

Unsupervised Learning of Mutagenesis Molecules Structure Based on an Evolutionary-Based Features Selection in DARA

Rayner Alfred, Irwansah Amran, Leau Yu Beng, and Tan Soo Fun

School of Engineering and Information Technology, Universiti Malaysia Sabah, Jalan UMS,
88400, Kota Kinabalu, Sabah, Malaysia
{ralfred, irwansah, lybeng, soofun}@ums.edu.my

Abstract. The importance of selecting relevant features for data modeling has been recognized already in machine learning. This paper discusses the application of an evolutionary-based feature selection method in order to generate input data for unsupervised learning in DARA (Dynamic Aggregation of Relational Attributes). The feature selection process which is based on the evolutionary algorithm is applied in order to improve the descriptive accuracy of the DARA (Dynamic Aggregation of Relational Attributes) algorithm. The DARA algorithm is designed to summarize data stored in the non-target tables by clustering them into groups, where multiple records stored in non-target tables correspond to a single record stored in a target table. This paper addresses the issue of optimizing the feature selection process to select relevant set of features for the DARA algorithm by using an evolutionary algorithm, which includes the evaluation of several scoring measures used as fitness functions to find the best set of relevant features. The results show the unsupervised learning in DARA can be improved by selecting a set of relevant features based on the specified fitness function which includes the measures of the dispersion and purity of the clusters produced.

Keywords: Feature Selection, Data Summarization, Clustering, Genetic Algorithm, Data Reduction.

1 Introduction

Learning relational data is difficult due to the existence of one-to-many relationships between records stored in multiple tables. Structured data such as data stored in a relational database that have one-to-many relationships between records stored in the target and non-target tables can be transformed into a single table. This process is known as propositionalization [1]. DARA (Dynamic Aggregation of Relational Attributes) algorithm is capable to transform relational data that have one-to-many relationships between records stored in the target and non-target tables, into vectors of features [2]. In order to assist the clustering process, a technique borrowed from information retrieval theory has been used to construct vectors of features that represent records stored in non-target tables (see Fig. 1). The approach is based on the idea that,

for each unique record stored in non-target tables, there is a vector of patterns that represents this record. With this representation of data, it is possible to compare records in order to cluster them. As a result, clustering can then be applied to summarize data stored in a relational database following the principles of information retrieval theory [3].

The application of feature construction has proved an improvement in the descriptive accuracy of the summarized data from relational domain [22]. However, in this paper, we apply a genetic based feature selection as a pre-processing step in order to further enhance our data summarization method called DARA [2, 23]. It is expected that the predictive accuracy of the classification task for data stored in the target table can be improved by improving the descriptive accuracy of the data summarization for the data stored in the non-target tables, as shown in Fig. 1. We propose that the summarized data to be appended into the list of attributes in the target table as one of the features considered for the classification task as shown in Fig. 1.

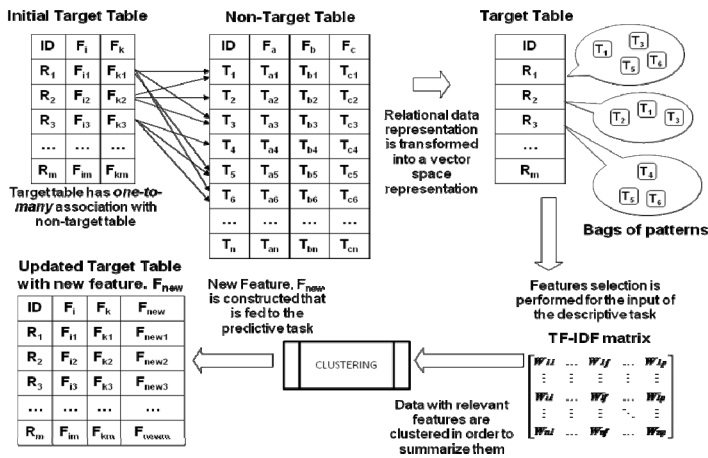


Fig. 1. A data transformation process for data stored in multiple tables with one-to-many relations into a vector space data representation

Section 2 describes the process of transforming a relational data into a vector representation by using the DARA algorithm. Section 3 provides an overview of the more general concept of feature selection that can be applied to improve the accuracy estimations of learning tasks. Section 4 describes a genetic-based (i.e., evolutionary) feature selection algorithm that is applied to reduce the dimensionality of the data for the purpose of summarizing data stored in the non-target tables using the DARA algorithm. This genetic-based feature selection algorithm selects features to characterize each unique object stored in the non-target table. The performance accuracy of the J48 classifier for the classification tasks using these summarized data will be presented in Section 5. Finally, this paper is concluded in Section 6 by presenting some of the findings obtained from the experiments performed in this study.

2 Data Transformation Using DARA

In this section, we first describe the concept of multi-relational setting for data stored in a relational database. Then, we describe how a single object stored in a target table that is associated with many objects stored in a non-target table can be represented in a vector space model.

2.1 Multi-relational Setting

In a relational database, a single record, R_i , stored in the target table can be associated with other records stored in the non-target table, as shown in Fig. 1. Let R denote a set of m records stored in the target table and let S denote a set of n records ($T_1, T_2, T_3, \dots, T_n$), stored in the non-target table. Let S_i be a subset of S , $S_i \subseteq S$, associated through a foreign key with a single record R_a stored in the target table, where $R_a \in R$. Thus, the association of these records can be described as $R_a \leftarrow S_i$. In this case, we have a single record stored in the target table that is associated with multiple records stored in the non-target table. The records stored in the non-target table that correspond to a particular record stored in the target table can be represented as vectors of features. As a result, based on the vector space model [4], a unique record stored in non-target table can be represented as a vector of features. In other words, a particular record stored in the target table that is related to several records stored in the non-target table can be represented as a bag of patterns, i.e., by the patterns it contains and their frequency, regardless of their order. The bag of patterns is defined as follows:

Definition 1. In a bag of patterns representation, each unique record stored in the non-target table is represented by the set of its pattern and the pattern frequencies.

This definition states that a single record data in a multi-relational setting can be described as a collection of individuals and the induced rules generalize over the individuals, mapping them to a class. For instance, individual-centered domains include classification problems in molecular biology where the individuals are molecules.

2.2 Relational Data Representation in a Vector Space Model

In this subsection, we describe the representation of data for objects stored in multiple tables with one-to-many relations. Let DB be a database consisting of n objects. Let $R := \{R_1, \dots, R_m\}$ be the set of different representations existing for objects in DB and each object may have zero or more than one representation of each R_i , such that $|R_i| \geq 0$, where $i = 1, \dots, m$. Each object $O_i \in DB$, where $i = 1, \dots, n$ can be described by maximally m different representations with each representation has its frequency,

$$O_i := \{R_1(O_i):|R_1(O_i)|:|Ob(R_1)|, \dots, R_m(O_i):|R_m(O_i)|:|Ob(R_m)|\} \quad (1)$$

where $R_j(O_i)$ represents the j -th representation in the i -th object and $|R_j(O_i)|$ represents the frequency of the j -th representation in the i -th object, and finally $|Ob(R_j)|$ represents

the frequency of object with j -th representation. If all different representations exist for O_i , then the total different representations for O_i is $|O_i| = m$ else $|O_i| < m$.

In relational instance-based learning, the distance measures are defined based on the attribute's type [5] and the distance between two objects is based on the minimum distance between pair of instances from the two objects. In our approach, we apply the vector-space model [6] to represent each object. In this model, each object O_i is considered as a vector in the representation-space. In particular, we employed the *rf-iof* term weighting model borrowed from [6], where in which each object O_i , $i = 1, \dots, n$ can be represented as

$$(rf_1 \cdot \log(n/of_1), rf_2 \cdot \log(n/of_2), \dots, rf_m \cdot \log(n/of_m)) \quad (2).$$

where rf_j is the frequency of the j -th representation in the object, of_j is the number of objects that contain the j -th representation and n is the number of objects. To account for objects of different lengths, the length of each object vector is normalized so that it is of unit length ($\|O_{rfiof}\| = 1$), that is each object is a vector on the unit hypersphere. In this experiment, we will assume that the vector representation for each object has been weighted using *rf-iof* and it has been normalized so that it is of unit length. In the vector-space model, the cosine similarity is the most commonly used method to compute the similarity between two objects O_i and O_j , $sim(O_i, O_j)$, which is defined as $cos(O_i, O_j) = O_i \cdot O_j / (\|O_i\| \cdot \|O_j\|)$. The cosine formula can be simplified to $cos(O_i, O_j) = O_i \cdot O_j$, when the record vectors are of unit length. This measure becomes one if the records are identical, and zero if there is nothing in common between them. The idea of our approach is to transform the data representation for all objects in a multi-relational environment into a vector space model and find the similarity distance measures for all objects to cluster them. These objects then are grouped based on the similarity of their characteristics, taking into account all possible representations and the frequency of each representation for all objects.

3 Feature Selection in DARA

Features used to describe instances are not necessarily all relevant and beneficial for inductive learning. In fact, a large number of features may slow down the induction process while giving similar results as obtained with a selected feature subset. The problem of feature subset selection can be considered as an optimization problem, in which the whole search space for optimization contains 2^F possible subsets of features, where F is the number of features.

3.1 Approaches to Feature Subset Selection

There are two main approaches to feature subset selection used in machine learning: the *Filtering* approach [7] and *Wrapper* approach [8]. Filtering approaches for feature subset selection attempt to assess the features and their merits using the data available independently of the learning method that will use the selected features. The decision

tree filter, FOCUS, RELIEF and its variants are some of the widely known Filtering algorithms. The main objective of the decision tree filter and FOCUS filter is to find the minimum number of features that can effectively categorize the given objects. The RELIEF algorithm is an instance-based filter that handles instance spaces with no more than two class labels. All these filtering algorithms evaluate the features independently of the classifiers that use them. Some of the statistical and information theoretic measures used such as *information gain* and *cross-entropy* can be used to weigh the relevance of the features [9]. Other measures that capture the relationship of a feature with the target concept include probabilistic distance measure, probabilistic dependence measures, inter-class measures and information theoretic measures such as the *entropy* measure [10].

On the other hand, wrapper approaches for feature subset selection attempt to select a feature subset using the evaluation function based on the same learning algorithm that will be used for learning the domain represented with the selected features. The process is repeated until no improvement is made or addition/deletion of new features reduces the accuracy of the target learner.

3.2 Search Techniques

In the context of search techniques, feature selection algorithms can be divided into two categories: *sequential* algorithms and *randomized* algorithms. The *sequential* algorithm was first introduced by Devijver and Kitler [9] and this algorithm either starts with the full or empty feature set, and proceeds by greedily adding or removing features. *Sequential backward elimination* (SBE) starts with the full set of n features, and considers each of the n subsets of $n - 1$ features by removing each feature once. From these n subsets, the one giving the highest classification performance is chosen. The process is then repeated for the set of $n - 1$ remaining features and so on until some termination criterion is fulfilled. Similarly, *sequential forward selection* (SFS) can be defined, where the initial state is the empty set of features, and features are greedily added. The feature is added at each step that most increases the performance of the learner. It is not clear whether backward elimination or forward selection will perform better on a data set with no prior information on feature correlation.

On the other hand, *randomized* search algorithms differ from sequential algorithms in several aspects. Instead of starting with the full or empty set of features, these algorithms generally start somewhere in between, by generating a random subset of the features. The search then proceeds by randomly adapting the initial solution until a stopping criterion is fulfilled. Techniques based on simulated annealing or hill-climbing start from an initial guess and apply random changes until the solution does not change any more, or a fixed number of iterations have elapsed [11]. However, these methods are prone to getting stuck in local extrema. To solve this disadvantage, Holland [12] introduced population-based techniques, such as genetic algorithms (GA). GA operates on a whole set of possible solutions that are selected from the population instead of one solution. Using an iterative approach that mimics nature's evolutionary principle, new generations of the initial population are created using a set of three operators:

- a) **Selection:** A number of good individuals is chosen and copied from one generation into the next (survival)
- b) **Crossover:** The parameters of two individuals are combined, to produce a new individual that is copied to the next generation.
- c) **Mutation:** A certain parameter of a chosen individual is adapted (mutated) and the resulting individual is copied to the next generation

In this way, a GA can be viewed as a stochastic iterative sampling procedure. GAs have been particularly useful when faced with complex optimization problems. In the original GA, individuals were represented as a string of bits, yet later on strings representing any kind of parameter type have been used [13, 14]. In this paper, we apply the genetic based filtering approach to select relevant features for the DARA algorithm to summarize data stored in a relational database.

4 A Genetic Based Feature Selection for Data Summarization

A Genetic Algorithm (GA) is generally a computational abstraction of biological evolution approach that can be very useful to solve some searching and optimization problems [12, 15]. GA process applies a series of biological genetic operators such as selection, crossover and mutation to a population of elements called chromosome. Chromosomes represent possible solutions to the searching and optimization problem. Firstly, a random population which represents different points in the search space is created. An objective and fitness function will be associated with each chromosome that represents the level of quality of the chromosome. Based on the principle of the survival of the fittest, several chromosomes are selected and each is assigned a number of copies which will be put into the mating pool. Crossover and mutation which are biologically inspired operators are applied to these strings to produce a new generation of strings. The combined process of selection, crossover and mutation repeated continuously for a fixed number of generations or till a termination condition is satisfied.

One of the ways to improve the data summarization results using clustering method is by identifying and removing the less important features in the dataset. In this paper we apply the string of bits to represent individual solution for our data summarization technique. As mentioned in Section 2, DARA algorithm transforms relational data into vectors of features as shown in Fig. 2. In Fig. 2, W_{if} represents the f th feature for the i th record. For instance, suppose we have 10 features, $p=10$, for record R_1 as $\{W_{11}, W_{12}, W_{13}, W_{14}, W_{15}, W_{16}, W_{17}, W_{18}, W_{19}, W_{110}\}$. Then, the possible chromosome that we can have will be $\{1,1,1,1,1,1,1,1,1,1\}$ in order to include all features for the clustering purposes. However, in case we have a chromosome with a string of bits as $\{0,1,1,0,1,1,1,0,1,1\}$, then only features $W_{12}, W_{13}, W_{15}, W_{16}, W_{17}, W_{19}$ and W_{110} will be used for clustering purposes.

$$\begin{bmatrix} W_{11} & \dots & W_{1f} & \dots & W_{1p} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ W_{i1} & \dots & W_{if} & \dots & W_{ip} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{n1} & \dots & W_{nf} & \dots & W_{np} \end{bmatrix}$$

Fig. 2. Relational data represented as vectors of features

The genetic algorithm will be used to get the best subset of features based on the cluster's dispersion and purity. In other words, cluster dispersion (e.g., Davies-Boulding Index, DBI [16]) and cluster purity (e.g., Gini Index [17,18]) are used as the fitness function in order to evaluate individual chromosome.

5 Experimental Setup and Results

In this experiment, we employ an algorithm, called DARA that converts the dataset representation in relational model into a space vector model and use a distanced-based method to group objects with multiple representations occurrence. With DARA algorithm, all representations of two objects are taken into consideration in measuring the similarity between these two objects. The DARA algorithm can also be seen as an aggregation function for multiple instances of an object, and is coupled with a few classifiers obtained from WEKA [19] that includes the *C4.5* classifier (J48 in WEKA), *RandomForest*, *HyperPipes*, *LogitBoost* and *SMO*, as induction algorithms that are run on the DARA's transformed data representation. We then evaluate the effectiveness of each data transformation with respect to these classifiers. All experiments with DARA and classifiers were performed using a leave-one-out cross validation estimation. We chose well-known dataset, Mutagenesis [21].

The mutagenesis data [21] describes 188 molecules falling in two classes, mutagenic (active) and non-mutagenic (inactive); 125 of these molecules are mutagenic. The description consists of the atoms and bonds that make up the compound. Thus, a molecule is described by listing its atoms *atom(AtomID, Element, Type, Charge)*, and the bonds *bond(Atom1, Atom2, BondType)* between atoms. In this experiment, we use only one set of background knowledge: B1. In B1, the atoms in the molecule are given, as well as the bonds between them; the type of each bond is given as well as the element and type of each atom. The table for B1 has the schema *Molecule(ID, ATOM1, ATOM2, TYPE_ATOM1, TYPE_ATOM2, BOND_TYPE)*, where each molecule is described over several rows, listing all pairs of atoms with a bond, and the type of each atom, and the type of bond between them.

In the selection process, we apply the roulette-wheel selection in order to select chromosomes which will be used to generate a new population. Other parameters include crossover probability, $p_c = 0.25$, mutation probability, $p_m = 0.10$, population size is set to 20 and the number of generations is set to 50. Table 1 shows the

predictive accuracies for *J48*, *RandomForest* and *LogitBoost* classifiers for the transformed data with genetic based feature selection method are higher compared to the transformed data without feature selection.

Table 1. Predictive accuracy results based on 10-fold cross-validation

Classifier	Without Clustering	Clustering Without Feature Selection	Clustering with Genetic Based Feature Selection
J48	78.2 %	80.3 %	84.0 %
RandomForest	80.9 %	82.4 %	84.6 %
HyperPipes	77.1 %	77.7 %	77.7 %
LogitBoost	83.5 %	83.5 %	84.6 %
SMO	84.0 %	84.6 %	84.6 %

6 Conclusion

In the process of learning a given target table that has a one-to-many relationship with another non-target table, a data summarization process can be performed to summarize records stored in the non-target table that correspond to records stored in the target table. In the case of a classification task, part of the data stored in the non-target table can be summarized based on the class label or without the class label. To summarize the non-target table, a record can be represented as a vector of features. In order to get a better clustering result, a genetic based feature selection method is applied. These objects are then clustered or summarized based on the basis of those selected features. In this work, a genetic-based feature selection algorithm has been proposed to select features that best represent the characteristics of records that have multiple instances stored in the non-target table.

Unlike other approaches to feature selection, this paper has outlined the usage of feature selection to improve the descriptive accuracy of the proposed data summarization approach (DARA). Most feature selection methods deal with problems to find the best set of selected features that can improve the predictive accuracy of a classification task. This paper has described how feature selection can be used in the data summarization process for relational data to get better descriptive accuracy, and indirectly improve the predictive accuracy of a classification task. It is shown in the experimental results that the quality of summarized data is directly influenced by the selected features used to characterize individual record. The results of the evaluation of the genetic-based feature selection algorithm show that the data summarization results can be improved by selecting relevant features.

References

1. Kramer, S., Lavrac, N., Flach, P.: Propositionalisation Approaches to Relational Data Mining. In: Deroski, S., Lavrac, N. (eds.) *Relational Data Mining*. Springer (2001)

2. Alfred, R., Kazakov, D.: Data Summarization Approach to Relational Domain Learning Based on Frequent Pattern to Support the Development of Decision Making. In: Li, X., Zai'ane, O.R., Li, Z.-h. (eds.) ADMA 2006. LNCS (LNAI), vol. 4093, pp. 889–898. Springer, Heidelberg (2006)
3. van Rijsbergen, C.J.: Information Retrieval. Butterworth (1979)
4. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill Book Company (1984)
5. Horvath, T., Wrobel, S., Bohnbeck, U.: Relational instance-based learning with lists and terms. *Machine Learning* 43(1), 53–80 (2001)
6. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. *Communications of the ACM* 18, 613–620 (1975)
7. Bensusan, H., Kuscü, I.: Constructive Induction using Genetic Programming. In: Fogarty, T., Venturini, G. (eds.) Evolutionary Computing and Machine Learning Workshop, ICML 1996 (1996)
8. Aha, D.W., Bankert, R.L.: Feature Selection for Case-Based Classification of Cloud Types. In: Proceedings of the AAAI 1994 Workshop on Case-Based Reasoning. AAAI Press, Seattle (1994)
9. Devijver, P.A., Kittler, J.V.: Pattern Recognition: A Statistical Approach. Prentice Hall (1982)
10. Doak, J.: An Evaluation of Feature Selection Methods and Their Application to Computer Security. Technical report, UC Davis Department of Computer Science (1992)
11. Skalak, D.B.: Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. In: ICML, pp. 293–301 (1994)
12. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press (1975)
13. Fernando, E.B., Monique, M.S., Freitas, A.A., Nievola, J.C.: Genetic Programming for Attribute Construction in Data Mining. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 384–393. Springer, Heidelberg (2003)
14. Kudo, M., Sklansky, J.: Comparison of Algorithms That Select Features for Pattern Classifiers. *Pattern Recognition* 33(1), 25–41 (2000)
15. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc. (1989)
16. Davies, D.L., Bouldin, D.W.: A Cluster Separation Measure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 224–227 (1979)
17. Breiman, L., Friedman, J., Olshen, T., Stone, C.: Classification and Regression Trees. Wadsworth International, California (1984)
18. Raileanu, L.E., Stoffel, K.: Theoretical Comparison Between the Gini Index and Information Gain Criteria. *Annals of Mathematics and Artificial Intelligence* 41(1), 77–93 (2004)
19. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufman (1999)
20. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Los Altos (1993)
21. Srinivasan, A., Muggleton, S.H., Sternberg, M.J.E., King, R.D.: Theories for mutagenicity: A Study in first-order and feature-based induction. *Artificial Intelligence* 85 (1996)
22. Alfred, R.: Feature transformation: A genetic-based feature construction method for data summarization. *Computational Intelligence* 26(3), 337–357 (2010)
23. Alfred, R.: The Study of Dynamic Aggregation of Relational Attributes on Relational Data Mining. In: Alhajj, R., Gao, H., Li, X., Li, J., Zai'ane, O.R. (eds.) ADMA 2007. LNCS (LNAI), vol. 4632, pp. 214–226. Springer, Heidelberg (2007)