2008 20th IEEE International Conference on Tools with Artificial Intelligence

RUSE-WARMR: Rule Selection for Classifier Induction in Multi-Relational Data-Sets

Carlos Abreu Ferreira
ISEP - Institute of Engineering of Porto and LIAAD- INESC LA, University of Porto
cferreira@liaad.up.pt

João Gama
LIAAD- INESC LA, University of Porto
jgama@liaad.up.pt

Vítor Santos Costa
CRACS- INESC LA, University of Porto
vsc@dcc.fc.up.pt

Abstract

One of the major challenges in knowledge discovery is how to extract meaningful and useful knowledge from the complex structured data that one finds in Scientific and Technological applications. One approach is to explore the logic relations in the database and using, say, an Inductive Logic Programming (ILP) algorithm find descriptive and expressive patterns. These patterns can then be used as features to characterize the target concept. The effectiveness of these algorithms depends both upon the algorithm we use to generate the patterns and upon the classifier. Rule mining provides an excellent framework for efficiently *mining the interesting patterns that are relevant. We propose a novel method to select discriminative patterns and evaluate the effectiveness of this method on a complex discovery application of practical interest.*

1. Introduction

One of major challenges in knowledge discovery is how to extract meaningful and useful knowledge from the complex structured data that one finds in Scientific and Technological applications. Such data may be structured, may stem from a number of very different sources with different reliability, and may be incomplete. We may have to deal with both huge amounts of data on some concepts and little data on other concepts, within the same dataset.

Work in this research area such as RSD [18], SAYU [19], nFOIL [12], kFOIL [11] suggests that an effective approach to tackle such a problem is what is called as *propositionalization*, where one searches for properties of interest in the data and uses the huge amount of research in feature-based learning to build a model for classification or descriptive purposes. More precisely, we explore the logic relations in the database and using an, say, Inductive Logic Programming (ILP) algorithm we find descriptive and expressive patterns. These patterns will be used as features to characterize the target concept. These patterns will then be used on their own or together with features originally known about the target concept, as input to a propositional learner or toolkit.

The effectiveness of these algorithms depends both upon the algorithm we use to generate the patterns and upon the classifier. There is an extensive body of work on propositional classifiers, so much of the novel contribution in this area results from addressing the first issue. The problem here is that the number of patterns we can generate from most interesting and powerful languages grows exponentially. On the one hand, if all these patterns are fed to the propositional learner, the learning task becomes very vulnerable to over-fitting. On the other hand, if we constrain the patterns, we risk to miss important patterns that may be extremely useful to the learning task.

The RUSE algorithm is motivated by the excellent results of Inductive Logic Programming pattern miners, such as WARMR, namely in terms of efficiency and robustness. In contrast to systems such as nFOIL or SAYU, where the pattern mining task is tightly integrated with classifier construction, and therefore is heavily dependent on the training examples, rule mining provides an excellent framework for *efficiently* mining the interesting patterns that are relevant on their own and that can scale very effectively on large datasets.

The contributions of this work are therefore:

- We develop a methodology that address multi-relational problems through mining the logic relations in the database.
- We propose a novel method to select discriminative patterns using a relaxed version of θ-subsumption.

1082-3409/08 $25.00 © 2008 IEEE
DOI 10.1109/ICTAI.2008.73

- We evaluate the effectiveness of this method on a complex discovery application of practical interest.

Our work is motivated by a challenging application in the area of health informatics. We therefore start by presenting this application and discussing why multi-relational classification is relevant to this domain. In section 4 we give a detailed description of our algorithm and present an illustrative example. In section 5 we demonstrate and discuss the effectiveness of our work with experimental results. Finally, in section 6 we concluded with an overview of this work and present future research directions.

2. Motivating Application

Hepatitis is a virus infection that inflames the human liver. This disease can produce liver fibrosis that could be more or less severe according to virus subtype. Of special interest are patients with B and C virus subtypes since these subtypes have more probability to evolve to liver cirrhosis or liver cancer.

The mechanisms of disease progression and diagnosis are intricate and there is substantial search for new knowledge and ways of diagnosis. Currently, diagnosis is grounded in biopsy examination, an invasive technique, but other data such as blood tests and urinalysis exists that can give some insights into the disease.

Our work is grounded on the Hepatitis dataset donated by Dr. Katsuhiko Takabayaski and Dr. Hideto Yokoi from the Chiba University Hospital, Japan.

The dataset consists of seven tables registering a long term, between 1982 and 1990, monitoring of 771 patients with hepatitis B and C. In this dataset there are two levels of information, one is administrative information and the other is clinical information concerning blood tests, urine analysis and biopsy examinations. The data is organized in a logical entity-relational model(see figure 1).

The patient table *ptData*, contains patient masked identification, sex and date of birth. Other useful tables contain information on blood and urinalysis examinations. One of them, the *inHospRes* table, registers in hospital urinalysis results and one other, *hematRes* table, the blood results. A third table, the *labDesc*, has background information about in hospital examinations, like the reference values of each exam. The *bioRes* table, that register patient biopsy results, contains among other, the date of examination, the hepatite type verified, and the degree of fibrosis results. We will use these results to label patients. We did not use the table *outLabRes* that describes out hospital urinalysis examinations neither the results of interferon treatment, table *interfRes*. All records of these tables, except *labDesc*, have the examination date and patient masked identification. This later field will be used as a foreign key.

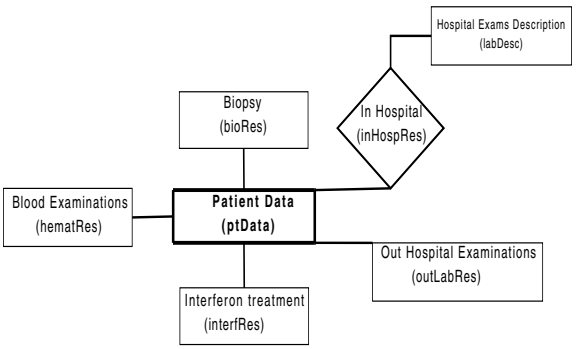

Figure 1. Hepatitis relational model

To the best of our knowledge, few works have explored the relations between tables in this dataset. Namely, we are aware of the work of Ohara et al. [3], that uses a time relation window to convert the dataset to graphs in which they search for frequent patterns using DT-GBI.

One interesting aspect of multi-relational datasets is that they are often source to a number of very different tasks. In this work we study two important classification problems. Our first task is to discriminate between Hepatitis B and C subtypes. The second task is to determine the degree of liver fibrosis. In both cases, the blood and urinalysis results taken before the classification date, the date of the first ground biopsy, are the most relevant information.

3. Methods and Related Work

ILP is a powerful relational formalism, where input is described by relations and learned theory is represented in a first order language. One interesting problem is how to best combine such clauses. One can construct clauses assuming one has to build a classifier [18, 16], or one can perform a two step algorithm where one first learn rules and then builds a classifier [17]. In a different vein, one can approach this problem as upgrading from propositional to a relational learner [4].

Recent work in systems such as SAYU [19], nFOIL [12], kFOIL [11] addresses some of the problems in propositionalition by doing greedy search in the space of clauses. SAYU and nFOIL use a Bayesian classifier to compose clauses, whereas kFOIL uses a kernel computed by calculating which examples match.

Our work is grounded on the excellent results obtained with frequent propositional logic patterns. This work is grounded on a number of algorithms that find frequent item-sets and association rules using a level-wise search. Apriori[1] is arguably the best known. There is also a substantial number of algorithms that use frequent attribute value patterns to achieve results that are undeniable

ex.[2]. But these algorithms have some drawbacks, especially when data has a more complex structure and is not well suited to propositional representation. The direct conversion of this complex data to a flat representation means that some important relational information will be lost. Also some irrelevant and redundant data could be created and consequently the computation of some measures of interest, like support and confidence measures, could be distorted.

Dehaspe [6] present WARMR the algorithm that discovers frequent DATALOG patterns on DATALOG databases in a levelwise way that is closest to APRIORI algorithm, it could be seen as an upgrade to first-order logic representation of the later. The user must specify what to count using a key predicate. The frequency of each generated query is the number of individuals, records in the key table, for which there are query bindings in the database. The search for frequent queries uses a levelwise strategy guided by a declarative language bias, mode and type declarations, and background knowledge. In the generation phase the algorithm uses θ-subsumption to define a order and equivalence relation across queries. This way the algorithm prunes infrequent and redundant queries. Queries that are θ-specializations of infrequent queries and generated queries that are θ-equivalent to previous generated clauses are removed.

Notice that non-ILP approaches have also been used to find frequent patterns in multi-relational data. One popular approach is graph mining. T. Matsuda upgrade the original GBI(Graph-based Induction) to deal direct or undirect graphs, with or without loops, colored or uncolored nodes and links. GBI tries to minimize the graph size using stepwise pair expansion (pairwise chunking), at each step a frequent pattern is compressed to a node.

4. The RUSE Algorithm

In this section we present our classifier. We first introduce some fundamental concepts. Then, we introduce our algorithm that cascades an ILP algorithm designed to discover frequent queries with a decision tree induction algorithm in order to produce a descriptive, easy to interpret and accurate classifier.

4.1. Concepts

When searching a database, $\mathbf{r}$, for frequent queries, Q, we define a bias language, L, that restricts size of the search space and guides the search. We then aim to find theory of frequent queries $Q = \{q \in L : sup(q, \mathbf{r}) > \lambda\}$, where $sup(q, \mathbf{r})$ is the number of times a query q succeeds in database $\mathbf{r}$, and λ is a user-defined support threshold. A query is said to succeed in a database if there are bindings, in the database, for all variables in the query, such that query holds true.

A special case is defined when exploring a database partition. There we define $Q_k = \{q \in L : sup(q, \mathbf{r}_k) > \lambda\}$, where $\mathbf{r}_k$ is a database partition.

When searching for frequent patterns within and across partition databases there is a need to introduce an equivalence relation. One of the most used is Plotkin's [7] θ-subsumption query equivalence.

Definition 1: Let q_1 and q_2 be two queries. If there exists a substitution θ such that $q_1\theta \subset q_2$, we say that q_1 θ-subsumes q_2.

Definition 2: Let q_1 and q_2 be two queries. We say that q_1 is θ-equivalent to q_2 iff q_1 θ-subsumes q_2 and q_2 θ-subsumes q_1.

In our work we need to select *discriminative* patterns. Notice that we will usually learn queries from different data partitions of the data. Queries of interest are queries that have different supports in different partitions, and queries that have support in a single partition.

The next definition formalizes the concept. We define two queries q_1 and q_2 to be agreeing queries, and write $q_1 \cong_\theta q_2$, if

1. q_1 is θ-subsumption equivalent to q_2, and,
2. $|sup(q_1, \mathbf{r}_i) - sup(q_2, \mathbf{r}_j)| < \lambda$

where $sup(q_1, \mathbf{r}_i)$ and $sup(q_2, \mathbf{r}_j)$ are the support values of the queries in different, $i \neq j$, dababase partitions and λ is a threshold value. In our algorithm we define λ equal to the WARMR support threshold.

This definition was motivated by the support value defined to search for frequent patterns. We consider that clauses with a support value above a given threshold are interesting, and we further argue that equivalent queries whose threshold differs by at least a pre-specified amount can be interesting and discriminative.

4.2. Architecture

In a nutshell, our algorithm proceeds in three steps. First, we use WARMR an ILP algorithm that finds frequent patterns, more precisely DATALOG queries. Second, we select discriminative and interesting patterns using two filters. Then we project selected patterns in the target table obtaining an enlarged table with Boolean attributes. Third, using this enlarged table as input to a propositional classifier, Quinlan's C4.5 algorithm [8], we induce a decision tree classifier.

At phase 1 we explore the database relations to find frequent queries. In this phase we search for relational knowledge aiming to find frequent patterns that summarize information from a multi-relational database of prolog facts.

input : a dataset $\mathbf{r}$; two thresholds λ and δ;
output: a decision tree model

1 $\mathbf{r}_1, \ldots, \mathbf{r}_k \leftarrow$ `partition` $(\mathbf{r})$
2 **for** i *to* k **do**
3 $\quad Q_k \leftarrow$ `WARMR` $(\boldsymbol{r}_i, \lambda)$
4 $Q_{disc} \leftarrow$ `discriminate` $(Q_1, \ldots, Q_c, \lambda)$
5 $Q_{sel} \leftarrow$ `chi-square` (Q_{disc}, δ)
6 $r_{enltarget} \leftarrow$ `Propositionalization` $(Q_{sel}, \boldsymbol{r})$
7 `return(C4.5(`$r_{enltarget}$`))`

Algorithm 1: Algorithm pseudo-code

Then, at phase 2, we select discriminative and interesting queries using two filters. One removes non-discriminative patterns using discriminative θ-subsumption. The other uses chi-square statistical test and the class labels of each patient to remove class uncorrelated patterns. Next we use the selected patterns and build an enlarged target table. This step is known as Propositionalisation.

At phase 3 we take the enlarged set as input to the C4.5 algorithm and build a decision tree.

Next we present algorithm 1 in more detail. The algorithm has three input parameters: the dataset of ground facts, $\mathbf{r}$; and two thresholds, the λ support threshold defines the minimum query frequency and is used in testing relaxed θ-subsumption equivalence; and δ value that specifies the confidence level for the chi-square test selection.

4.2.1 *Phase 1 - Search for First Order Descriptors*

In algorithm step 1 we split the dataset according to the number of classes, building a partition, $\mathbf{r}_k$, for each class.

At step 2, using domain specific knowledge we use mode and type declarations, defined in the settings file of WARMR algorithm, and eventually background knowledge, specified as Prolog procedures, to run WARMR in each partition of database $\mathbf{r}_k$. We obtain for each partition k a set of frequent queries Q_k.

4.2.2 *Phase 2 - Filter Selection*

The goal of steps 4 and 5 is to filter uninteresting queries before running the tree induction algorithm. In step 4 we remove non-discriminative queries using relaxed θ-subsumption equivalent queries across all dataset partitions, and get a tighter set

$Q_{disc} = \{q_i | \forall_{q_i \in Q_i} \ \forall_{q_k \in Q_{k \neq i}} \ q_i \ncong_\theta \ q_k\}$ of discriminative patterns.

Even so, we still have redundant queries among the Q_{disc} set. In the next step we use a selection filter (step 5) to remove patterns that are independent or unrelated to the query. The filter uses a chi-square hypothesis test with a user defined confidence level, input parameter δ.

Table 1. Enlarged target table

MID	subtype	sex	bornDate	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}
1	b	m	19540304	1	1	1	0	0	0	0	0	0	0	0
2	c	m	19590220	1	1	1	0	1	0	0	0	1	0	0

Last, in step 6 we project/propositionalize the data in the target table. For each interesting and discriminative patterns/queries we add one new Boolean attribute to the target table obtaining an enlarged table. For each instance in the enlarged target table we set the value of each new attribute according to existence of query bindings. If there exists a binding of the primitive attributes for which the all query succeeds, the attribute takes value one for this individual. Otherwise it takes value zero.

4.2.3 *Phase 3 - Induce using a Propositional Classifier*

At this level we have an enlarged target table extended with new descriptors. Using this enlarged propositional dataset we learn(step 7) a model that besides being accurate should be easily understandable and give insight into underlying concepts. To this, we build a decision tree model by running C4.5 algorithm in the enlarged target table. The generated model can use all attributes of the enlarged target table.

4.3. Illustrative Example

Toward obtaining some intuition on our algorithm we present in some detail a specific run, in this case the 5th run of a ten fold stratified cross validation experiment.

As a result of a pre-processing step, we have a dataset with a total of 503 labeled patients, 206 examples labeled as B subtype and 297 labeled as C subtype. Next, we split the training set in two partitions, one for hepatitis subtype B, $\mathbf{r}_1$, and the other for hepatitis subtype C, $\mathbf{r}_2$.

We then run WARMR in each partition. In this example, we define a support threshold to be $\lambda = 20\%$. This way WARMR algorithm founded 107 frequent queries in the train $\mathbf{r}_1$ partition and 100 in the train $\mathbf{r}_2$ partition.

Next, we join these WARMR findings obtaining 207 frequent queries. Among these there are low discriminative and redundant queries which we aim to eliminate. We eliminate using agreeing queries and obtain 63 queries only.

Among these 63 queries some remain that are uncorrelated with class labels. To eliminate them we use a chi-square statistic test setting $\delta = 5\%$ and selected 11 patterns. We use these patterns to create new attributes on the target table, the table *ptBioData* that is the result of joining tables *ptData* and *bioRes*, this later table is used to lable patients.

In table 1 we present two records of the enlarged target table. In this table we have 11 new attribute, one for each query. For illustrative purposes we present two of the

generated queries:
q_1: ptBioData(A,B,C,D), aggregate(avg, hematRes(A, E, F, G, H, I, J, K, L, M, N),J,R), $R > 46.4$
q_2: ptBioData(A,B,C,D), inHospRes(A,E,F,'g-gtp',G), labDesc('g-gtp', H, I, J, K, L, M), $G > J$

For each target table record, the attribute q_1 takes value 1 if the average of all *J* parameter values, the *R* value, in this case the percentage of hematocrit, recorded on table *hematRes*, before biopsy, have a value higher than the 46.4 threshold. Otherwise the attribute takes value zero. The attribute q_2 takes value one if the patient urinanalysis *g-gtp* examination result, G, is larger than the upper bound of a normal examination.

Using this procedure, known as propositionalization, we enlarged the target table.

Now that we have an enlarged table we use the C4.5 propositional learner to induce a decision tree, figure 2.

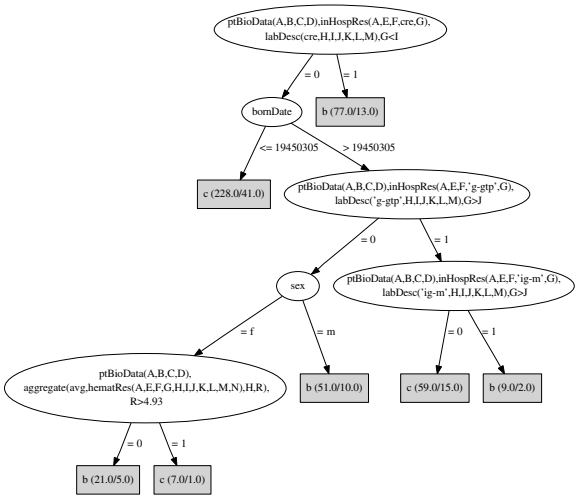

Figure 2. RUSE-WARMR induced decision tree for the subtype problem

5. Experimental Evaluation

In this section we evaluate our algorithm with a number of tasks in the multi-relational hepatitis dataset.

First we will describe the configuration of the experiments. Then we will discuss the preprocessing required for this dataset, and the results obtained for each problem. At the end of this section we analyze and compare the obtained results.

5.1. Experimental Configuration

As mentioned at the Introduction, our algorithm arquitecture is based on a framework that can couple a wide range of algorithms. In this experiments we choose to couple two pairs of algorithms. We present experiments combining WARMR and C4.5 algorithms, and experiments combining WARMR and a linear kernel SVM - Support Vector Machine [15].

We test our algorithm by defining the input parameter λ to be 20% and δ equal to be either 5% or 25%. These values allow us to study filter effectiveness.

We further study the effectiveness of the discriminative θ-subsumption and of the chi-square filters by discarding steps 4 and 5 of our algorithm pseudo-code(Algorithm 1). In the tables below, that describe our results, we write *No-θ* when we do not use any of the two filters and *No-chi* when we only remove agreeing clauses. We also present results when using a *Standard* algorithm, an algorithm that takes the full training set as input (no partitions!) and uses no filter to prune WARMR findings. These descriptions appear in the δ column.

To best analyze the contribution of the algorithm used in phase 3, we test two different classifiers: C4.5 and a linear kernel SVM - Support Vector Machine.

We obtained results by running tenfold stratified cross-validation test. Using these results we compute the mean and standard deviation of the generalization error across all runs.

We use Wilcoxon signed ranked pair-test, with a 95% confidence level, to evaluate how significantly our algorithms differ from a standard approach, *WARMR + C4.5* or *WARMR +SVM*. The null hypothesis is that the median of the differences is zero.

To implemented our work we use ACE [13], data mining System and WEKA [14], a collection of data mining algorithms.

5.2. Dataset Pre-processing

Our first step was to build a new table, named *ptBioData*, that merges all information from the patient table and the hepatitis type and fibrosis degree from the biopsy table. This new table will be our target table. We also use all the information in *inHospRes*, *labDesc* and *hematRes* tables. We further observe that biopsy examination results could imply the prescription of medication, namely interferon, and change patient habits. We thus chose to select only examinations recorded before the date of the first biopsy. Furthermore, all patients that have missing information on the biopsy examination, subtype and degree of fibrosis, were discarded and all related tables were also removed from the

database. Note that even after this preprocessing we still have 503 examples.

In order to run WARMR, we discretize all the results in the *hematRes* table, we used equal frequency bins metodologie [9] with three thresholds. These thresholds are then used in WARMR's mode and type declarative bias. We set the maximum level of refinement to 2.

WARMR specifies bias through the rmode formalism. We use the same bias for every problem. Our bias incorporates background knowledge such as the fact that each parameter of a normal urinalysis examination should be inside a reference interval. Since one patient can do the same exam in different time periods, we also introduce a refinement that aggregates, for each patient and exam, all the results using average [10].

We define two main styles of refinements. The first uses the reference values of an examination to generate rules that describe if patient exams are within or out of the reference bounds. The second uses the average operator and the thresholds obtained at the discretization to generate queries. The queries obtained using this later refinement express the knowledge that the average of results obtained by a patient in one parameter/exam is above or below a certain threshold.

5.3. Classification Problems

We address two classification problems in the hepatitis dataset. First we address the hepatitis subtype problem, were we aim to distinguish patients with B and C virus. Second we address the fibrosis problem, to distinguish among patients having diferent levels of fibrosis [3].

5.3.1 *Hepatitis subtypes B and C*

In the hepatitis subtype problem we used all 206 patients with subtype B and 297 with subtype C hepatitis.

When running WARMR with support threshold $\lambda = 20\%$ the algorithm founded an average of 206 queries in each run. Then, when we apply the discriminative filter we selected an average of 62.4 queries. The number of patterns selected to enlarge the target table after applying the chi-square filter is shown in table 2(the number of presented patterns includes primitive attributes). In this table we also present the results obtained using the C4.5 algorithm at phase 3 of the algorithm. For comparison purposes we also show results using a linear kernel SVM algorithm at phase 3.

The tree presented in the illustrative example was the obtained by C4.5 at the best run with $\lambda = 20\%$ and $\delta = 5\%$.

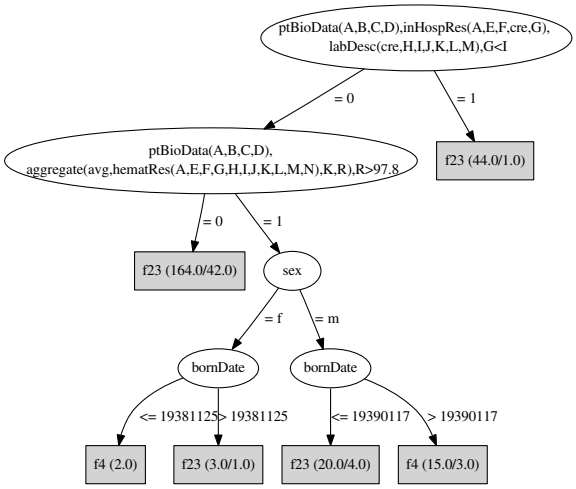

Figure 3. Best decision tree for the fibrosis problem

5.3.2 *Figrose degree $\{F2,F3\}$ vs $\{F4\}$*

In this experiment we address the problem of distinguishing between patients having no liver cirrhosis, stages F2 and F3, and patients having liver cirrhosis, stage F4, again following Ohara [3]. We discarded patients having degree level 0 and 1. Thus we selected 276 patients, 67 patients labeled F4 and 209 patients labeled F2 or F3.

When running WARMR with $\lambda = 20\%$ we found an average of 212.8 queries in each run. Then, when we apply the discriminative filter we selected 11.6 queries. The number of patterns selected using chi-square filter and the results obtained using C4.5 algorithm at phase 3 are shown in table 3(the number of presented patterns includes primitive attributes). For comparison purpose we also show the results obtained when using SVM algorithm at phase 3.

Figure 3 presents the tree generated in the best run obtained using $\lambda = 20\%$ and $\delta = 5\%$, and was generated in 4th run.

5.4. Analysis

The Hepatitis dataset is a difficult dataset. Our results are somewhat better than base accuracy, with the best results being obtained through the RUSE algorithm. In both cases, there seems to be a substantial benefit in using the RUSE rule selection steps. This is particularly clear when using the C4.5 classifier, and is unsurprising, given that SVMs are known to be more robust to extra parameters. It is very interesting to notice the effectiveness of the filters. Using both filters we pruned the WARMR findings by more than 30%, in some cases we get a reduction of 97%.

Also our algorithm proved to be accurate and produce

λ	δ	Mean Number of Patterns and Std. deviation after both filters	Mean Error (Std. Deviation)		Wilcoxon Test p-palue	
			C4.5	**SVMs**	**C4.5**	**SVMs**
20	5	15.3 (3.3)	34.87 (8.48)	38.87 (8.38)	0.015	1
	25	30.7 (4.19)	37.39 (7.19)	36.6 (6.83)	0.192	0.529
	No-Chi	65.4 (2.37)	37.76 (5.85)	37.87 (7.35)	0.232	0.624
	No-θ	211 (1.33)	39.64 (9.54)	41.67 (10.76)	0.106	0.081
	Standard	109.1 (1.1)	41.63 (9.2)	39.25 (10.52)	-	-

Table 2. Generalization Error using C4.5 and SVMs for the B vs C subtype problem

λ	δ	Mean Number of Patterns and Std. deviation after both filters	Mean Error (Std. Deviation)		Wilcoxon Test p-value	
			C4.5	**SVMs**	**C4.5**	**SVMs**
20	5	6.5 (1.14)	24.95 (4.28)	24.23 (1.86)	0.012	0.006
	25	8.3 (1.7)	25.66 (6.36)	24.23 (1.86)	0.012	0.006
	No-Chi	14.6 (2.5)	32.99 (8.44)	36.15 (8.56)	0.102	0.114
	No-θ	196 (2.33)	37.66 (9.71)	35.41 (8.55)	0.557	0.04
	Standard	109.2 (1.75)	42.06 (13.54)	47.97 (12.56)	-	-

Table 3. Generalization Error using C4.5 and SVMs for the fibrosis stage problem

compact and understandable trees. The analysis of the generalization error and standard deviation shows that the filter methodologies can improve the performance of the classifier by more than 10%. This is because the system always selects very few rules, the most interesting ones. This again suggests that our tests are removing uncorrelated features effectively.

When we analyze the p-value of the Wilcoxon hypothesis test, we observe that our algorithm differ from the standard approach algorithm. This is specially clear in the fibrosis stage problem. One interesting issue is the difference between the $No - \theta$ algorithm and the standard approach. None of the algorithms use any kind of filter, the only difference between then is that the first takes as input the dataset partitions whereas the other takes the full dataset. This can be explained by the number of patterns founded by WARMR. In the standard algorithm this number is approximately half of our algorithm. This was expected because our algorithm uses the same bias refinements to search in both partitions.

To the best of our knowledge, best results on this dataset were obtained by Ohara [3]. Unfortunately, we cannot compare with their results because they use different preprocessing that we could not repeat. Namely, our results focus on early diagnosis and are thus based on data prior to the biopsies. Thus, we did not use the hospital examination *outLabRes* table, and we discarded all examination done later than the first biopsy.

One interesting issue that will be the focus of our attention in the future is the run time of our classifier. Because of the parallel nature of our classifier we get results faster than is usual in ILP algorithms.

We try to compare our algorithm against TILDE [5], but we get out of memory in a 4 GB machine.

Some patterns that we present in this work, and the generated trees, represent current knowledge on the hepatitis disease. Other patterns, also some that we found and did not present here, need to be analyzed by a clinical technician.

6. Conclusions and Future Work

In this work we present a general classification algorithm for complex structured data. In contrast to the classical approach that propositionalizes data as a preprocessing step, our algorithm explores the logical relational structure in the database to produce more expressive and accurate classifiers.

We developed a three phase arquitecture that combines sequentially two algorithms and a selection filter. In the first phase we search for frequent first-order logic patterns using WARMR algorithm. In the second phase we introduce two filters to remove non discriminative and select interesting class correlated patterns. To do so we introduce a new equivalence relation and we use the chi-square statistics test. Then, we use this patterns to extend the target table with new Boolean attributes. In the third phase we use a standard classifier, such as a decision tree model or a SVM.

One interesting contribution is that we do not run the WARMR algorithm in the full dataset. Instead, we search for frequent patterns in each class partion of the available dataset. We select discriminative patterns using all frequent patterns found in every partition. With this we achieve efficiency and a natural parallelized algorithm.

The models generated by our classifier proof to be expressive/readable. The methodology seems to be effective since all obtained models have both ILP patterns and primitive attributes from the target table at the inner nodes. An

interesting observation is that in specific runs all tree inner nodes have only ILP patterns.

Though our algorithm uses WARMR at phase 1 and C4.5 (or SVM) at phase 3, Wilcoxon significance test prove that our algorithm differs from the classical approach, whereas both algorithms are combined without using any pattern pruning/selection strategy.

In the future we aim to improve the quality of rules we generate and study other problems from this and other datasets. Regardless, the hepatitis dataset is complex and challenging dataset that will continue to be the subject of our attention. We will try other discretization and aggregation methodologies. We will also explore time-dependent changes by introducing change detection and concept drift techniques. We are already searching for a clinical technician to analyze the results obtained in the experiments.

Acknowledgements

Gratitude is expressed to the project ALES II - Adaptive LEarning Systems program (POSC/EIA/55340/2004). Vítor Santos Costa is partially supported by the Fundação para a Ciência e Tecnologia, and by the JEDI (PTDC/EIA/66924/2006) and STAMPA (PTDC/EIA/67738/2006) projects.

References

[1] Agrawal, R. and Srikant, R., *Fast Algorithms for Mining Association Rules*, Proceedings of the Intelligence Conference on Very Large Databases, Santiago, Chile, 1994, pp. 487-499.

[2] Ferreira, C. and Gama, J., *Rank Ensemble Features for Constructive Induction*, Proceedings of the Workshop on General Artificial Intelligence, in the 13th Portuguese Conference on Artificial Intelligence (EPIA), Guimarães, Portugal, 2007, pp. 45-57.

[3] Ohara, K. et al., *Analysis of Hepatitis Dataset by Decision Tree Graph-Based Induction*, Proceedings of Discovery Challenge Workshop held in conjunction with the 15th European Conference on Machine Learning and and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Pisa, Italy, 2004, pp.173-184.

[4] Laer, V. W. and Luc de Raedt, *How to Upgrade Propositional Learners to First Order Logic: a Case Study*, Relational Data Mining, New York, USA, 2001, pp. 235-256.

[5] Blockeel, H. and Raedt, L., *Top-Down Induction of First-Order Logical Decision Trees*, Artif. Intell., IOS press, 1998, pp. 119-120.

[6] Dehaspe, L. and Toivonen, H., *Discovery of frequent DATALOG patterns*, Data Mining and Knowledge Discovery, Kluwer Publishers, 1999, pp. 7-36.

[7] G. Plotkin, *A note on inductive generalization*, In Machine Intelligence, 1970, pp. 153-163.

[8] Quinlan, J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[9] Dougherty, J. et al, *Supervised and Unsupervised Discretization of Continuous Features*, International Conference on Machine Learning (ICML), 1995, pp. 194-202.

[10] Assche, V. A. et al, *First order random forests: Learning relational classifiers with complex aggregates*, Machine Learning, 2006, pp. 149-182.

[11] Landwehr, N. et al, *kFOIL: Learning Simple Relational Kernels*, AAAI, 2006.

[12] Landwehr, N. et al, *nFOIL: Integrating Nave Bayes and FOIL*, AAAI, 2005.

[13] Blockeel, H. et al, *Executing query packs in Inductive Logic Programming*, Proceedings of the 10th International Conference in Inductive Logic Programming, 2000, pp. 60-77.

[14] Witten, I. and Frank, E., *Data mining: practical machine learning tools with Java Implementations*, Morgan Kaufmann, 1999.

[15] Platt, J., *Sequential minimal optimization: A fast algorithm for training support vector machines*, TR 98-14, Microsoft Research, 1998.

[16] Flach,P. A. and Lachiche, N., *Naive Bayesian Classification of Structured Data*, Machine Learning, 2004.

[17] Pompe, U., and Kononenko, I., *Naive Bayesian classifier within ILP-R*, Proceedings of the 5th InternationalWorkshop on Inductive Logic Programming, 1995, pp. 417-436.

[18] Zelezny F. and Lavrac N., *Propositionalization-Based Relational Subgroup Discovery with RSD*, Machine Learning, 2006, pp. 33-63.

[19] Davis, J. et al, *Learning Bayesian networks of rules with SAYU*, Proceedings of the 4th international workshop on Multi-relational mining, 2005.