USING BI-SETS THAT CHARACTERIZE BI-PARTITIONS AS FEATURES FOR CLASSIFICATION: AN APPLICATION TO MICROARRAY DATA ANALYSIS

Ivica Slavkov[1], Ruggero Pensa[2], Sašo Džeroski[1]
1. Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39, SI-1000, Ljubljana, Slovenia
{Ivica.Slavkov,Saso.Dzeroski}@ijs.si
2. INSA Lyon, LIRIS CNRS,
UMR 5205 F-69621, Villeurbanne cedex, France
Ruggero.Pensa@insa-lyon.fr

ABSTRACT

As part of the efforts for building a unified Inductive Databases (IDBs) framework, an important step would be to find a way to combine discovered local patterns from the data with global models of predictive nature. In this paper, we investigate the possibility of using bi-sets (local patterns) as features during classification. When searching for bi-sets from Boolean data, despite reasonable frequency constraints, a large number of sets are usually generated. In order to discern which of these bi-sets could be potentially useful as features for classification, we are using a scoring function which includes as parameters the bi-sets coverage and size. After a feature construction process, we perform an experimental evaluation on Huntington's disesase (HD) microarray data. We apply Predictive Clustering Trees for the problem of distinguishinh between HD and healthy subjects and also for determining the stage of the development of the disease.

1 INTRODUCTION

A recent emerging area in data mining are Inductive Databases (IDBs), which offer a database perspective on the process of knowledge discovery. IDBs contain not only data, but also patterns. They can be either local patterns (e.g., frequent itemsets), which are of descriptive nature, or global models (e.g., decision trees), which are generally of predictive nature. The idea behind IDBs is that data and patterns (models) are handled in the same way and the user can query and manipulate the patterns (models) of interest by means of a query language [4] As part of this general framework, in this paper we are exploring the use of bi-sets (as local patterns) for classification purposes (as global models). We first perform a feature construction procedure and test the performance of these features by constructing decision trees. The data that was used for testing is Huntington's disease microarray data, which was previously discretized i.e. binarized.

This paper is organized as follows. In Section 2 we describe the dataset and the discretization technique that was used. Section 3 gives an overview of the methodology, including a short definition of bi-clusters and bi-sets followed by the feature scoring and selection process. Section 4 concerns the experimental design and results. Finally, in Section 5 we give conclusion and discussion of the obtained results.

2 DATA DESCRIPTION AND PREPROCESSING

2.1 Huntington's disease microarray data

Huntington's disease (HD) is an autosomal dominant neurodegenerative disorder characterized by progressive motor impairment, cognitive decline, and various psychiatric symptoms, with the typical age of onset in the third to fifth decades [1]. It is caused by the expansion of an unstable triplet repeat in *huntingtin* gene, which encodes for the ubiquitously distributed huntingtin protein. The microarray data is from Slovene patients and it consisted of three different types of samples. The first two types are samples from HD patients, which are in two distinct stages of the disease: presymptomatic and symptomatic. The third type are control (healthy) subjects. All together, there were 24 samples of which: 9 presymptomatic, 5 symtomatic (14 HD) and 10 control. For each sample the expression levels for 54.675 probes from an Affymetrix HG.U133A 2.0 chip were measured. The expression levels were obtained by using the MAS 5.0 software.

2.2 Data preprocessing

The first step of the data preprocessing was filtering out the measured microarray transcripts which could be considered as unreliable. This was a simple elimination of genes, which under all experimental conditions had signal strength less than 100. The number of genes was reduced to 8910. Furthermore, there was need for discretization of the numerical data and converting it into a Boolean format.

After the discretization each gene had three possible values: overexpressed, normaly expressed and underexpressed. We had two threshold for discretization of the values of each gene. Values under threshold 1 were considered as underexpressed, values above threshold 2 were overexpressed and the rest were normally expressed. We tried different values of the two threshold and discretized the data several times. We assessed which threshold values were optimal by using the score described in [8].

3 METHODOLOGY

3.1 Predictive Clustering Trees (PCTs)

Decision trees are usually considered for classification purposes. Each tree consists of three elements: internal nodes, branches and leaves. The internal nodes are labeled with some attribute (variable name) and each branch is labeled with a predicate that can be applied to the attribute associated with the parent node. The leaves however, are labeled with a class. Following the branches from the root to a leaf gives sufficient conditions for classification (Figure 1).

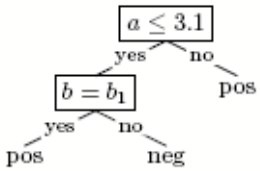

Figure 1: *A typical classification tree with classes "pos" and "neg"*

An alternative view of decision trees is that they correspond to the concept of hierarchical clustering [2,10]. Each node (and leaf) corresponds to a cluster and the tree as a whole represents a kind of taxonomy or hierarchy. Thus we can use the concept of **TDIDT** (Top-Down Induction of Decision Trees) for inducing clustering trees. We assume that two types of functions exist. A prototype function, which is used to get the best description of the members of a cluster, and a distance function for measuring the distance between prototypes and also between members of the cluster and the prototype. This leads to a simple method for building trees that allow prediction of multiple target variables at once.

When inducing the clustering tree the TDICT (Top-Down Induction of Clustering Trees) algorithm uses as a heuristic the minimization of intra-cluster variance (and maximization of inter-cluster variance). The minimization of the intra-cluster variance means minimizing the average distance between the members of the cluster and the prototype, which describes it. Maximization of the inter-cluster variance maximizes the distance between the prototypes. At the end we get a clustering tree in which the top-level node corresponds to one cluster containing all of the data, which is recursively partitioned into smaller clusters while moving down the tree. The leaves of the clustering tree are clusters, but they also store information about the cluster prototype. Because in essence the prototype describes the cluster, it can also be considered as a prediction of the values in that cluster with a certain amount of error.

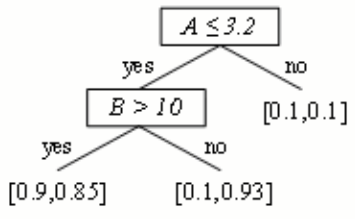

Figure 2: *A multi-objective regression tree with two predicted values in the leaves*

When PCTs are used for prediction of one target variable, they are actually regular decision trees. But their real advantage is when predicting multiple variables at the same time (multi objective classification or regression) shown in Figure 2.

3.2 Bi-sets and bi-clustering

When mining for local patterns in Boolean data (Figure 3) it is interesting to see not only which items appear together often, but also which items occur together in which situation (i.e. object). The sets which give us this information are called bi-sets[7]. One such bi-set from Figure 2 would be (*{o1; o2};{i1; i2}*).

	Items		
	i_1	i_2	i_3
o_1	1	1	1
o_2	1	1	0
o_3	1	0	1
o_4	1	0	0
o_5	0	1	0

Figure 3: *Boolean table*

Bi-clustering has been previously used for mining microarray data [3] and for discovering co-expressed sets of genes. The bi-clustering algorithms are performing clustering on the data by taking into account the rows as well as the columns. Esentially, given a set of m objects and n items (i.e. an m × n matrix), the bi-clustering algorithm generates subsets of objects that exhibit similar behavior across a subset of items, or vice versa.

3.1 Feature construction for classification

In order to use the information from the bi-sets and bi-clusters (local patterns) for constructing classifiers a process of feature construction is needed. The algorithm for feature construction [9] is given as:

Input: Boolean labeled data **D,** maximum number of features **f**
Output: List of features feature for classification **Lf**

1. **S**= empty; score of each bi-set
2. **Bp=** generate bi-partitions from labeled data **D**
3. **Bs** = generate bi-sets from labeled data **D**
4. ***Repeat***
5. **bs**=first bi-set from **Bs**
6. **Bs**=**Bs****bs**
7. **s**=score(**bs, Bp**)
8. **S=S** U **s**
9. ***Until*** **Bs** is empty
10. **S**=sort(**S**)
11. ***Repeat***
12. **l**=extract genes from concept **bs** with score **S**[**l**]
13. **Lf=Lf** U **l**
14. **l--**
15. ***Until*** **f** =/=0
16. Return **Lf**

The whole process of feature construction begins by generating the bi-partitions from the data. When generating the bi-partitions we constrain one of the dimesions of clustering to match to the classes. This means that the number of clusters is identical to the number of classes and members of one cluster are the same type of samples. Furthermore, we mine for all of the bi-sets from the Boolean data. We asses (score) each of the bi-set by taking into account its' relationship with each of the bi-clusters [9]. We have four main parameters which we take into account:

1. Coverage ratio *C:* number of covered class examples/number of class examples
2. Object Confidence *OC*: number of covered objects from the bi-partition /total number of covered objects by the bi-set
3. Item Confidence *PC:* number of covered items from the bi-partition /total number of covered items by the bi-set
4. Feature size *N* (i.e., maximum number of genes, N)

As the three parameters are between 0 and 1, we can calculate the score by:

$$score = \frac{\sqrt[3]{C \times OC \times PC}}{N};$$

After scoring each bi-set in terms of how good it describes a bi-cluster and sorting the bi sets by the score, we can extract the top f features where f is user defined. As features we consider the co-expressed genes extracted from the top scoring bi-sets.

4 EXPERIMENTAL DESIGN AND RESULTS

As a classification model we used ordinary decision trees and the previously described Predictive Clustering Trees (PCTs), which can predict several class values at the same time (Figure 4).

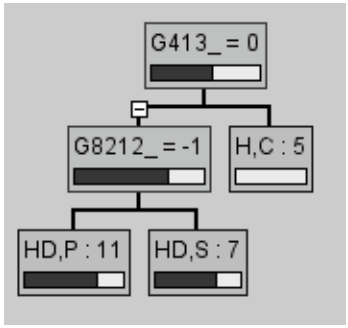

Figure 4: *Example of the constructed PCT*

Due to the small sample size we performed a leave-one-out cross validation We compared our results against those obtained by constructing predictive models on the original, numeric data. First we selected the "relevant" features from the training data and then we performed feature construction for the full set. We then built a model from the training data and tested the model on the left-out sample. The results are summarized in Table 1.

Type of model	Class	Class values	LOO-CV accuracy for numerical data	LOO-CV accuracy for discretized data
Classification tree	Huntington	{HD,C}	51%	75%
Classification tree	Stage	{P,S,C}	44%	62,5%
Predictive Clustering Tree	Huntington Stage	{HD,C} {P,S,C}	74% 74%	79,1% 62,5%

Table 1: Experimental results from the original and from the discretized data with features f=12

When constructing the features we selected number of features f=12. From the results the following can be seen: For ordinary decision trees, when using the discretized data and the feature construction process, there is a significant improvement in accuracy compared to the numerical data. The same is true when constructing PCTs, but only for the class "Huntington". For the class "Stage" the accuracy is worse compared to the numerical data, but it is the same as when constructing ordinary decision trees. This could mean

that ordinary decision trees have a significant benefit when used in conjuction with the bi-sets feature construction process, while PCTs do not have the same improvement (leverage) of performance.

The set of genes that were identified as important when using the numerical data was different than the genes used as features from the discretized data. This was expected due to the different type of modeling the data, but also because of the biological complexity involved. The number of genes which are interconnected between themselves and have a role in the genesis of the disease is numerous. That is why we searched for the role of these genes in the Gene Ontology (GO) database. Although they were not the same set of genes, there was an overlap in some of their functions, which by previous studies were conected to the mechanism of Huntington's disease. This included disturbed transcriptional activities [5,11] and disturbed protein functioning [6].

5 CONCLUSION

An important aspect when working with high-dimensional data is selecting the features which are most informative (important). In this paper we are attempting to demonstrate the possibility of using a feature construction (selection) process for microarray data, which combines local pattern mining with global predictive modeling. The results showed that there could be two possible benefits when using the bi-sets feature construction process: first, the significant reduction of features (in this case from 8910 to 12) and second, the improvement of accuracy.

This scenario for analysis of microarray data can be further expanded towards any type of data. Further work would include testing this type of analysis to other type of data besides microarray and also in the direction of exploring other types of scenarios for analysis which combine different types of local patterns (association rules, frequent itemset) with predictive models.

References

[1] Borovecki et al. : Genome-wide expression profiling of human blood reveals biomarkers for Huntington's disease. In Proceedings of the National Academy of Sciences of the USA, August 2 2002, vol 102., no 31, p 11023-11028

[2] H. Blockeel, L. De Raedt and J. Ramon. Top-down induction of clustering trees. In Proceedings of the 15th International Conference on Machine Learning, p.55-63, 1998.

[3] Y. Cheng, G. M. Church. Biclustering of Expression Data Source. Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology , p. 93 – 103, 2000

[4] L. De Raedt. A perspective on inductive databases. SIGKDD Explorations, 4(2):69-77, 2002.

[5] Dunah AW, Jeong H, Griffin A et al. Sp1 and TAFII130 transcriptional activity disrupted in early huntington's disease. Science, 2002; 296: 2238-43.

[6] Harjes P, Wanker EE. The hunt for huntingtin function: interaction partners tell many different stories. Trends Biochem Sci. 2003; 28 (8): 425-33.

[7] R. Pensa, J-F. Boulicaut. Boolean property encoding for local set pattern discovery: an application to gene expression data analysis. Local Pattern Detection. Springer-Verlag LNAI 3539, p. 115-134, 2005.

[8] R. Pensa, C. Leschi, J. Besson, J-F. Boulicaut. Assessment of discretization techniques for relevant pattern discovery from gene expression data. Proceedings of the 4th ACM SIGKDD Workshop on Data Mining in Bioinformatics BIOKDD'04 , 2004. pp. 24-30.

[9] R. Pensa, J-F. Boulicaut. From local pattern mining to relevant bi-cluster characterization. Proceedings of the 6th International Symposium on Intelligent Data Analysis IDA 2005, Madrid, Spain, September 8-10, 2005. Springer-Verlag LNCS Volume 3646, A. F. Famili, J.M. Pena, A. Siebes, J. Kok (Eds.), pp. 293-304, 2005

[10] J. Struyf and S. Dzeroski, Constraint based induction of multi-objective regression trees. In proceedings of the 4th International Workshop on Knowledge Discovery in Inductive Databases, pages 110-121. 2005.

[11] Sugars KL, Rubinsztein DC. Transcriptional abnormalities in Huntington disease. Trends in Genetics, 2003; 19: 233-238.