precup

McGill University
School of Computer Science
dprecup@cs.mcgill.ca

Abstract

Recent research on point-based approximation algorithms for POMDPs demonstrated that good solutions to POMDP problems can be obtained without considering the entire belief simplex. For instance, the Point Based Value Iteration (PBVI) algorithm [Pineau *et al.*, 2003] computes the value function only for a small set of belief states and iteratively adds more points to the set as needed. A key component of the algorithm is the strategy for selecting belief points, such that the space of reachable beliefs is well covered. This paper presents a new method for selecting an initial set of representative belief points, which relies on finding first the basis for the reachable belief simplex. Our approach has better worst-case performance than the original PBVI heuristic, and performs well in several standard POMDP tasks.

1 Introduction

Partially Observable Markov Decision Processes (POMDPs) provide a general framework for planning under uncertainty [Kaelbling *et al.*, 1998; Sondik, 1971]. One of the standard algorithms used to provide solutions to POMDPs is value iteration, which associates values to probability distributions over states. Because exact value iteration is intractable, a lot of recent work has focused on approximate algorithms, which rely on compressing the belief space or considering a finite set of reachable beliefs. A representative algorithm for this class is point-based value iteration (PBVI) [Pineau *et al.*, 2003]. In PBVI, a finite set of reachable belief points is selected heuristically, and values are computed only for these points. The success of PBVI depends crucially on the selection of the belief points. In particular, these points should cover the space of reachable beliefs as evenly as possible. The main PBVI heuristic uses one-step simulated trajectories in order to find reachable beliefs. The belief that is “farthest away” from the points already included is greedily added to the set. We explore an alternative heuristic for choosing belief points. We aim to find *core beliefs*, i.e. beliefs which form a basis for the space of all reachable beliefs. This approach is aimed at covering more quickly the space of beliefs that are reachable, by looking farther into the future. This improves the worst-case

behavior of the PBVI algorithm. We discuss how core beliefs can be found, and we illustrate the behavior of the algorithm on several standard POMDP benchmarks.

2 POMDPs and point-based value iteration

A POMDP is described by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, \Omega, R, \gamma, b_0 \rangle$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions and $\mathcal{O}$ is a finite set of observations. The probability of any state at $t = 0$, is described by the initial belief, b_0 . The transition function $T(s, a, s')$ describes the probability of going from state s to state s' given that action a is chosen. The observation function $\Omega(o, s, a)$ describes the probability of observation o given that action a was taken and state s was reached. The reward function, $R(s, a)$, specifies the expected immediate reward, obtained after executing action a in state s . The discount factor $\gamma \in (0, 1)$ is used to weigh less delayed rewards. The goal of an agent acting in a POMDP is to find a way of choosing actions (policy) maximizing the future return: $E[\sum_t \gamma^t R(s_t, a_t)]$. In order to achieve this goal, the agent must keep either a complete history of its actions and observations, or a sufficient statistic of the history. The sufficient statistic in a POMDP is the belief state, b , a vector with $|\mathcal{S}|$ components, which represents a probability distribution over states:

$$b_t(s) = P(s_t = s | b_0, a_0 o_1 \dots a_{t-1} o_t)$$

On each time step, after taking action a and making observation o , the agent updates its belief state using Bayes rule:

$$b(s') \leftarrow P(s' | a, o, b) = \frac{\Omega(o, s', a) \sum_s b(s) T(s, a, s')}{P(o | a, b)}, \quad (1)$$

where the denominator is just a normalizing constant.

In value-based POMDP methods, the agent computes a value function, which is maintained over belief states. A lot of recent research has been devoted to computing such values approximately. Compression-based methods [Roy and Gordon, 2003; Poupart and Boutilier, 2004] aim to compress the belief space, and maintain a value function only on this compressed version. Point-based value iteration (PBVI) [Pineau *et al.*, 2003] maintains values and gradients (α -vectors, in POMDP terminology) for a selected set of belief points, B . The idea is that much of the belief simplex will not be reachable in general. The set B is expanded over time, in order to cover more of the reachable belief space, and these expansions are interleaved with computing new value estimates.

3 Belief selection

In order to extend the belief set B , the standard PBVI heuristic considers, for each $b \in B$, all possible actions a and samples one observation o for each action. Among these beliefs reachable from b , the belief b' that is farthest away from B is picked and added to B . This heuristic is motivated by an analytical upper bound on the approximation error, which depends on the maximum L_1 distance from any reachable belief to B :

$$\varepsilon_B = \max_{b' \in \bar{\Delta}} \min_{b \in B} \|b - b'\|_1$$

where $\bar{\Delta}$ is the set of all reachable beliefs. PBVI does not keep an upper bound on the value function when expanding its belief set. However Smith and Simmons [Smith and Simmons, 2004] have developed an alternative anytime solution method, HSVI, based on maintaining an upper and a lower bound of the optimal value function to guide the local value updates. They choose the belief points to update using forward heuristic search. Vlassis and Spaan [Vlassis and Spaan, 2004] instead sample a large set of reachable beliefs B during a random walk, but then only update a subset of B , sufficient to improve values overall.

We propose an approach for covering the space of reachable beliefs more quickly, by considering longer courses of action, as well as by removing the strictly greedy character of the PBVI algorithm. We will use core beliefs, which form a basis for the belief simplex. The idea of core beliefs is inspired by work on predictive state representations (PSRs) [Littman *et al.*, 2002], an alternative way of representing stochastic dynamical systems. In order to reason about the space of reachable beliefs, one can consider the initial belief vector, b_0 , and all possible sequences of actions and observations following it. These sequences are called tests. James and Singh (2004) introduced the concept of an infinite matrix, whose rows correspond to all histories and whose columns correspond to all tests. In this paper we use the same infinite matrix but the rows correspond to all reachable beliefs from a given initial belief point. For a row b and a test $a_0 o_1, \dots, a_{n-1} o_n$, the content of the corresponding element in the matrix is the probability that $o_1, \dots, o_n$ is observed given that action sequence $a_0, \dots, a_{n-1}$ is executed starting from b . This matrix has finite rank, at most equal to the number of states in the POMDP. We define core beliefs as the set of linearly independent rows of this matrix.

The computation of core beliefs can be done exactly using the POMDP model, as shown in Algorithm 1. It is similar to the computation of core tests in [James and Singh, 2004], except that here we compute the elements of the matrix having the POMDP model while they estimate these elements from data without the model. But in large environments, considering all one-step extensions for all tests is prohibitive. We experimented with two heuristics for the extensions:

1. Random: pick a random action, but consider all observations.
2. ε -greedy: consider actions that are likely to be picked by a good policy. We generate first a subset of the core beliefs based on all one-step tests. Then, we run PBVI on this subset, for a fixed number of iterations, in order

Algorithm 1 Finding core beliefs

Initialize the matrix with b_0 and rows corresponding to all beliefs reachable by one-step tests a, o . One column is all 1's and the others correspond to one-step tests a, o .

for all elements (b, ao) **do**

 Compute $P(o|b, a)$, using (1) and store it in row b , column a, o

end for

repeat

 Compute a set of linearly independent rows and columns
 Add the linearly independent rows to the set of core beliefs

 Eliminate all linearly dependent rows and columns

 Add all one-step extensions to the matrix and compute the rank of the extended matrix

until The rank is unchanged

to get a first approximation of the value function. We extend the core belief matrix by picking one action, in ε -greedy fashion. The next observation can be sampled as well.

Note that for PBVI, the maximum error value for ε_B that can be achieved is 2, attained for the case in which two beliefs are non-zero over different states. This case cannot occur when all core beliefs are part of B . Hence, the worst-case error when using core beliefs is strictly smaller.

4 Experimental results

We performed experiments on several standard POMDP domains used in the literature. For the smaller three domains (4x4 grid, 4x3 grid and Cheese) we computed the core beliefs exactly (column “CBVI-all” of Table 1). The proposed heuristics are run using all possible observations in the one-step extensions. We also ran standard PBVI with a similar number of initial beliefs. In all cases we performed 5 iterations of PBVI to obtain a value function. Then, we ran 251 trials, with each trial cut at 251 steps. We averaged the results obtained over these trials, and over 5 independent runs.

All versions of the CBVI algorithm yield improvements over the PBVI heuristic. Surprisingly, in the 4x4 and Cheese domains, the best result is achieved by the ε -greedy heuristic. We conjecture that the reason is that the ε -greedy heuristic actually focuses the computation on good actions, that are likely to be used in the optimal policy.

In the larger domains, we only used the ε -greedy heuristic with sampled observations, as the other versions are too expensive. The experimental setup is as above. In this case, we find a first set of core beliefs, then run PBVI for 5 iterations. We take the resulting belief set, and add one-step extensions using ε -greedy actions and sampled observations, as above. This results in a new matrix, which yields a new set of core beliefs. This process is repeated at most 5 times; if no more core beliefs are detected in a new iteration, the process stops. The results are presented in the columns CBVI(1) to CBVI(5) in Table 2. Note that for the Coffee domain, all core beliefs are found after two such repetitions. Hence, there is no data for the CBVI(3)-CBVI(5). The reward for standard

Domain	PBVI	CBVI-all	CBVI-rand	CBVI-egr
4x4 grid	3.17 ± 0.22	3.80 ± 0.11	3.45 ± 0.22	3.96 ± 0.28
4x3 grid	1.58 ± 0.35	2.45 ± 0.44	2.10 ± 0.09	2.07 ± 0.11
Cheese	3.53 ± 0.13	3.53 ± 0.08	3.65 ± 0.22	3.81 ± 0.35

Table 1: Small domains

Domain	PBVI-orig	VS	PBVI	CBVI(1)	CBVI(2)	CBVI(3)	CBVI(4)	CBVI(5)
Maze33	2.25	2.34	2.248	2.185 ± 0.3	2.285 ± 0.31	2.165 ± 0.38	2.234 ± 0.25	2.301 ± 0.26
Hallway1	0.53	0.51	0.503	0.535 ± 0.03	0.617 ± 0.08	0.556 ± 0.03	0.574 ± 0.04	0.582 ± 0.08
Hallway2	0.35	0.31	0.379	0.333 ± 0.04	0.334 ± 0.04	0.392 ± 0.05	0.403 ± 0.04	0.400 ± 0.03
Coffee	-3.00	-	-2.759 ± 0.93	0.024 ± 2.14	0.58 ± 2.72	-	-	-

Table 2: Large domains

PBVI using a similar number of initial beliefs is in the column PBVI in Table 2. The results reported in the original PBVI paper (PBVI-orig)[Pineau *et al.*, 2003] and the results by Vlassis and Spaan (VS)[Vlassis and Spaan, 2004] are in the columns PBVI-orig and VS respectively. As shown in Table 2, for Maze33 all algorithms obtain very similar results. The Coffee domain is a very favorable case for CBVI, because it has a nice structure, and the reachable space of beliefs lies in a very low dimensional part of the entire belief simplex. Generating core beliefs can be done in 2-3 seconds for this domain, because there are only two core beliefs for this problem, although its POMDP representation has 32 states. Our experiments show a huge value improvement for this particular case over PBVI. For the larger domains, Hallway1 and Hallway2, CBVI also has a definite advantage, especially in the later iterations, when the initial subset of core beliefs is larger. The time complexity of CBVI is the same as PBVI if generating the core beliefs can be done in a preprocessing phase. In small domains and large domains with small intrinsic dimensionality (e.g. Coffee) this can be done in a few seconds. However, generating all core beliefs in large domains in general is very expensive. In our experiments, we reached the maximum number of core beliefs in at most five extensions of the matrix in algorithm 1; however, the computation time, taking into account the process of belief generation is two orders of magnitude longer than for PBVI. Although this approach improves the value function and the policy that is obtained, it is significantly more expensive, at the moment, unless the problem at hand has special structure. We are investigating different ways to compute a reasonably large portion of the space of core beliefs at a lower computation cost.

Acknowledgments

This research was supported in part by funding from NSERC and CFI. We thank Joelle Pineau for very helpful discussions and insights into point-based algorithms, as well as for constructive feedback. We thank Matthijs Spaan for providing code for his point-based value iteration algorithm.

References

- [James and Singh, 2004] Michael James and Satinder Singh. Predictive State Representations: A New Theory for Modeling Dynamical Systems. *ICML 21*, pages 417-424, 2004.
- [Kaelbling *et al.*, 1998] Leslie P. Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.
- [Littman *et al.*, 2002] Michael Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *NIPS 14*, pages 1555–1561, 2002.
- [Pineau *et al.*, 2003] Joelle Pineau, Geoff Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithms for POMDPs. In *Proceedings of IJCAI*, pages 1025–1032, 2003.
- [Poupart and Boutilier, 2004] Pascal Poupart and Craig Boutilier. VDCBPI: an approximate scalable algorithm for large scale POMDPs. In *NIPS 17*, 2004.
- [Roy and Gordon, 2003] Nicholas Roy and Geoffrey Gordon. Exponential family PCA for belief compression in POMDPs. In *NIPS 15*, pages 1635–1642, 2003.
- [Smith and Simmons, 2004] Tery Smith and Reid Simmons. Heuristic Search Value Iteration for POMDPs. In *UAI 20*, pages 520-527, 2004.
- [Sondik, 1971] Edward J. Sondik. *The optimal control of Partially Observable Markov Decision Processes*. PhD Thesis, Stanford University, 1971.
- [Vlassis and Spaan, 2004] Nikos Vlassis and Matthijs Spaan. A point-based POMDP algorithm for robot planning. In *Proceedings of ICRA*, 2004.

Approximating Pseudo-Boolean Functions on Non-Uniform Domains*

R. F. Lax¹, Guoli Ding¹, Peter P. Chen², and J. Chen²

¹ Department of Mathematics, LSU, Baton Rouge, LA 70803

² Department of Computer Science, 298 Coates Hall, LSU, Baton Rouge, LA 70803
lax@math.lsu.edu, ding@math.lsu.edu, chen@bit.csc.lsu.edu, jianhua@bit.csc.lsu.edu,

Abstract

In Machine Learning (ML) and Evolutionary Computation (EC), it is often beneficial to approximate a complicated function by a simpler one, such as a linear or quadratic function, for computational efficiency or feasibility reasons (cf. [Jin, 2005]). A complicated function (the target function in ML or the fitness function in EC) may require an exponential amount of computation to learn/evaluate, and thus approximations by simpler functions are needed. We consider the problem of approximating pseudo-Boolean functions by simpler (e.g., linear) functions when the instance space is associated with a probability distribution. We consider $\{0, 1\}^n$ as a sample space with a (possibly non-uniform) probability measure on it, thus making pseudo-Boolean functions into random variables. This is also in the spirit of the PAC learning framework of Valiant [Valiant, 1984] where the instance space has a probability distribution on it. The best approximation to a target function f is then defined as the function g (from all possible approximating functions of the simpler form) that minimizes the expected distance to f . In an example, we use methods from linear algebra to find, in this more general setting, the best approximation to a given pseudo-Boolean function by a linear function.

1 Introduction

A pseudo-Boolean function of n variables is a function from $\{0, 1\}^n$ to the real numbers. Such functions are used in 0-1 optimization problems, cooperative game theory, multicriteria decision making, and as fitness functions. It is not hard to see that such a function $f(x_1, \dots, x_n)$ has a unique expression as a multilinear polynomial

$$f(x_1, \dots, x_n) = \sum_{T \subseteq N} \left[a_T \prod_{i \in T} x_i \right],$$

*Research partially supported by NSF grant ITR-0326387 and AFOSR grants F49620-03-1-0238, F49620-03-1-0239, and F49620-03-1-0241

where $N = \{1, \dots, n\}$ and the a_T are real numbers. By the *degree* of a pseudo-Boolean function, we mean the degree of its multilinear polynomial representation.

Several authors have considered the problem of finding the best pseudo-Boolean function of degree $\leq k$ approximating a given pseudo-Boolean function f , where “best” means a least squares criterion. Hammer and Holzman [Hammer and Holzman, 1992] derived a system of equations for finding such a best degree $\leq k$ approximation, and gave explicit solutions when $k = 1$ and $k = 2$. They proved that such an approximation is characterized as the unique function of degree $\leq k$ that agrees with f in all average m -th order derivatives for $m = 0, 1, \dots, k$, in analogy with the Taylor polynomials from calculus. Grabisch, Marichal, and Roubens [Grabisch *et al.*, 2000] solve the system of equations derived by Hammer and Holzman, and give explicit formulas for the coefficients of the best degree $\leq k$ function. Zhang and Rowe [Zhang and Rowe, 2004] use linear algebra to find the best approximation that lies in a linear subspace of the space of pseudo-Boolean functions; for example, these methods can be used to find the best approximation of degree $\leq k$.

Here, instead of simply viewing the domain of a pseudo-Boolean function as the set $\{0, 1\}^n$, we consider $\{0, 1\}^n$ as a discrete sample space and introduce a probability measure on this space. Thus, a pseudo-Boolean function will be a random variable on this sample space. (Viewing $\{0, 1\}$ simply as a set corresponds to viewing all of its points as equally likely outcomes.) Given a pseudo-Boolean random variable f , we then use methods from linear algebra to find the best approximation to f that lies in a linear subspace, taking into account the weighting of the elements of $\{0, 1\}^n$. Such a best approximation will then be close to f at the “most likely” n -tuples, and may not be so close to f at the “least likely” n -tuples.

2 Best Approximation on a Non-Uniform Domain

We will identify the integers $0, 1, \dots, 2^n - 1$ with the elements in B^n via binary representation. Let $p(i), i = 0, 1, \dots, 2^n - 1$, be a probability measure on B^n . Let $\mathcal{F}$ denote the space of all pseudo-Boolean functions in n variables. Then $\mathcal{F}$ has the structure of a real vector space. Define

an inner product $\langle \cdot, \cdot \rangle_p$ on $\mathcal{F}$ by

$$\langle f, g \rangle_p = \sum_{i=0}^{2^n-1} f(i)g(i)p(i).$$

We note that $\langle f, g \rangle_p$ is the expected value of the random variable fg . Put $\|f\|_p = \sqrt{\langle f, f \rangle_p}$.

Now let $\mathcal{L}$ be a vector subspace of $\mathcal{F}$ of dimension m . For example, $\mathcal{L}$ might be the space of all pseudo-Boolean functions of degree at most k , for some fixed k . We recall how to use an orthonormal basis of $\mathcal{L}$ to find the best approximation to a given element of $\mathcal{F}$ (cf. [Hoffman and Kunze, 1971]).

Let $v_1, \dots, v_m$ be a basis for $\mathcal{L}$. We can find an orthonormal basis $u_1, \dots, u_m$ for $\mathcal{L}$ by applying the Gram-Schmidt algorithm. This orthonormal basis satisfies the property $\langle u_r, u_s \rangle_p = \delta_{rs}$ for $r, s = 1, \dots, m$, where δ_{rs} equals 0 if $r \neq s$ and equals 1 if $r = s$. The orthonormal basis can be obtained as follows: Take $u_1 = (1/\|v_1\|_p)v_1$. If $u_1, \dots, u_{r-1}$ have been obtained, then put $w_r = v_r - \sum_{j=1}^{r-1} \langle v_r, u_j \rangle_p u_j$, and take $u_r = (1/\|w_r\|_p)w_r$.

Given $f \in \mathcal{F}$, the “best approximation” to f by functions in $\mathcal{L}$ is that function $g \in \mathcal{L}$ that minimizes

$$\|f - g\|_p = \sqrt{\sum_{i=0}^{2^n-1} (f(i) - g(i))^2 p(i)}.$$

Notice that if we take the uniform distribution on B^n , so that $p(i) = (1/2)^n$ for all i , then the best approximation to f in $\mathcal{L}$ is the function $g \in \mathcal{L}$ that also minimizes $\sum_{i=0}^{2^n-1} (f(i) - g(i))^2$. This is the usual “least squares” condition used in [Hammer and Holzman, 1992], [Grabisch *et al.*, 2000], [Zhang and Rowe, 2004], and in this case one may simply use the usual Euclidean inner product in $\mathbb{R}^{2^n}$. In our more general setting, it follows from section 8.2 of [Hoffman and Kunze, 1971] that the best approximation to f by functions in $\mathcal{L}$ is the unique function $g = \sum_{j=1}^m \langle f, u_j \rangle_p u_j$.

3 Example

To illustrate these ideas, we look at an example considered by [Zhang and Rowe, 2004]. Take $n = 3$ and $f(x_1, x_2, x_3) = 5x_1 + 13x_3 + 9x_1x_2 - 4x_1x_3 - 4x_2x_3 + 4x_1x_2x_3$. We wish to approximate f by the best linear function, so we let $\mathcal{L}$ be the space spanned by the functions $v_1 = 1, v_2 = x_1, v_3 = x_2, v_4 = x_3$. If we take the uniform distribution on B^3 , so that $p(i) = 1/8$ for $i = 0, 1, \dots, 7$, then by applying the Gram-Schmidt algorithm we get the following orthonormal basis for $\mathcal{L}$ with respect to the inner product $\langle \cdot, \cdot \rangle_p$:

$$u_1 = 1, u_2 = 2x_1 - 1, u_3 = 2x_2 - 1, u_4 = 2x_3 - 1.$$

(More generally, one can show that, for any n , an orthonormal basis for the space of pseudo-Boolean functions of degree at most 1 with respect to the uniform distribution is $1, 2x_1 - 1, \dots, 2x_n - 1$.) Then the best linear approximation to f is $g(x_1, x_2, x_3) = \sum_{j=1}^4 \langle f, u_j \rangle_p u_j =$

$$\begin{aligned} &= \frac{39}{4} \cdot 1 + \frac{17}{4}(2x_1 - 1) + \frac{7}{4}(2x_2 - 1) + 5(2x_3 - 1) \\ &= -\frac{5}{4} + \frac{17}{2}x_1 + \frac{7}{2}x_2 + 10x_3, \end{aligned}$$

in agreement with Example 4.1 of [Zhang and Rowe, 2004]. Here, $\|f - g\|_p \approx 2.88$.

Now we take a different probability measure on B^3 . Supposing that a “1” is twice as likely as a “0” we define a probability measure $\tilde{p}$ on B^3 by $\tilde{p}(0) = 1/27, \tilde{p}(1) = 2/27, \tilde{p}(2) = 2/27, \tilde{p}(3) = 4/27, \tilde{p}(4) = 2/27, \tilde{p}(5) = 4/27, \tilde{p}(6) = 4/27, \tilde{p}(7) = 8/27$. An orthonormal basis for $\mathcal{L}$ with respect to the inner product $\langle \cdot, \cdot \rangle_{\tilde{p}}$ is then

$$\tilde{u}_1 = 1, \tilde{u}_2 = \frac{3x_1 - 2}{\sqrt{2}}, \tilde{u}_3 = \frac{3x_2 - 2}{\sqrt{2}}, \tilde{u}_4 = \frac{3x_3 - 2}{\sqrt{2}}.$$

Then the best linear approximation to f is now $\tilde{g}(x_1, x_2, x_3) = \sum_{j=1}^4 \langle f, \tilde{u}_j \rangle_{\tilde{p}} \tilde{u}_j =$

$$\begin{aligned} &= \frac{368}{27} \cdot 1 + \frac{91\sqrt{2}}{27}\tilde{u}_2 + \frac{46\sqrt{2}}{27}\tilde{u}_3 + \frac{85\sqrt{2}}{27}\tilde{u}_4 \\ &= (1/27)(-76 + 273x_1 + 138x_2 + 255x_3). \end{aligned}$$

Here, $\|f - \tilde{g}\|_{\tilde{p}} \approx 2.55$. For comparison, the distance now between the linear function g we found above and the function f is $\|f - g\|_{\tilde{p}} \approx 2.79$.

4 Conclusion

Instead of considering $B^n = \{0, 1\}^n$ simply as a set, we allow it to be viewed as a sample space with a probability measure p . Then pseudo-Boolean functions are random variables on this sample space. Given a complicated pseudo-Boolean function, it is natural to want to approximate it by a simpler function, for example a linear or quadratic function. As an example, we found the best linear approximation to a given pseudo-Boolean function in three variables with respect to two different probability measures on B^3 . Further research is needed to find an effective method of computing the best approximation on a non-uniform domain when the number of variables is large.

References

- [Grabisch *et al.*, 2000] Michel Grabisch, Jean-Luc Marichal, and Marc Roubens. Equivalent representations of set functions. *Mathematics of Operations Research*, 25(2):157–178, May 2000.
- [Hammer and Holzman, 1992] P. L. Hammer and R. Holzman. Approximations of pseudo-boolean functions; applications to game theory. *ZOR-Methods and Models of Operations Research*, 36:3–21, 1992.
- [Hoffman and Kunze, 1971] Kenneth Hoffman and Ray Kunze. *Linear Algebra, 2nd edition*. Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
- [Jin, 2005] Yaochu Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005.
- [Valiant, 1984] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- [Zhang and Rowe, 2004] Hang Zhang and Jonathan E. Rowe. Best approximations of fitness functions of binary strings. *Natural Computing*, 3(1):113–124, 2004.

A Modal Logic for Reasoning about Possibilistic Belief Fusion

Churn-Jung Liao

Institute of Information Science

Academia Sinica

Taipei 115, Taiwan

liaucj@iis.sinica.edu.tw

Tuan-Fang Fan

Institute of Information Management

National Chiao-Tung University

Hsinchu 300, Taiwan

tffan.iim92g@nctu.edu.tw

1 Introduction

In this paper, we propose a modal logic for reasoning about possibilistic belief fusion. This is a combination of multi-agent epistemic logic and possibilistic logic. We use graded epistemic operators to represent agents' uncertain beliefs, and the operators are interpreted in accordance with possibilistic semantics. We employ ordered fusion based on a level skipping strategy to resolve the inconsistency caused by direct fusion; that is, the level at which the inconsistency occurs is skipped. Here, we present the formal semantics and an axiomatic system for the logic.

2 Syntax

To encode the degrees of reliability of n agents, we use ordering relations over any subset of $\{1, \dots, n\}$. Let $\mathcal{TO}_n$ denote the set of all possible strict total orders over any non-empty subset of $\{1, \dots, n\}$. Then, we can associate a unique syntactic notation with each strict total order in $\mathcal{TO}_n$. Let $X = \{i_1, i_2, \dots, i_m\}$ be a non-empty subset of $\{1, \dots, n\}$ and $>$ be a strict total order such that $i_j > i_k$ iff $j < k$ for all $1 \leq j, k \leq m$. Then, the syntactic notation for $(X, >)$ is the string $i_1 > i_2 > \dots > i_m$.

In this paper, the capital letter O is used to denote meta-variables ranging over such notations. Let O be the string $i_1 > i_2 > \dots > i_m$, then the set $\{i_1, i_2, \dots, i_m\}$ is called the domain of O , denoted by $\delta(O)$. In this case, $O > i_{m+1}$ denotes $i_1 > i_2 > \dots > i_m > i_{m+1}$ if $i_{m+1} \notin \delta(O)$. As the syntactic notation is unique for each total order, we can also identify the notation with the total order itself, and write $O \in \mathcal{TO}_n$. Furthermore, the upper-case Greek letter Ω is used to denote meta-variables ranging over nonempty subsets of $\mathcal{TO}_n$.

We now present the syntax of our logic for reasoning about possibilistic belief fusion based on a level skipping strategy. The logic is called $\text{SFPL}_n^{\otimes, \epsilon}$, where $\epsilon \in [0, 1]$ is the inconsistency tolerance degree of the logic and $\otimes$ is a continuous T -norm¹. Let Φ_0 be a set of propositional symbols. Then the set of well-formed formulas (wff) of $\text{SFPL}_n^{\otimes, \epsilon}$ is defined as the smallest set containing $\Phi_0 \cup \{\perp, \top\}$, and is closed under Boolean operators and the following rule:

- if φ is a wff, then $[\Omega]_a \varphi$ and $[\Omega]_a^+ \varphi$ are wffs for any nonempty $\Omega \subseteq \mathcal{TO}_n$, and any rational number $a \in [0, 1]$.

If $\Omega = \{O\}$ is a singleton, we write $[O]_a \varphi$ (resp. $[O]_a^+ \varphi$), instead of $[\{O\}]_a \varphi$ (resp. $[\{O\}]_a^+ \varphi$). Intuitively, $[O]_a \varphi$ (resp. $[O]_a^+ \varphi$) means that an agent merging distributed beliefs in accordance with the ordering of O will believe φ with a strength at least equal to (resp. more than) a . We can view each O as a virtual agent, and $[\Omega]$ corresponds to a distributed belief operator [Fagin *et al.*, 1996] of virtual agents in Ω . Note that our purpose is to ensure the consistency of ordered fusion based on any single order O . When Ω contains more than one order, $[\Omega]_e^+ \perp$ may hold. This does not matter, however, since a non-singleton Ω plays only an auxiliary role in our logic.

3 Semantics

To present the semantics, we briefly review possibility theory [Zadeh, 1978]. In this theory, each possibility distribution $\pi : W \rightarrow [0, 1]$ can derive an associated possibility measure $\Pi : 2^W \rightarrow [0, 1]$ and a necessity measure $N : 2^W \rightarrow [0, 1]$, as $\Pi(X) = \sup_{x \in X} \pi(x)$ and $N(X) = 1 - \sup_{x \notin X} \pi(x)$. A possibility distribution $\pi : W \rightarrow [0, 1]$ is normalized if $\Pi(W) = 1$. A normalized possibility distribution represents a consistent belief state. If π is not normalized, i.e., $\sup_{x \in W} \pi(x) < 1$, it represents a partially inconsistent belief state. $1 - \sup_{x \in W} \pi(x)$ is called the *inconsistency degree* of π , and is denoted by $\iota(\pi)$.

An $\text{SFPL}_n^{\otimes, \epsilon}$ -structure is a tuple $M = (W, (\pi_i)_{1 \leq i \leq n}, V)$, where W is a set of possible worlds, each π_i maps each world w to a possibility distribution $\pi_{i,w} : W \rightarrow [0, 1]$ over W such that $\iota(\pi_{i,w}) \leq \epsilon$, and V maps elements in Φ_0 to subsets of W . We define $\pi_{O,w}$ for each $O \in \mathcal{TO}_n$ and $w \in W$ inductively as follows:

$$\pi_{O>i,w} = \begin{cases} \pi_{O,w} & \text{if } \iota(\pi_{O,w} \otimes \pi_{i,w}) > \epsilon, \\ \pi_{O,w} \otimes \pi_{i,w} & \text{otherwise.} \end{cases}$$

Furthermore, we also define $\pi_{\Omega,w}$ for each $\Omega \subseteq \mathcal{TO}_n$ and $w \in W$, as $\pi_{\Omega,w} = \bigotimes_{O \in \Omega} \pi_{O,w}$. Then, the satisfaction relation $\models$ for $\text{SFPL}_n^{\otimes, \epsilon}$ -model is defined as

- $w \models p$ iff $w \in V(p)$, for each $p \in \Phi_0$,
- $w \not\models \perp$ and $w \models \top$,
- $w \models \neg \varphi$ iff $w \not\models \varphi$,

¹A T-norm is any binary operation on $[0,1]$ that is commutative, associative, non-decreasing in each argument, and has 1 as unit.

- $w \models \varphi \vee \psi$ iff $w \models \varphi$ or $w \models \psi$,
- $w \models [\Omega]_a \varphi$ iff $N_{\Omega,w}(|\varphi|) \geq a$,
- $w \models [\Omega]_a^+ \varphi$ iff $N_{\Omega,w}(|\varphi|) > a$,

where $|\varphi|$ is the truth set of φ in the model and $N_{\Omega,w}$ is the necessity measure associated with $\pi_{\Omega,w}$.

A set of wffs Σ is satisfied in a world w , written as $w \models \Sigma$, if $w \models \varphi$ for all $\varphi \in \Sigma$. We write $\Sigma \models_M \varphi$ if, for each possible world w in M , $w \models \Sigma$ implies $w \models \varphi$; and $\Sigma \models_{\text{SFPL}_n^{\otimes, \epsilon}} \varphi$ if $\Sigma \models_M \varphi$ for each SFPL $^{\otimes, \epsilon}_n$ -structure M . Σ can be omitted when it is an empty set, so a wff φ is valid in M , denoted by $\models_M \varphi$, if $\emptyset \models_M \varphi$, and $\models_{\text{SFPL}_n^{\otimes, \epsilon}} \varphi$ denotes $\emptyset \models_{\text{SFPL}_n^{\otimes, \epsilon}} \varphi$. The subscript is usually omitted if it is clear from the context.

• Axioms:

1. P: all tautologies of the propositional calculus

2. Bookkeeping:

- (a) $[\Omega]_c \varphi \supset [\Omega]_d^+ \varphi$ if $c > d$
- (b) $[\Omega]_c^+ \varphi \supset [\Omega]_c \varphi$
- (c) $[\Omega]_0 \varphi$
- (d) $\neg[\Omega]_1^+ \varphi$

3. V1:

- (a) $([\Omega]_a \varphi \wedge [\Omega]_a(\varphi \supset \psi)) \supset [\Omega]_a \psi$
- (b) $([\Omega]_a^+ \varphi \wedge [\Omega]_a^+(\varphi \supset \psi)) \supset [\Omega]_a^+ \psi$

4. V2: $\neg[i]_\epsilon^+ \perp$

5. V3: if $\Omega_1 \cap \Omega_2 = \emptyset$, then

- (a) $([\Omega_1]_a \varphi \wedge [\Omega_2]_b \varphi) \supset [\Omega_1 \cup \Omega_2]_{a \oplus b} \varphi$
- (b) $([\Omega_1]_a^+ \varphi \wedge [\Omega_2]_b^+ \varphi) \supset [\Omega_1 \cup \Omega_2]_{a \oplus b}^+ \varphi$

6. O1:

- (a) $\neg[\{O, i\}]_\epsilon^+ \perp \supset ([\Omega \cup \{O > i\}]_a \varphi \equiv [\Omega \cup \{O, i\}]_a \varphi)$
- (b) $\neg[\{O, i\}]_\epsilon^+ \perp \supset ([\Omega \cup \{O > i\}]_a^+ \varphi \equiv [\Omega \cup \{O, i\}]_a^+ \varphi)$

7. O2:

- (a) $[\{O, i\}]_\epsilon^+ \perp \supset ([\Omega \cup \{O > i\}]_a \varphi \equiv [\Omega \cup \{O\}]_a \varphi)$
- (b) $[\{O, i\}]_\epsilon^+ \perp \supset ([\Omega \cup \{O > i\}]_a^+ \varphi \equiv [\Omega \cup \{O\}]_a^+ \varphi)$

• Rules of Inference:

1. R1 (Modus ponens, MP):

$$\frac{\varphi \quad \varphi \supset \psi}{\psi}$$

2. R2 (Generalization, Gen):

$$\frac{\varphi}{[\Omega]_1 \varphi}$$

4 Proof Theory

An axiomatic system can be developed for SFPL $^{\otimes, \epsilon}_n$ by generalizing the corresponding axioms of DBF s_n [Liau, 2005] and PL $^{\otimes}_n$ [Boldrin and Saffiotti, 1999]. The axiomatic system for SFPL $^{\otimes, \epsilon}_n$ is presented in Figure 1. The symbol $\oplus$ in axiom V3 denotes the T-conorm corresponding to $\otimes$, which is defined as $a \oplus b = 1 - (1 - a) \otimes (1 - b)$. The axiom V2 is the requirement that the inconsistency degree of each agent's belief state is not more than ϵ , which is the inconsistency tolerance degree of the logic. The axioms O1 and O2 further enforce the same property for each virtual agent O .

A wff φ is derivable from the system SFPL $^{\otimes, \epsilon}_n$, or simply, φ is a *theorem* of SFPL $^{\otimes, \epsilon}_n$, if there is a finite sequence $\varphi_1, \dots, \varphi_m$ such that $\varphi = \varphi_m$ and every φ_i is an instance of an axiom schema; or it is obtained from earlier φ_j 's by the application of an inference rule. It is written as $\vdash_{\text{SFPL}_n^{\otimes, \epsilon}} \varphi$ if φ is a theorem of SFPL $^{\otimes, \epsilon}_n$. Let $\Sigma \cup \{\varphi\}$ be a subset of wffs, then φ is derivable from Σ in the system SFPL $^{\otimes, \epsilon}_n$, written as $\Sigma \vdash_{\text{SFPL}_n^{\otimes, \epsilon}} \varphi$, if there is a finite subset Σ' of Σ such that $\vdash_{\text{SFPL}_n^{\otimes, \epsilon}} \bigwedge \Sigma' \supset \varphi$. We drop the subscript when no confusion occurs. We then have the soundness and completeness theorem for SFPL $^{\otimes, \epsilon}_n$.

Theorem 1 For any wff φ of SFPL $^{\otimes, \epsilon}_n$, $\models \varphi$ iff $\vdash \varphi$.

5 Concluding Remarks

In this paper, we present a modal logic for reasoning about ordered fusion of possibilistic beliefs based on a level skipping strategy. While direct fusion and ordered fusion in epistemic logic [Fagin *et al.*, 1996; Cholvy, 1994; Liau, 2005], as well as direct fusion in possibilistic logic [Boldrin and Saffiotti, 1999] have been proposed previously in the literature, the results in this paper fill a gap in the previous works. The modal logic should be applicable to reasoning in multi-agent systems. In future work, it should be possible to consider operations other than T-norms for the fusion of possibility distributions.

References

- [Boldrin and Saffiotti, 1999] L. Boldrin and A. Saffiotti. A modal logic for merging partial belief of multiple reasoners. *Journal of Logic and Computation*, 9(1):81–103, 1999.
- [Cholvy, 1994] L. Cholvy. A logical approach to multi-sources reasoning. In M. Masuch and L. Pólos, editors, *Knowledge Representation and Reasoning under Uncertainty*, LNCS 808, pages 183–196. Springer-Verlag, 1994.
- [Fagin *et al.*, 1996] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1996.
- [Liau, 2005] C.J. Liau. A modal logic framework for multi-agent belief fusion. *ACM Transactions on Computational Logic*, 6(1):124–174, 2005.
- [Zadeh, 1978] L.A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3–28, 1978.

Figure 1: The axiomatic system for SFPL $^{\otimes, \epsilon}_n$

Networked Distributed POMDPs: A Synergy of Distributed Constraint Optimization and POMDPs

Ranjit Nair Pradeep Varakantham, Milind Tambe Makoto Yokoo

Knowledge Services Group
Honeywell Laboratories
ranjit.nair@honeywell.com

Computer Science Dept.
University of Southern California
{varakant,tambe}@usc.edu

Dept. of Intelligent Systems
Kyushu University
yokoo@is.kyushu-u.ac.jp

Abstract

In many real-world multiagent applications such as distributed sensor nets, a network of agents is formed based on each agent's limited interactions with a small number of neighbors. While distributed POMDPs capture the real-world uncertainty in multiagent domains, they fail to exploit such locality of interaction. Distributed constraint optimization (DCOP) captures the locality of interaction but fails to capture planning under uncertainty. This paper presents a new model synthesized from distributed POMDPs and DCOPs, called Networked Distributed POMDPs (ND-POMDPs). Exploiting network structure enables us to present a distributed policy generation algorithm that performs local search.

1 Introduction

In many real-world multiagent applications such as distributed sensor nets, a network of agents is formed based on each agent's limited interactions with a small number of neighbors. For instance, in distributed sensor nets, multiple sensor agents must coordinate with their neighboring agents in order to track individual targets moving through an area. In particular, we consider in this paper a problem motivated by the real-world challenge in [Lesser *et al.*, 2003]. Here, each sensor node can scan in one of four directions — North, South, East or West (see Figure 1), and to track a target, two sensors with overlapping scanning areas must coordinate by scanning the same area simultaneously. We assume that there are two independent targets and that each target's movement is uncertain but unaffected by the actions of the sensor agents. Additionally, each sensor receives observations only from the area it is scanning and this observation can have both false positives and false negatives. Each agent pays a cost for scanning whether the target is present or not but no cost if turned off. Existing distributed POMDP algorithms [Montemerlo *et al.*, 2004; Nair *et al.*, 2003; Becker *et al.*, 2004; Hansen *et al.*, 2004] although rich enough to capture the uncertainties in this

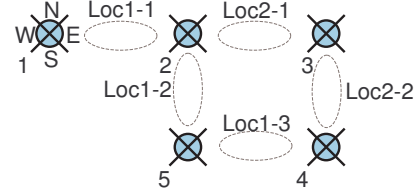


Figure 1: Sensor net scenario: If present, target1 is in Loc1-1, Loc1-2 or Loc1-3, and target2 is in Loc2-1 or Loc2-2.

domain, are unlikely to work well for such a domain because they are not geared to take advantage of the locality of interaction. As a result they will have to consider all possible action choices of even non-interacting agents, in trying to solve the distributed POMDP. Distributed constraint satisfaction and distributed constraint optimization (DCOP) have been applied to sensor nets but these approaches cannot capture the uncertainty in the domain. Hence, we introduce the networked distributed POMDP (ND-POMDP) model, a hybrid of POMDP and DCOP, that can handle the uncertainties in the domain as well as take advantage of locality of interaction. Exploiting network structure enables us to present a novel algorithm for ND-POMDPs – a distributed policy generation algorithm that performs local search.

2 ND-POMDPs

We define an ND-POMDP for a group Ag of n agents as a tuple $\langle S, A, P, \Omega, O, R, b \rangle$, where $S = \times_{1 \leq i \leq n} S_i \times S_u$ is the set of world states. S_i refers to the set of local states of agent i and S_u is the set of unaffected states. Unaffected state refers to that part of the world state that cannot be affected by the agents' actions, e.g. environmental factors like target locations that no agent can control. $A = \times_{1 \leq i \leq n} A_i$ is the set of joint actions, where A_i is the set of action for agent i .

We assume a *transition independent* distributed POMDP model, where the transition function is defined as $P(s, a, s') = P_u(s_u, s'_u) \cdot \prod_{1 \leq i \leq n} P_i(s_i, s'_i, a_i, s'_i)$, where $a = \langle a_1, \dots, a_n \rangle$ is the joint action performed in state $s = \langle s_1, \dots, s_n, s_u \rangle$ and $s' = \langle s'_1, \dots, s'_n, s'_u \rangle$ is the re-

sulting state. Agent i 's transition function is defined as $P_i(s_i, s_u, a_i, s'_i) = \Pr(s'_i | s_i, s_u, a_i)$ and the unaffected transition function is defined as $P_u(s_u, s'_u) = \Pr(s'_u | s_u)$. Becker *et al.* [2004] also relied on transition independence, and Goldman and Zilberstein [2004] introduced the possibility of uncontrollable state features. In both works, the authors assumed that the state is *collectively observable*, an assumption that does not hold for our domains of interest.

$\Omega = \times_{1 \leq i \leq n} \Omega_i$ is the set of joint observations where Ω_i is the set of observations for agents i . We make an assumption of *observational independence*, i.e., we define the joint observation function as $O(s, a, \omega) = \prod_{1 \leq i \leq n} O_i(s_i, s_u, a_i, \omega_i)$, where $s = \langle s_1, \dots, s_n, s_u \rangle$, $a = \langle a_1, \dots, a_n \rangle$, $\omega = \langle \omega_1, \dots, \omega_n \rangle$, and $O_i(s_i, s_u, a_i, \omega_i) = \Pr(\omega_i | s_1, s_u, a_i)$.

The reward function, R , is defined as $R(s, a) = \sum_l R_l(s_{l1}, \dots, s_{lk}, s_u, \langle a_{l1}, \dots, a_{lk} \rangle)$, where each l could refer to any sub-group of agents and $k = |l|$. In the sensor grid example, the reward function is expressed as the sum of rewards between sensor agents that have overlapping areas ($k = 2$) and the reward functions for an individual agent's cost for sensing ($k = 1$). Based on the reward function, we construct an *interaction hypergraph* where a hyper-link, l , exists between a subset of agents for all R_l that comprise R . *Interaction hypergraph* is defined as $G = (Ag, E)$, where the agents, Ag , are the vertices and $E = \{l | l \subseteq Ag \wedge R_l \text{ is a component of } R\}$ are the edges. *Neighborhood* of i is defined as $N_i = \{j \in Ag | j \neq i \wedge (\exists l \in E, i \in l \wedge j \in l)\}$. $S_{N_i} = \times_{j \in N_i} S_j$ refers to the states of i 's neighborhood. Similarly we define A_{N_i} , Ω_{N_i} , P_{N_i} and O_{N_i} .

b , the distribution over the initial state, is defined as $b(s) = b_u(s_u) \cdot \prod_{1 \leq i \leq n} b_i(s_i)$ where b_u and b_i refer to the distributions over initial unaffected state and over i 's initial state, respectively. We define $b_{N_i} = \prod_{j \in N_i} b_j(s_j)$. We assume that b is available to all agents (although it is possible to refine our model to make available to agent i only b_u , b_i and b_{N_i}). The goal in ND-POMDP is to compute joint policy $\pi = \langle \pi_1, \dots, \pi_n \rangle$ that maximizes the team's expected reward over a finite horizon T starting from b . π_i refers to the individual policy of agent i and is a mapping from the set of observation histories of i to A_i . π_{N_i} and π_l refer to the joint policies of the agents in N_i and hyper-link l respectively.

ND-POMDP can be thought of as an n -ary DCOP where the variable at each node is an individual agent's policy. The reward component R_l where $|l| = 1$ can be thought of as a local constraint while the reward component R_l where $|l| > 1$ corresponds to a non-local constraint in the constraint graph. In the next section, we push this analogy further by taking inspiration from the DBA algorithm [Yokoo and Hirayama, 1996], an algorithm for distributed constraint satisfaction, to develop an algorithm for solving ND-POMDPs.

We define *Local neighborhood utility* of agent i as the expected reward accruing due to the hyper-links that

contain agent i :

$$V_\pi[N_i] = \sum_{s_i, s_{N_i}, s_u} b_u(s_u) b_{N_i}(s_{N_i}) b_i(s_i) \sum_{l \in E \text{ s.t. } i \in l} V_{\pi_l}^0(s_{l1}, \dots, s_{lk}, s_u, \langle \rangle) \quad (1)$$

While trying to find best policy for agent i given its neighbors' policies, we do not need to consider non-neighbors' policies. This is the property of *locality of interaction* that is used in the following section.

3 Locally optimal policy generation

The locally optimal policy generation algorithm called LID-JESP (Locally interacting distributed joint equilibrium search for policies) is based on the DBA algorithm [Yokoo and Hirayama, 1996] and JESP [Nair *et al.*, 2003]. In this algorithm (see Algorithm 1), each agent tries to improve its policy with respect to its neighbors' policies in a distributed manner similar to DBA. Initially each agent i starts with a random policy and exchanges its policies with its neighbors. It then evaluates its contribution to the global value function, from its initial belief state b with respect to its current policy and its neighbors' policy (function EVALUATE()). Agent i then tries to improve upon its current policy by calling function GETVALUE, which returns the value of agent i 's best response to its neighbors' policies. Agent i then computes the gain that it can make to its local neighborhood utility, and exchanges its gain its neighbors. If i 's gain is greater than that one any of its neighbors, i changes its policy and sends its new policy to all its neighbors. This process of trying to improve the local policy is continued until termination, which is based on maintaining and exchanging a counter that counts the number of cycles where $gain_i = 0$. This is omitted from this article in order to simplify the presentation.

The algorithm for computing the best response is a dynamic-programming approach similar to that used in JESP. Here, we define an *episode* of agent i at time t as $e_i^t = \langle s_u^t, s_i^t, s_{N_i}^t, \tilde{\omega}_{N_i}^t \rangle$. Given that the neighbors' policies are fixed, treating episode as the state, results in a single agent POMDP, where the transition function and observation function can be defined as follows:

$$\begin{aligned} P'(e_i^t, a_i^t, e_i^{t+1}) &= P_u(s_u^t, s_u^{t+1}) \cdot P_i(s_i^t, s_u^t, a_i, s_i^{t+1}) \cdot \\ &\quad P_{N_i}(s_{N_i}^t, s_u^t, a_{N_i}, s_{N_i}^{t+1}) \cdot O_{N_i}(s_{N_i}^{t+1}, s_u^{t+1}, a_{N_i}, \omega_{N_i}) \\ O'(e_i^{t+1}, a_i^t, \omega_i^{t+1}) &= O_i(s_i^{t+1}, s_u^{t+1}, a_i, \omega_i) \end{aligned}$$

The function GETVALUE() returns the optimal policy for agent i given the above definitions of transition and observation functions and the policies of N_i . In this paper, GETVALUE() was implemented via value iteration but any other single agent POMDP algorithm could have been used.

4 Experimental Results

For our experiments, we ran the LID-JESP algorithm on the sensor domain (see Figure 1). The first benchmark

Algorithm 1 LID-JESP(*Agent i*)

```

1:  $\pi_i \leftarrow$  randomly selected policy,  $prevVal \leftarrow 0$ 
2: Exchange  $\pi_i$  with  $N_i$ 
3: while termination not detected do
4:   for all  $s_i, s_{N_i}, s_u$  do
5:      $prevVal \leftarrow \frac{1}{2} b_u(s_u) \cdot b_i(s_i) \cdot b_{N_i}(s_{N_i}) \cdot$ 
       EVALUATE(Agent i,  $s_i, s_u, s_{N_i}, \pi_i, \pi_{N_i}, \langle \cdot \rangle, \langle \cdot \rangle, 0, T$ )
6:    $gain_i \leftarrow$  GETVALUE(Agent i,  $b, \pi_{N_i}, 0, T$ )  $- prevVal$ 
7:   Exchange  $gain_i$  with  $N_i$ 
8:    $maxGain \leftarrow \max_{j \in N_i \cup \{i\}} gain_j$ 
9:    $winner \leftarrow \operatorname{argmax}_{j \in N_i \cup \{i\}} gain_j$ 
10:  if  $maxGain > 0$  and  $i = winner$  then
11:    initialize  $\pi_i$ 
12:    FINDPOLICY(Agent i,  $b, \langle \cdot \rangle, \pi_{N_i}, 0, T$ )
13:    Communicate  $\pi_i$  with  $N_i$ 
14:  else if  $maxGain > 0$  then
15:    Receive  $\pi_{winner}$  from winner and update  $\pi_{N_i}$ 
16: return  $\pi_i$ 

```

(JESP) is Nair *et al.*'s JESP algorithm [2003], which uses a centralized processor to find a locally optimal joint policy and does not consider the *interaction graph*. The second benchmark (LID-JESP-no-nw) is LID-JESP with a fully connected *interaction graph*. Figure 2(a) shows the run time in seconds on a logscale on the Y-axis for increasing finite horizon T on the X-axis, while Figure 2(b) shows the value of policy found on the Y-axis and increasing finite horizon T on the X-axis. All run times and values were obtained by averaging 5 runs, each with different randomly chosen starting policies. As seen in Figure 2(b), the values obtained for LID-JESP, JESP and LID-JESP-no-nw are quite similar, although LID-JESP and LID-JESP-no-nw often converged on a higher local optima than JESP. In comparing the run times, it should be noted that LID-JESP outperforms LID-JESP-no-nw and JESP (which could not be run for $T > 4$ within 10000 secs).

5 Acknowledgments

This material is based upon work supported by the DARPA/IPTO COORDINATORS program and the Air Force Research Laboratory under Contract No. FA8750-05-C-0030. The views and conclusions contained in this document are those of the authors, and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

References

- [Becker *et al.*, 2004] R. Becker, S. Zilberstein, V. Lesser, and C.V. Goldman. Solving transition independent decentralized Markov decision processes. *JAIR*, 22:423–455, 2004.
- [Goldman and Zilberstein, 2004] C.V. Goldman and S. Zilberstein. Decentralized control of cooperative systems: Categorization and complexity analysis. *JAIR*, 22:143–174, 2004.

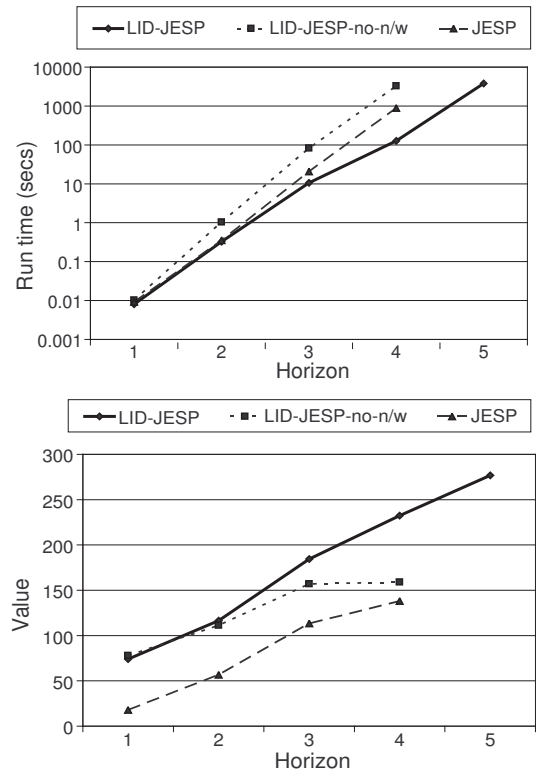


Figure 2: 5 agent F-config. (a) Run time (secs), (b) Value

- [Hansen *et al.*, 2004] E.A. Hansen, D.S. Bernstein, and S. Zilberstein. Dynamic Programming for Partially Observable Stochastic Games. In *AAAI*, 2004.
- [Lesser *et al.*, 2003] V. Lesser, C. Ortiz, and M. Tambe. *Distributed sensor nets: A multiagent perspective*. Kluwer academic publishers, 2003.
- [Montemerlo *et al.*, 2004] R. E. Montemerlo, G. Gordon, J. Schneider, and S. Thrun. Approximate solutions for partially observable stochastic games with common payoffs. In *AAMAS*, 2004.
- [Nair *et al.*, 2003] R. Nair, D. Pynadath, M. Yokoo, M. Tambe, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI*, 2003.
- [Yokoo and Hirayama, 1996] M. Yokoo and K. Hirayama. Distributed breakout algorithm for solving distributed constraint satisfaction problems. In *ICMAS*, 1996.

Coping with exceptions in multiclass ILP problems using possibilistic logic

Mathieu Serrurier

UPS, IRIT, 118 route de Narbonne,
31062 Toulouse, France
serrurie@irit.fr

Henri Prade

UPS, IRIT, 118 route de Narbonne,
31062 Toulouse, France
prade@irit.fr

Abstract

The handling of exceptions in multiclass problems is a tricky issue in inductive logic programming (ILP). In this paper we propose a new formalization of the ILP problem which accounts for default reasoning, and is encoded with first-order possibilistic logic. We show that this formalization allows us to handle rules with exceptions, and to prevent an example to be classified in more than one class. The possibilistic logic view of ILP problem, can be easily handled at the algorithmic level as an optimization problem.

1 Introduction

The handling of exceptions is a serious bottleneck in ILP due to the formalization in first-order logic. Indeed, when a rule has some exceptions, i.e. some examples are misclassified by it, there is no way to compensate these exceptions by means of another rule. So, a hypothesis accumulates all the exceptions of the rules that appear in it. Moreover, when dealing with more than two classes, a rule with some exceptions may prevent another one to perform the right classification. A proper handling of exceptions can be made by adding to the standard ILP setting a logical constraint that expresses that an example can be only classified into one class. Thus, having some exceptions may lead to inconsistency. In contrast with first-order logic, default reasoning encoded in possibilistic logic is well-suited for dealing with inconsistency. In this paper, we extend the ILP settings in a possibilistic way in order to handle inconsistency due to exceptions.

2 Preliminaries

Stated in the general context of first-order logic, the task of ILP is to find a non-trivial set of formulas H such that $B \cup H \models E$ given a background theory B and a set of examples E of the form $C(x, y)$ where x denotes the identification key of an example and y a class. E, B and H here denote sets of Horn clauses. In order to treat multiple classification of an example as inconsistency, we reformulate the ILP problem, by adding the following classification constraint $D \equiv \forall X, Y, Z \ C(X, Y) \cup C(X, Z) \rightarrow Y = Z$, as follows : given B, D and E , the goal is to find H such as

$$B \cup D \cup H \models E.$$

We extend propositional possibilistic logic [Dubois *et al.*, 1994] to the first-order case. Possibilistic logic is sound and complete for refutation, using an extended resolution rule, with respect to a semantics in terms of a complete plausibility preorder on the interpretations (encoded by a possibility distribution) [Dubois *et al.*, 1994]. S denotes a set of Herbrand interpretations. A possibility distribution can be defined on Herbrand interpretations as well. The possibilistic degree of a formula ϕ is the maximum possibility level of its models and is denoted $\Pi(\phi)$. Necessity is the dual notion of possibility. It refers to the possibility degrees of the counter-models of a formula : $N(\phi) = 1 - \Pi(\neg\phi)$. A possibilistic first-order formula is a pair (ϕ, α) where ϕ is a first-order formula and $\alpha \in]0, 1]$. It is understood as a constraint on an unknown necessity measure of the form $N(\phi) \geq \alpha$. Given a set K of possibilistic formulas, the α -cut of K is $K_\alpha = \{(\phi, \beta) \in K, \beta \geq \alpha\}$. Given T a set of classical first-order formulas, T is minimal w. r. t. a formula ϕ iff $T \models \phi$ and $\forall \psi \in T, T - \psi \not\models \phi$. The following definition avoids the drowning problem (i.e., consequences are lost when their implicants are taken in inconsistent sets of formulas, although they don't contribute to inconsistency) by checking if it exists a proof of ϕ in K_α , which is free from inconsistencies at level α .

Definition 1 Given K a set of possibilistic first-order formulas, $K \models_\pi (\phi, \alpha)$ iff $\exists K' \subset K_\alpha, K' \not\models \perp$ such as K' minimal w. r. t. ϕ and $\nexists K''$ minimal w. r. t. $\perp$ such that $K' \subset K'' \subset K_\alpha$.

In practice, this definition makes sure that, if two logical consequences are inconsistent, the one which has the highest necessity degree is preferred. As already advocated in the introduction, we propose to use possibilistic logic in ILP for a better handling of exceptions.

3 Possibilistic ILP

Given H a standard ILP hypothesis, let N_H denote a function (called priority function) which at each rule in H associates a priority level, to be understood as a necessity degree. This gives birth to a possibilistic hypothesis $H_p = \{(h, N_H(h)); h \in H\}$. Let B_p and D_p be composed by all

formulas that appear respectively in B and D , with 1 as necessity level. The priority function N_E over the examples is deduced from H_p as follows :

$$e \in E, N_E(e) = \begin{cases} 1, \text{ by convention, if } \exists \alpha > 0 \\ \text{such that } D_p \cup B_p \cup H_p \models_{\pi} (e, \alpha) \\ \max\{\alpha > 0, D_p \cup B_p \cup H_p \models_{\pi} (e, \alpha)\}, \\ \text{otherwise.} \end{cases} \quad (1)$$

Then, given D , B and E , the goal of possibilistic ILP is to find H_p , composed by a classical hypothesis H and an associated priority function N_H such that $B_p \cup D_p \cup H_p \models_{\pi} E_p$ with $E_p = \{(e, N_E(e)); e \in E\}$. Note that, if it exists such H_p , this hypothesis will be correct and complete. This enlarges the scope of classical ILP by learning sets of default rules in a framework that handles exceptions, the one of possibilistic logic.

Any possibilistic hypothesis, even it contains some rules with exceptions, can be completed in order to be correct and complete by adding the misclassified examples with 1 as necessity level. Then, the possibilistic ILP problem can be reformulated as an optimization problem :

Given D , B and E , the goal of possibilistic ILP is to find H_p that maximizes the accuracy (i.e. the proportion of well-classified examples). This definition of possibilistic ILP problem is fully in agreement with the paradigm of the minimization of the empirical risk. Since here, necessity levels are only used for obtaining an ordering of the formulas in H , this induces equivalence classes of priority functions N_H . Two priority functions N_H and N'_H , belong to the same equivalence class, i.e. $N_H \equiv N'_H$ if and only if $\forall h1, h2 \in H$ if $N_H(h1) > N_H(h2)$ then $N'_H(h1) > N'_H(h2)$.

Proposition 1 *Given H , finding the class of equivalence of priority functions such the accuracy is maximal is NP-complete with respect to the number of formulas in H .*

It shows that, although using a possibilistic rather than the classical setting is always more effective, finding the best priority function over formulas in H may be computationally very costly. Note that choosing a particular ordering based on the confidence or the support degrees of the rules is not optimal in general. It suggests to use heuristics for inducing hypotheses together with their priority function.

4 Experimentations

	A	B	C	D	E	Σ
Maximum	52	38	28	57	89	277
Foil	17	5	7	9	5	43
Indigo	21	14	9	18	33	95
Cilgg	26	12	10	18	37	103
Pilp avg	23	11.6	9.5	18.3	55.3	117.8
	[0]	[1.3]	[0.5]	[3.6]	[1.5]	[5.03]
Pilp max	23	13	10	23	57	125

In order to learn possibilistic logic rules, we learn a hypothesis directly together with its priority function. The hypotheses are learnt by using a stochastic exploration of the hypothesis space as explained in [Serrurier *et al.*, 2004]. The priority is found by using a greedy algorithm based on the 2-opt method

(switching the order of rules two by two while accuracy increases). Our algorithm is denoted as Pilp. Since Pilp is a non deterministic algorithm, the results that are presented are average results on 50 running steps with the same settings. The value in the brackets are standard derivations for the 50 results. For each experiment, two results are shown : Pilp avg is the average result for the 50 tests, and Pilp max is the best result found in the 50 tests. The experimentation is made with the finite element MESH Design dataset because it represents a typical hard ILP multiple class problem. It describes the structure of a mesh with unary and binary predicates. The dataset contains 277 examples that describe 13 classes. The dataset is split in 5 sub datasets denoted by A, B, C, D and E. The test of the accuracy of the algorithm is made by testing on one subset a hypothesis induced from the other sub datasets. Results are shown in the previous table. The results for the algorithms other than Pilp can be found in [Kietz, 2002]. The results are clearly in the favor of Pilp algorithm which increases the number of examples covered by the most effective algorithm up to 10% in average and 20% at maximum.

5 Conclusion

In this paper, we have proposed a new ILP formalization for dealing with exceptions in a multiple class problem. In order to do that, we have extended possibilistic propositional logic to a first-order setting. Then, by treating exceptions as inconsistency, we have reformulate the ILP problem in first-order possibilistic logic. In this reformulation, the ILP problem is turned in an optimization problem. This formalization allows our algorithm to learn sets of default rules. In this context, it may exist a correct and complete hypothesis which contains rules that have some exceptions. Possibilistic ILP is more flexible and more general than first-order decision lists [Mooney and Califf, 1995] and allows us to correctly cope with recursive hypotheses. Experiments have proved that an implementation of possibilistic ILP may be very effective for propositional or for relational learning and can compete with the best machine learning algorithms.

References

- [Dubois *et al.*, 1994] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. Gabbay *et al.*, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 3, pages 439–513. Oxford University Press, 1994.
- [Kietz, 2002] J. U. Kietz. Learnability of description logic programs. In S. Matwin and C. Sammut, editors, *Proceedings of the 12th Inter. Conference of Inductive Logic Programming, ILP 2002*, pages 117–132. Springer, 2002.
- [Mooney and Califf, 1995] R. Mooney and M. E. Califf. Induction of first-order decision lists: results on learning the past tense of english verbs. *Journal of Artificial Intelligence Research*, 3:1–24, 1995.
- [Serrurier *et al.*, 2004] M. Serrurier, H. Prade, and G. Richard. A simulated annealing framework for ilp. In R. Camacho, R. King, and A. Srinivasan, editors, *Proceedings of ILP 2004*, pages 288–304. LNAI 3194, Springer, 2004.

RESEARCH EXCELLENCE AWARD

What kind of a graphical model is the brain?

Geoffrey E. Hinton

Canadian Institute for Advanced Research &
Department of Computer Science, University of Toronto
10 Kings College Road
Toronto, Canada M5S 3G4
hinton@cs.toronto.edu

Abstract

If neurons are treated as latent variables, our visual systems are non-linear, densely-connected graphical models containing billions of variables and thousands of billions of parameters. Current algorithms would have difficulty learning a graphical model of this scale. Starting with an algorithm that has difficulty learning more than a few thousand parameters, I describe a series of progressively better learning algorithms all of which are designed to run on neuron-like hardware. The latest member of this series can learn deep, multi-layer belief nets quite rapidly. It turns a generic network with three hidden layers and 1.7 million connections into a very good generative model of handwritten digits. After learning, the model gives classification performance that is comparable to the best discriminative methods.

1 Introduction

Our perceptual systems make sense of the visual input using a neural network that contains about 10^{13} synapses. There has been much debate about whether our perceptual abilities should be attributed to a few million generations of blind evolution or to a few hundred million seconds of visual experience. Evolutionary search suffers from an information bottleneck because fitness is a scalar, so my bet is that the main contribution of evolution was to endow us with a learning algorithm that could make use of high-dimensional gradient vectors. These vectors provide millions of bits of information every second thus allowing us to perform a much larger search in one lifetime than evolution could perform in our entire evolutionary history.

So what is this magic learning algorithm? I have been involved in attempts to answer this question using undirected graphical models [Hinton and Sejnowski, 1986], directed graphical models [Hinton *et al.*, 1995], or no graphical model at all [Rumelhart *et al.*, 1986]. These attempts have failed as scientific theories of how the brain learns because they simply do not work well enough. They have, however, produced two neat tricks, one for learning undirected models and one for learning directed models. In this paper, I describe some

recent work in collaboration with Simon Osindero and Yee-Whye Teh that combines these two tricks in a surprising way to learn a hybrid generative model that was first proposed by Yee-Whye Teh. In this model, the top two layers form an *undirected* associative memory. The remaining layers form a directed acyclic graph that converts the representation in the associative memory into observable variables such as the pixels of an image. In addition to working well, this hybrid model has some other nice features:

1. The learning finds a fairly good model quickly even in deep directed networks with millions of parameters and many hidden layers. For optimal performance, however, a slower fine-tuning phase is required.
2. The learning algorithm builds a full generative model of the data which makes it easy to see what the distributed representations in the deeper layers have in mind.
3. The inference required for forming a percept is both fast and accurate.
4. The learning algorithm is unsupervised. For labeled data, it learns a model that generates both the label and the data.
5. The learning is local: adjustments to a synapse strength depend only on the states of the pre-synaptic and post-synaptic neuron.
6. The communication is simple: neurons only need to communicate their stochastic binary states.

Section 2 describes a simple learning algorithm for undirected, densely-connected, networks composed of stochastic binary variables some of which are unobserved. Section 3 shows how to make this simple algorithm efficient by restricting the architecture of the network. Section 4 introduces the idea of variational approximations for learning directed graphical models in which correct inference is intractable and describes the “wake-sleep” algorithm that makes use of a variational approximation in a multi-layer, directed network of stochastic binary variables. All of these sections can be safely ignored by people already familiar with these ideas.

Section 5 introduces the novel idea of a “complementary” prior. Complementary priors seem about as probable as father Christmas because, by definition, they exactly cancel the “explaining away” phenomenon that makes inference difficult in

directed models. Section 5.1 includes a simple example of a complementary prior and shows the equivalence between restricted Boltzmann machines and infinite directed networks with tied weights.

Section 6 introduces a fast, greedy learning algorithm for constructing multi-layer directed networks one layer at a time. Using a variational bound it shows that as each new layer is added, the overall generative model improves. The greedy algorithm resembles boosting in its repeated use of the same “weak” learner, but instead of re-weighting each data-vector to ensure that the next step learns something new, it re-represents it. Curiously, the weak learner that is used to construct deep directed nets is itself an undirected graphical model.

Section 7 shows how the weights produced by the efficient greedy algorithm can be fine-tuned using the “up-down” algorithm which is a contrastive version of the wake-sleep algorithm.

Section 8 shows the pattern recognition performance of a network with three hidden layers and about 1.7 million weights on the standard MNIST set of handwritten digits. When no knowledge of geometry is provided and there is no special preprocessing, the generalization performance of the network is 1.25% errors on the official test set. This beats the 1.5% achieved by the best back-propagation nets when they are not hand-crafted for this particular application, and it is quite close to the 1.1% or 1.0% achieved by the best support vector machines.

Finally, section 9 shows what happens in the mind of the model when it is running without being constrained by visual input. The network has a full generative model, so it is easy to look into its mind – we simply generate an image from its high-level representations.

Throughout the paper, we will consider nets composed of stochastic binary variables but the ideas can be generalized to other models in which the log probability of a variable is an additive function of the states of its directly-connected neighbours.

2 The Boltzmann machine learning algorithm

A Boltzmann machine (Hinton and Sejnowski, 1986) is a network of stochastic binary units with symmetric connections. It is usually divided into a set of “visible” units which can have data-vectors clamped on them, and a set of hidden units that act as latent variables (see figure 1). Each unit, i , adopts its “on” state with a probability that is a logistic function of the inputs it receives from other units, j :

$$p(s_i = 1) = \frac{1}{1 + \exp(-b_i - \sum_j s_j w_{ij})} \quad (1)$$

where b_i is the bias of unit i , and w_{ij} is the weight on the symmetric connection between i and j . The weights and biases of a Boltzmann machine define an energy function over global configurations (i.e. binary state vectors) of the net. Using α as an index over configurations of the visible units, and β for configurations of the hidden units:

$$E^{\alpha\beta} = - \sum_i b_i s_i^{\alpha\beta} - \sum_{i < j} s_i^{\alpha\beta} s_j^{\alpha\beta} w_{ij} \quad (2)$$

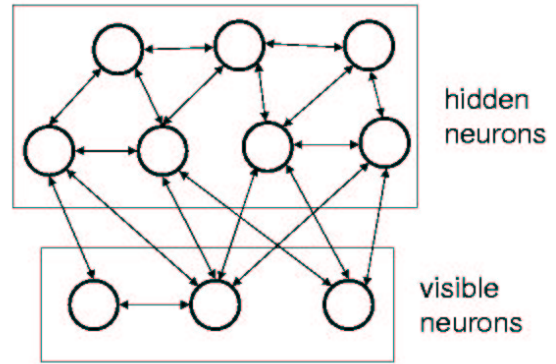


Figure 1: A Boltzmann machine composed of stochastic binary units with symmetric connections. When data is clamped on the visible units, a simple stochastic updating rule infers a configuration of states of the hidden units that is a good interpretation of the data. If no data is clamped and the same updating rule is used for all of the units, the network generates visible vectors from its model.

where $s_i^{\alpha\beta}$ is the binary state of unit i in the global binary configuration $\alpha\beta$. If units are chosen at random and their binary states are updated using the stochastic activation rule in Eq. 1 the Boltzmann machine will eventually converge to a stationary probability distribution in which the probability of finding it in any global state is determined by the energy of that state relative to the energies of all the other global states:

$$P^{\alpha\beta} = \frac{\exp(-E^{\alpha\beta})}{\sum_{\gamma\delta} \exp(-E^{\gamma\delta})} \quad (3)$$

If we sum over all configurations of the hidden units, we get the probability, at thermal equilibrium, of finding the visible units in configuration α

$$P^\alpha = \frac{\sum_\beta \exp(-E^{\alpha\beta})}{\sum_{\gamma\delta} \exp(-E^{\gamma\delta})} \quad (4)$$

A Boltzmann machine can be viewed as a generative model that assigns a probability, via Eq. 4, to each possible binary state vector over its visible units. By changing the weights and biases, we can change the probability that the model assigns to each possible visible vector. So we can model a set of training vectors by adjusting the weights and biases to maximize the sum of their log probabilities. A nice feature of Boltzmann machines is that the maximum likelihood learning rule for the weights is both simple and local. We can learn a locally optimal set of weights by collecting two sets of statistics:

- In the *positive phase* we clamp a training vector on the visible units and then repeatedly update the hidden units in random order using Eq. 1. Once the distribution over hidden configurations has reached “thermal” equilibrium with the clamped data-vector, we sample the hidden state and record which pairs of units are both

on. By repeating this for the entire training set, we can compute $\langle s_i s_j \rangle^+$, the average correlation between i and j when data is clamped on the visible units.

- In the *negative phase* we let the network run freely with the visible units unclamped. Their states are updated in just the same way as the hidden units. Once the distribution over global configurations has reached equilibrium, we sample the states of all the units and record which pairs are both on. By repeating this many times, we can compute $\langle s_i s_j \rangle^-$, the average correlation between i and j when the network is running freely and therefore producing samples from its generative model.

We can then follow the gradient of the log probability of the data by using the simple rule

$$\Delta w_{ij} = \epsilon (\langle s_i s_j \rangle^+ - \langle s_i s_j \rangle^-) \quad (5)$$

where ϵ is a learning rate. It is very surprising that the learning rule is so simple because the gradient of the log likelihood with respect to one weight depends in complicated ways on all the other weights. In the back-propagation algorithm, these dependencies are computed explicitly in the backward pass. In the Boltzmann machine they show up as the difference between the local correlations in the positive and negative phases.

Unfortunately, the simplicity and generality of the Boltzmann machine learning algorithm come at a price. It can take a very long time for the network to settle to thermal equilibrium, especially in the negative phase when it is unconstrained by data but needs to be highly multi-modal. Also, the gradient used for learning is very noisy because it is the difference of two noisy expectations. These problems make the general form of the algorithm impractical for large networks with many hidden units.

3 Restricted Boltzmann machines and contrastive divergence learning

If we are willing to restrict the architecture of a Boltzmann machine by not allowing connections between hidden units, the positive phase no longer requires any settling. With a data-vector clamped on the visible units, the hidden units are all conditionally independent, so we can apply the update rule in Eq. 1 to all the units at the same time to get an unbiased sample from the posterior distribution over hidden configurations. This makes it easy to measure the first correlation in Eq. 5.

If we also prohibit connections between visible units, we can update all of the visible units in parallel given a hidden configuration. So the second correlation in Eq. 5 can be found by alternating Gibbs sampling as shown in figure 2. Unfortunately we may need to run the alternating Gibbs sampling for a long time before the Markov chain converges to the equilibrium distribution. Fortunately, if we start the Markov chain at the data distribution, learning still works well even if we only run the chain for a few steps [Hinton, 2002]. This gives an efficient learning rule:

$$\Delta w_{ij} = \epsilon (\langle s_i s_j \rangle^0 - \langle s_i s_j \rangle^n) \quad (6)$$

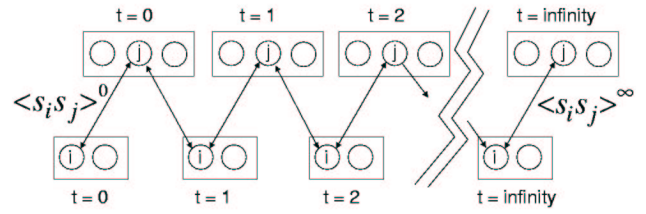


Figure 2: This depicts a Markov chain that uses alternating Gibbs sampling. In one full step of Gibbs sampling, the hidden units in the top layer are all updated in parallel by applying Eq. 1 to the inputs received from the the current states of the visible units in the bottom layer, then the visible units are all updated in parallel given the current hidden states. The chain is initialized by setting the binary states of the visible units to be the same as a data-vector. The correlations in the activities of a visible and a hidden unit are measured after the first update of the hidden units and again at the end of the chain. The difference of these two correlations provides the learning signal for updating the weight on the connection.

where the superscript 0 indicates that the correlation is measured at the start of the chain with a data-vector clamped on the visible units and n indicates that the correlation is measured after n full steps of Gibbs sampling. The angle brackets denote expectations over both the choice of data-vector and the stochastic updating used in the Gibbs sampling. This learning rule does not follow the gradient of the log likelihood, but it does closely approximate the gradient of another function, contrastive divergence, which is the difference of two Kullback-Leibler divergences [Hinton, 2002]. Intuitively, it is not necessary to run the chain to equilibrium in order to see how the data distribution is being systematically distorted by the model. If we just run the chain for a few steps and then lower the energy of the data and raise the energy of whichever configuration the chain preferred to the data, we will make the model more likely to generate the data and less likely to generate the alternatives. An empirical investigation of the relationship between the maximum likelihood and the contrastive divergence learning rules can be found in [Carreira-Perpinan and Hinton, 2005].

Contrastive divergence learning in a restricted Boltzmann machine is efficient enough to be practical [Mayraz and Hinton, 2001]. Variations that use real-valued units and different sampling schemes are described in [Teh *et al.*, 2003] and have been quite successful for modeling the formation of topographic maps [Welling *et al.*, 2003] and for denoising natural images [Roth and Black, 2005]. However, it appears that the efficiency has been bought at a high price: It is not possible to have deep, multilayer nets because these take far too long even to reach *conditional* equilibrium with a clamped data-vector. Also, nets with symmetric connections do not give a causal model in which the data is explained in terms of underlying causes.

The next section describes a simple learning algorithm for an apparently quite different type of network that uses di-

rected connections. This learning algorithm also has deficiencies, but it can be combined with contrastive divergence learning in a surprising way to produce an algorithm that works much better and is significantly more similar to the real brain.

4 Variational learning

Inference in directed graphical models that use non-linear, distributed representations is difficult because of a phenomenon called “explaining away” [Pearl, 1988] which creates dependencies between hidden variables. This is illustrated in figure 3. Radford Neal [Neal, 1992] showed that it was possible to use Gibbs sampling to perform inference correctly in multilayer *directed* networks composed of the same type of binary stochastic units as are used in Boltzmann machines. The communication required is more complicated than in a Boltzmann machine because in addition to seeing the binary states of its ancestors and descendants, a unit needs to see the probability that each of its descendants would be turned on by the current states of all that descendant’s ancestors. However, once we have a sample from the posterior distribution over configurations of the hidden units, the maximum likelihood learning rule for updating the directed connection from j to i is very simple:

$$\Delta w_{ji} = \epsilon s_j (s_i - \hat{s}_i) \quad (7)$$

where ϵ is a learning rate and $\hat{s}_i$ is the probability that i would be turned on by the current states of all its ancestors. There is no need for a “negative phase” because directed models do not require the awkward normalizing term that shows up in the denominator of Eq. 3. Radford Neal showed that logistic belief nets are somewhat easier to learn than Boltzmann machines, but the use of Gibbs sampling to get samples from the posterior distribution still makes it very tedious to learn large, deep nets.

Rich Zemel and I realised that it might still be possible to learn a belief net that contained a layer of binary stochastic hidden units even when the cost of computing the posterior distribution, or sampling from it, was prohibitive. Instead of trying to perform maximum likelihood learning, we adopted a coding perspective and attempted to learn a model that would minimize the description length of the data [Zemel and Hinton, 1995]. The idea is that the sender and receiver both have access to the model and instead of communicating a data-vector directly, the sender first communicates a hidden configuration of the model. This costs some bits, but it also gives the receiver a good idea of what data to expect. Given these expectations, the data-vector can be communicated more cheaply¹. So it appears that the expected cost of communicating a data-vector is:

$$C(x) = - \sum_{\alpha} Q(y_{\alpha}|x) [\log p(y_{\alpha}) + \log p(x|y_{\alpha})] \quad (8)$$

¹Shannon showed that, using an efficient block coding scheme, the cost of communicating a discrete value to a receiver is asymptotically equal to the negative log probability of that value under a probability distribution that has already been agreed upon by the sender and the receiver.

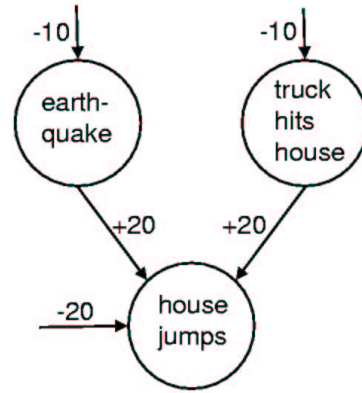


Figure 3: A simple logistic belief net containing two independent, rare causes that become highly anti-correlated when we observe the house jumping. The bias of -10 on the earthquake node means that, in the absence of any observation, this node is e^{10} times more likely to be off than on. If the earthquake node is on and the truck node is off, the jump node has a total input of 0 which means that it has an even chance of being on. This is a much better explanation of the observation that the house jumped than the odds of e^{-20} which apply if neither of the hidden causes is active. But it is wasteful to turn on both hidden causes to explain the observation because the odds on them both happening are $e^{-10} \times e^{-10} = e^{-20}$.

where y_{α} is a configuration of the hidden units, $Q(y_{\alpha}|x)$ is the probability that the sender will choose to use y_{α} in order to communicate data-vector x , $\log p(y_{\alpha})$ is the cost of communicating y_{α} to a receiver who already has the model and $\log p(x|y_{\alpha})$ is the cost of communicating x to a receiver who has both the model and the hidden configuration y_{α} . Rich soon discovered that it was better to minimize a different function and we eventually understood why.

Suppose there are two different hidden configurations that give the same communication cost for a data-vector. The sender can flip a coin to decide which one to use. After receiving the data-vector, the receiver can figure out what choices were available to the sender and he can therefore figure out the value of the random bit produced by the coin. So if there are two equally good hidden configurations, the sender can communicate one additional bit from a random bit stream by his choice of configuration. In general, the number of extra bits is equal to the entropy of the sender’s distribution across hidden configurations. All of these extra bits can be used to communicate some other bit string, so we need to subtract them from the communication cost of the data-vector:

$$C(x) = - \sum_{\alpha} Q(y_{\alpha}|x) [\log p(y_{\alpha}) + \log p(x|y_{\alpha})] - \left(- \sum_{\alpha} Q(y_{\alpha}|x) \log Q(y_{\alpha}|x) \right) \quad (9)$$

If the sender picks hidden configurations from their true posterior distribution, the communication cost is minimized and is equal to the negative log probability of the data-vector

under the model. But if it is hard to sample from the true posterior, the sender can use any other distribution he chooses. The communication cost goes up, but it is still perfectly well-defined. The sender could, for example, insist on using a factorial distribution in which the states of the hidden units are chosen independently. The communication cost will then be an upper bound on the negative log probability of the data under the model and by minimizing the communication cost we will either push down the negative log probability of the data or we will make the bound tighter. The looseness of the bound is just the Kullback-Liebler divergence between the distribution used by the sender and the true posterior, $P(y_\alpha|x)$.

$$-\log p(x) = C(x) - \sum_{\alpha} Q(y_{\alpha}|x) \log \frac{Q(y_{\alpha}|x)}{p(y_{\alpha}|x)} \quad (10)$$

The use of an approximate posterior distribution to bound $\log p(x)$ [Neal and Hinton, 1998] is now a standard way to learn belief nets in which inference is intractable [Jordan *et al.*, 1999].

4.1 The wake-sleep algorithm

A simple way to make use of variational learning in a multi-layer logistic belief net is to use a set of “recognition” connections that compute a factorial approximation to the posterior distribution in one layer when given the binary states of the units in the layer below [Hinton *et al.*, 1995]. These recognition connections will not, in general, have the same values as the corresponding generative connections. Given a set of recognition weights, it is easy to update the generative weights to follow the gradient of the description cost in Eq. 9. We use a data-vector to set the states of the visible units and then we use the recognition connections to compute a probability for each unit in the first hidden layer. Then we use these probabilities to pick independent binary states for all the units in that layer. This is repeated for each hidden layer in turn until we have a sample from the approximate posterior. Given this sample the learning rule for the generative, top-down weights is given by Eq. 7. This is the “wake” phase of the wake-sleep algorithm.

The “correct” way to learn the recognition weights is to follow the derivative of the cost in Eq. 9. The recognition weights only affect the Q terms; they do not affect the p terms. However, the derivatives w.r.t the Q terms are messy because Q comes outside the log. So we make a further approximation. In the “sleep” phase, we perform an ancestral pass to generate a sample from the generative model. Starting at the top layer, we pick binary states for the units from their independent priors. Then we pick states for the units in each lower layer using the probabilities computed by applying the generative weights to the states in the layer above. Once we have completed an ancestral pass, we have both a visible vector and its true hidden causes. So we can adjust the recognition weights to be better at recovering the hidden causes from the states of the units in the layer below:

$$\Delta w_{ij} = \epsilon s_i (s_j - \hat{s}_j) \quad (11)$$

where $\hat{s}_j$ is the probability that j would be turned on by the current states of all its descendants.

The wake-sleep algorithm works quite well, but the sleep phase is not exactly following the gradient of the variational bound. As a result, it does the wrong thing when a data-vector can be generated by two quite different hidden configurations: Instead of picking one of these hidden configurations and sticking with it, it averages the configurations to produce a vague factorial distribution that gives significant probability to many poor configurations.

5 Complementary priors

The phenomenon of explaining away makes inference difficult in directed networks. It is comforting that we can still improve the parameters even when inference is done incorrectly, but it would be much better to find a way of eliminating explaining away altogether, even in models whose hidden variables have highly correlated effects on the visible variables. Most people who use directed graphical models regard this as impossible.

In a logistic belief net with one hidden layer, the prior distribution over the hidden variables is factorial because their binary states are chosen independently when the model is used to generate data. The non-independence in the posterior distribution is created by the likelihood term coming from the data. Perhaps we could eliminate explaining away in the first hidden layer by using extra hidden layers to create a “complementary” prior that has exactly the opposite correlations to those in the likelihood term. Then, when the likelihood term is multiplied by the prior, we will get a posterior that is exactly factorial. This seems pretty implausible, but figure 4 shows a simple example of a logistic belief net with replicated weights in which the priors are complementary *at every hidden layer*. This net has some interesting properties.

5.1 An infinite directed model with tied weights

We can generate data from the infinite directed net by starting with a random configuration at an infinitely deep hidden layer and then performing an ancestral pass all the way down to the visible variables. Clearly, the distribution that we will get over the visible variables is exactly the same as the distribution produced by the Markov chain in figure 2 so the infinite directed net with tied weights is equivalent to a restricted Boltzmann machine.²

We can sample from the true posterior distribution over all of the hidden layers of the infinite directed net by starting with a data vector on the visible units and then using the transposed weight matrices to infer the factorial distributions over each hidden layer in turn. At each hidden layer we sample from the factorial posterior before computing the factorial posterior for the layer above. This is exactly the same process as starting a restricted Boltzmann machine at the data and letting it settle to equilibrium. It is also exactly the same as the inference procedure used in the wake-sleep algorithm, but in this net it gives unbiased samples because the complementary prior at each layer ensures that the posterior distribution really is factorial.

²We can interpret any undirected model that uses Gibbs sampling as an infinite directed model with tied weights.

Since we can easily sample from the true posterior, it is easy to learn the weights in the infinite directed net. Let us start by computing the derivative for a generative weight, w_{ij}^{00} , from a unit j in layer H_0 to unit i in layer V_0 (see figure 4). In a logistic belief net, the maximum likelihood learning rule for a single data-vector, x , is:

$$\frac{\partial \log p(x)}{\partial w_{ij}^{00}} = \langle s_j^0 (s_i^0 - \hat{s}_i^0) \rangle \quad (12)$$

where $\langle \cdot \rangle$ denotes an average over the sampled states and $\hat{s}_i^0$ is the probability that unit i would be turned on if the visible vector was stochastically reconstructed from the sampled hidden states. Computing the posterior distribution over the second hidden layer, V_1 , from the sampled binary states in the first hidden layer, H_0 , is exactly the same process as reconstructing the data, so s_i^1 is a sample from $\hat{s}_i^0$ and the learning rule becomes:

$$\frac{\partial \log p(x)}{\partial w_{ij}^{00}} = \langle s_j^0 (s_i^0 - s_i^1) \rangle \quad (13)$$

Since the weights are replicated, the full derivative for a generative weight is obtained by summing the derivatives of the generative weights between all pairs of layers:

$$\begin{aligned} \frac{\partial \log p(x)}{\partial w_{ij}} &= \langle s_j^0 (s_i^0 - s_i^1) \rangle \\ &\quad + \langle s_i^1 (s_j^0 - s_j^1) \rangle \\ &\quad + \langle s_j^1 (s_i^1 - s_i^2) \rangle \\ &\quad + \dots \end{aligned}$$

All of the vertically aligned terms cancel leaving the Boltzmann machine learning rule of Eq. 5. The same weights are also used for inference and one might expect this to contribute extra derivatives. Fortunately, all of these derivatives are zero. The inference is exact so, to first order, small changes in the inferred posteriors make no change in the log probability of the data. The only reason the recognition weights ever change is because they are tied to the generative weights.

6 A greedy learning algorithm for transforming representations

An efficient way to learn a complicated model is to combine a set of simpler models that are learned sequentially. To force each model in the sequence to learn something different from the previous models, the data is modified in some way after each model has been learned. In boosting [Freund, 1995], each model in the sequence is trained on re-weighted data that emphasizes the cases that the preceding models got wrong. In one version of principal components analysis, the variance in a modeled direction is removed thus forcing the next modeled direction to lie in the orthogonal subspace. In projection pursuit [Friedman and Stuetzle, 1981], the data is transformed by nonlinearly distorting one direction in the data-space to remove all non-Gaussianity in that direction. The idea behind our greedy algorithm is to allow each model in the sequence to receive a different representation of the data. The model

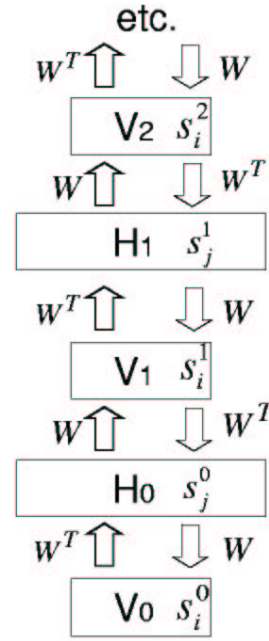


Figure 4: An infinite logistic belief net with tied weights.

performs a non-linear transformation on its input vectors and produces as output the vectors that will be used as input for the next model in the sequence.

Figure 5 shows a multilayer generative model in which the top two layers interact via undirected connections and all of the other connections are directed. There are no intra-layer connections and, to simplify the analysis, all layers have the same number of units. It is possible to learn sensible (though not optimal) values for the parameters W_0 by assuming that the parameters between higher layers will be used to construct a complimentary prior for W_0 . This is equivalent to assuming that all of the weight matrices are constrained to be equal. The task of learning W_0 under this assumption reduces to the task of learning an RBM and although this is still difficult, good approximate solutions can be found rapidly by minimizing contrastive divergence (Hinton, 2002). Once W_0 has been learned, the data can be mapped through W_0^T to create higher-level “data” at the first hidden layer.

If the RBM is a perfect model of the original data, the higher-level “data” will already be modeled perfectly by the higher-level weight matrices. Generally, however, the RBM will not be able to model the original data perfectly and we can make the generative model better using the following greedy algorithm:

1. Learn W_0 assuming all the weight matrices are tied.
2. Freeze W_0 and commit ourselves to using W_0^T to infer factorial approximate posterior distributions over the states of the variables in the first hidden layer.
3. Keeping all the higher weight matrices tied to each other, but untied from W_0 , learn an RBM model of the

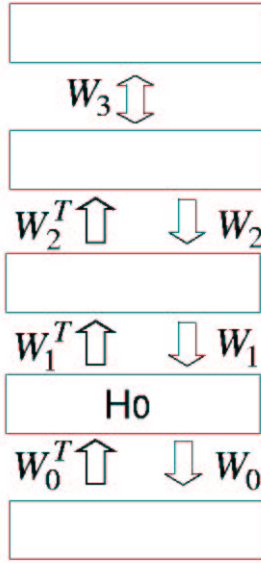


Figure 5: A hybrid network. The top two layers have undirected connections and form an associative memory. The layers below have directed, top-down, generative connections that can be used to map a state of the associative memory to an image. There are also directed, bottom-up, recognition connections that are used to infer a factorial representation in one layer from the binary activities in the layer below. In the greedy initial learning the recognition connections are tied to the generative connections.

higher-level “data” that is produced by using W_0^T to transform the original data.

If this greedy algorithm changes the higher-level weight matrices, it is guaranteed to improve the generative model. The log probability of a single data-vector, x , under the multilayer generative model is bounded by

$$\begin{aligned} \log p(x) &\geq \sum_{\alpha} Q(y_{\alpha}|x) (\log p(y_{\alpha}) + \log p(x|y_{\alpha})) \\ &\quad - \sum_{\alpha} Q(y_{\alpha}|x) \log Q(y_{\alpha}|x) \end{aligned}$$

where y_{α} is a binary configuration of the units in the first hidden layer, $p(y_{\alpha})$ is the prior probability of y_{α} under the model defined by the weights above H_0 and $Q(\cdot|x)$ is any probability distribution over y . The bound becomes an equality if and only if $Q(\cdot|x)$ is the true posterior distribution.

When all of the weight matrices are tied together, the factorial distribution over H_0 produced by applying W_0^T to a data-vector is the true posterior distribution, so at step 2 of the greedy algorithm $\log p(x)$ is equal to the bound. Step 2 freezes both $Q(\cdot|x)$ and $p(x|y_{\alpha})$ and with these terms fixed, the derivative of the bound is the same as the derivative of

$$\sum_{\alpha} Q(y_{\alpha}|x) \log p(y_{\alpha}) \quad (14)$$

So maximizing the bound w.r.t. the weights in the higher layers is exactly equivalent to maximizing the log probability of

a dataset in which y_{α} occurs with probability $Q(y_{\alpha}|x)$. If the bound becomes tighter, it is possible for $\log p(x)$ to fall even though the lower bound on it increases, but $\log p(x)$ can never fall below its value at step 2 of the greedy algorithm because the bound is tight at this point and the bound always increases.

The greedy algorithm can clearly be applied recursively, so if we use the full maximum likelihood Boltzmann machine learning algorithm to learn each set of tied weights and then we untie the bottom layer of the set from the weights above, we can learn the weights one layer at a time with a guarantee³ that we will never decrease the log probability of the data under the full generative model. In practice, we replace maximum likelihood Boltzmann machine learning algorithm by contrastive divergence learning because it works well and is much faster. The use of contrastive divergence voids the guarantee, but it is still reassuring to know that extra layers are guaranteed to improve imperfect models if we learn each layer with sufficient patience.

To guarantee that the generative model is improved by greedily learning more layers, it is convenient to consider models in which all layers are the same size so that the higher-level weights can be initialized to the values learned before they are untied from the weights in the layer below. The same greedy algorithm, however, can be applied even when the layers are different sizes.

7 Back-Fitting with the up-down algorithm

Learning the weight matrices one layer at a time is efficient but far from optimal. Once the weights in higher layers have been learned, neither the weights nor the simple inference procedure are optimal for the lower layers. The suboptimality produced by greedy learning is relatively innocuous for supervised methods like boosting. Labels are often scarce and each label may only provide a few bits of constraint on the parameters, so over-fitting is typically more of a problem than under-fitting. Going back and refitting the earlier models may, therefore, cause more harm than good. Unsupervised methods, however, can use very large unlabeled datasets and each case may be very high-dimensional thus providing many bits of constraint on a generative model. Under-fitting is then a serious problem which can be alleviated by a subsequent stage of back-fitting in which the weights that were learned first are revised to fit in better with the weights that were learned later.

After greedily learning good initial values for the weights in every layer, we untie the “recognition” weights that are used for inference from the “generative” weights that define the model, but retain the restriction that the posterior in each layer must be approximated by a factorial distribution in which the variables within a layer are conditionally independent given the values of the variables in the layer below. A variant of the wake-sleep algorithm described in section 4.1 can then be used to allow the higher-level weights to influence the lower level ones. In the “up-pass”, the recognition weights are used in a bottom-up pass that stochastically picks

³The guarantee is on the *expected* change in the log probability.

a state for every hidden variable. The generative weights on the directed connections are then adjusted using the maximum likelihood learning rule in Eq. 7. Because weights are no longer tied to the weights above them, $\hat{s}_i$ must be computed using the states of the variables in the layer above i and the generative weights from these variables to i . The weights on the undirected connections at the top level are learned as before by fitting the top-level RBM to the posterior distribution of the penultimate layer.

The “down-pass” starts with a state of the top-level associative memory and uses the top-down generative connections to stochastically activate each lower layer in turn. During the down-pass, the top-level undirected connections and the generative directed connections are not changed. Only the bottom-up recognition weights are modified. This is equivalent to the sleep phase of the wake-sleep algorithm if the associative memory is allowed to settle to its equilibrium distribution before initiating the down-pass. But if the associative memory is initialized by an up-pass and then only allowed to run for a few iterations of alternating Gibbs sampling before initiating the down-pass, this is a “contrastive” form of the wake-sleep algorithm which eliminates the need to sample from the equilibrium distribution of the associative memory. The contrastive form also fixes several other problems of the sleep phase. It ensures that the recognition weights are being learned for representations that resemble those used for real data and it also helps to eliminate the problem of mode averaging. If, given a particular data vector, the current recognition weights always pick a particular mode at the level above and ignore other very different modes that are equally good at generating the data, the learning in the down-pass will not try to alter those recognition weights to recover any of the other modes as it would if the sleep phase used a pure ancestral pass.

By using a top-level associative memory we also eliminate a problem in the wake phase: Independent top-level units seem to be required to allow an ancestral pass, but they mean that the variational approximation is very poor for the top layer of weights.

8 Performance on the MNIST database

8.1 Training the network

The MNIST database of handwritten digits contains 60,000 training images and 10,000 test images. Results for many different pattern recognition techniques are already published for this database so it is ideal for evaluating new pattern recognition methods. The network⁴ shown in figure 6 was trained on 44,000 of the training images that were divided into 440 balanced mini-batches each containing 10 examples of each digit class. The weights were updated after each mini-batch.

In the initial phase of training, the greedy algorithm described in section 6 was used to train each layer of weights

⁴Preliminary experiments with 16×16 images of handwritten digits from the USPS database showed that a good way to model the joint distribution of digit images and their labels was to use an architecture of this type, but for 16×16 images, only 3/5 as many units were used in each hidden layer.

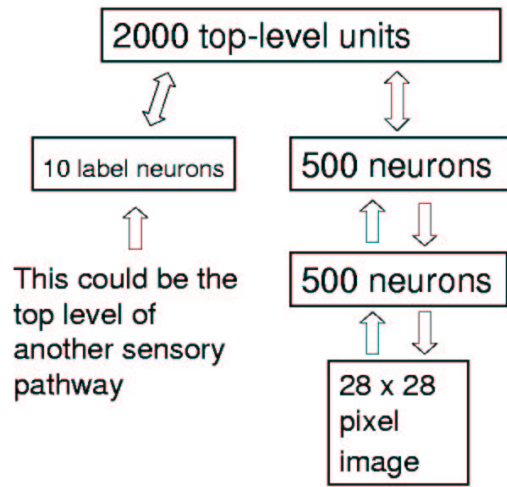


Figure 6: The network used to model the joint distribution of digit images and digit labels.

separately, starting at the bottom. Each layer was trained for 30 sweeps through the training set (called “epochs”). During training, the units in the “visible” layer of each RBM had real-valued activities between 0 and 1. These were the normalized pixel intensities when learning the the bottom layer of weights. For training higher layers of weights, the real-valued activities of the visible units in the RBM were the activation probabilities of the hidden units in the lower-level RBM. The hidden layer of each RBM used stochastic binary values when that RBM was being trained. The greedy training took a few hours in Matlab on a 3GHz Xeon processor and when it was done, the error-rate on the test set was 2.49% (see below for details of how the network is tested).

When training the top layer of weights (the ones in the associative memory) the labels were provided as part of the input. The labels were represented by turning on one unit in a “softmax” group of 10 units. When the activities in this group were reconstructed from the activities in the layer above, exactly one unit was allowed to be active and the probability of picking unit i was given by:

$$p_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad (15)$$

where x_i is the total input received by unit i . Curiously, the learning rules are unaffected by the competition between units in a softmax group, so the synapses do not need to know which unit is competing with which other unit. The competition affects the probability of a unit turning on, but it is only this probability that affects the learning.

After the greedy layer-by-layer training, the network was trained, with a different learning rate and weight-decay, for 300 epochs using the up-down algorithm described in section 7. The learning rate, momentum, and weight-decay⁵ were

⁵No attempt was made to use different learning rates or weight-decays for different layers and the learning rate and momentum were

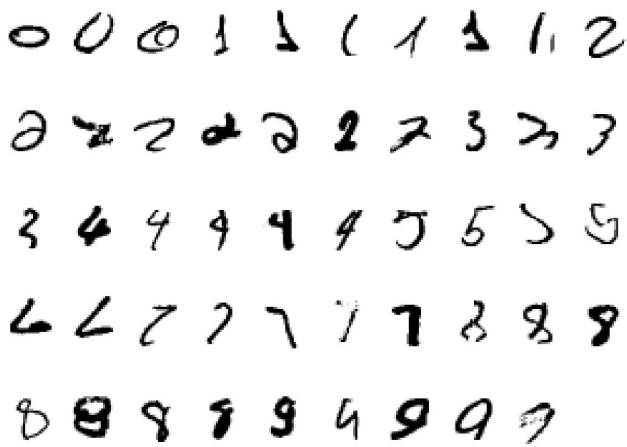


Figure 7: All 49 cases in which the network guessed right but had a second guess whose probability was within 0.3 of the probability of the best guess. The true classes are arranged in standard scan order.

chosen by training the network several times and observing its performance on a separate validation set of 10,000 images that were taken from the remainder of the full training set. For the first 100 epochs of the up-down algorithm, the up-pass was followed by three full iterations of alternating Gibbs sampling in the associative memory before performing the down-pass. For the second 100 epochs, six iterations were performed, and for the last 100 epochs, ten iterations were performed. Each time the number of iterations of Gibbs sampling was raised, the error on the validation set decreased noticeably.

The network that performed best on the validation set was then tested and had an error rate of 1.39%. This network was then trained on all 60,000 training images⁶ until its error-rate on the full training set was as low as its final error-rate had been on the initial training set of 44,000 images. This took a further 59 epochs making the total learning time about a week. The final network had an error-rate of 1.25%. The errors made by the network are shown in figure 8. The 49 cases that the network gets correct but for which the second best probability is within 0.3 of the best probability are shown in figure 7.

The error-rate of 1.25% compares very favorably with the error-rates achieved by discriminative, feed-forward neural networks that have one or two hidden layers and are trained by back-propagation. When the detailed connectivity of these networks is not hand-crafted for this particular task, the best reported error-rate for stochastic on-line learning with a separate squared error on each of the 10 output units is 2.95%. These error-rates can be reduced to 1.51% by using small

always set quite conservatively to avoid oscillations. It is highly likely that the learning speed could be considerably improved by a more careful choice of learning parameters, though it is possible that this would lead to worse solutions.

⁶The training set has unequal numbers of each class, so images were assigned randomly to each of the 600 mini-batches.



Figure 8: The 125 test cases that the network got wrong. Each case is labeled by the network's guess. The true classes are arranged in standard scan order.

initial weights, "softmax" output units, a cross-entropy error function, and either very gentle learning (John Platt, personal communication) or a regularizer that penalizes the squared weights by an amount that is carefully chosen using a validation set. For comparison, nearest neighbor has a reported error rate (google MNIST) of 3.1% if all 60,000 training cases are used (which is extremely slow) and 4.4% if 20,000 are used. This can be reduced to 2.8% and 4.0% by using an L3 norm.

The only standard machine learning algorithm that outperforms the generative model is support vector machines which give error rates of 1.1% or 1.0%. But it is hard to see how support vector machines can make use of the domain-specific tricks, like weight-sharing and sub-sampling, which LeCun et.al. use to improve the performance of discriminative neural networks from 1.5% to 0.95%. There is no obvious reason why weight-sharing and sub-sampling cannot be used to reduce the error-rate of the generative model. Further improvements can always be achieved by averaging the opinions of multiple networks or by enhancing the training set with distorted data, but these techniques are available to all methods, though data enhancement can seriously slow-down methods that do not scale sub-linearly with the size of the training set.



Figure 9: Each row shows 10 samples from the generative model with a particular label clamped on. The top-level associative memory is run for 1000 iterations of alternating Gibbs sampling between samples.

8.2 Testing the network

One way to test the network is use a stochastic up-pass from the image to fix the binary states of the 500 units in the lower layer of the associative memory. With these states fixed, the label units are given initial real-valued activities of 0.1 and a few iterations of alternating Gibbs sampling are then used to activate the correct label unit. This method of testing gives error rates that are almost 1% higher than the rates reported above.

A better method is to first fix the binary states of the 500 units in the lower layer of the associative memory and to then turn on each of the label units in turn and compute the exact free energy of the resulting 510 component binary vector. Almost all the computation required is independent of which label unit is turned on [Teh and Hinton, 2001] and this method computes the exact conditional equilibrium distribution over labels instead of approximating it by Gibbs sampling which is what the previous method is doing. This method gives error rates that are about 0.5% higher than the ones quoted because of the stochastic decisions made in the up-pass. We can remove this noise in two ways. The simplest is to make the up-pass deterministic by using probabilities of activation in place of stochastic binary states. The second is to repeat the stochastic up-pass twenty times and average either the label probabilities or the label log probabilities over the twenty repetitions before picking the best one. The two types of average give almost identical results and these results are also very similar to using a deterministic up-pass, which was the method used for the reported results.

9 Looking into the mind of a neural network

To generate samples from the model, we perform alternating Gibbs sampling in the top-level associative memory until the Markov chain converges to the equilibrium distribution. Then



Figure 10: Each row shows 10 samples from the generative model with a particular label clamped on. The top-level associative memory is initialized by an up-pass from a random binary image in which each pixel is on with a probability of 0.5. The first column shows the results of a down-pass from this initial high-level state. Subsequent columns are produced by 20 iterations of alternating Gibbs sampling in the associative memory.

we use a sample from this distribution as input to the layers below and generate an image by a single down-pass through the generative connections. If we clamp the label units to a particular class during the Gibbs sampling we can see images from the model's class-conditional distributions. Figure 9 shows a sequence of images for each class that were generated by allowing 1000 iterations of Gibbs sampling between samples.

We can also initialize the state of the top two layers by providing a random binary image as input. Figure 10 shows how the class-conditional state of the associative memory then evolves when it is allowed to run freely, but with the label clamped. This internal state is "observed" by performing a down-pass every 20 iterations to see what the associative memory has in mind. This use of the word "mind" is not metaphorical. A mental state is the state of a hypothetical world in which a high-level internal representation would constitute veridical perception. That hypothetical world is what the figure shows.

10 Conclusion

The network in figure 6 has about as many parameters as 0.002 cubic millimeters of mouse cortex. Several hundred networks of this complexity would fit within a single voxel of a high resolution fMRI scan. Learning algorithms are getting better, but they still have a long way to go.

Acknowledgments

The ideas presented in this paper came from research collaborations with Terry Sejnowski, Yann LeCun, Jay McClelland, Radford Neal, Rich Zemel, Peter Dayan, Michael Jordan, Brendan Frey, Zoubin Ghahramani, Yee-Whye Teh, Max Welling, Simon Osindero and many other researchers. Andriy Mnih helped to prepare the manuscript. The research was supported by NSERC, the Gatsby Charitable Foundation, CFI and OIT. GEH holds a Canada Research Chair in machine learning.

References

- [Carreira-Perpinan and Hinton, 2005] M. A. Carreira-Perpinan and G. E. Hinton. On contrastive divergence learning. In *Artificial Intelligence and Statistics, 2005*, 2005.
- [Freund, 1995] Y. Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 12(2):256–285, 1995.
- [Friedman and Stuetzle, 1981] J.H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American Statistical Association*, 76:817–823, 1981.
- [Hinton and Sejnowski, 1986] G. E. Hinton and T. J. Sejnowski. Learning and relearning in Boltzmann machines. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel Distributed Processing: Volume 1: Foundations*, pages 282–317. MIT Press, Cambridge, 1986.
- [Hinton et al., 1995] G. E. Hinton, P. Dayan, B. J. Frey, and R. Neal. The wake-sleep algorithm for self-organizing neural networks. *Science*, 268:1158–1161, 1995.
- [Hinton, 2002] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1711–1800, 2002.
- [Jordan et al., 1999] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. In M. I. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1999.
- [Mayraz and Hinton, 2001] G. Mayraz and G. E. Hinton. Recognizing hand-written digits using hierarchical products of experts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:189–197, 2001.
- [Neal and Hinton, 1998] R. M. Neal and G. E. Hinton. A new view of the EM algorithm that justifies incremental, sparse and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.
- [Neal, 1992] R. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113, 1992.
- [Pearl, 1988] J. Pearl. *Probabilistic Inference in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.
- [Roth and Black, 2005] S. Roth and M. J. Black. Fields of experts: A framework for learning image priors. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2005.
- [Rumelhart et al., 1986] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.
- [Teh and Hinton, 2001] Y.W. Teh and G. E. Hinton. Rate-coded restricted Boltzmann machines for face recognition. In *Advances in Neural Information Processing Systems*, volume 13, 2001.
- [Teh et al., 2003] Y.W. Teh, M. Welling, S. Osindero, and G. E. Hinton. Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260, Dec 2003.
- [Welling et al., 2003] M. Welling, G. Hinton, and S. Osindero. Learning sparse topographic representations with products of Student-t distributions. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1359–1366. MIT Press, Cambridge, MA, 2003.
- [Zemel and Hinton, 1995] R. S. Zemel and G. E. Hinton. Learning population codes by minimizing description length. *Neural Computation*, 7:549–564, 1995.

NEXT SECTION