Multi-Relational Data Mining in Medical Databases

Amaury Habrard, Marc Bernard, and François Jacquenet

EURISE – Université de Saint-Etienne – 23, rue du Dr Paul Michelon
42023 Saint-Etienne cedex 2 – France
{Amaury.Habrard,Marc.Bernard,Francois.Jacquenet}@univ-st-etienne.fr

Abstract. This paper presents the application of a method for mining data in a multi-relational database that contains some information about patients strucked down by chronic hepatitis. Our approach may be used on any kind of multirelational database and aims at extracting probabilistic tree patterns from a database using Grammatical Inference techniques. We propose to use a representation of the database by trees in order to extract these patterns. Trees provide a natural way to represent structured information taking into account the statistical distribution of the data. In this work we try to show how they can be useful for interpreting knowledge in the medical domain.

1 Introduction

The main objective of Data Mining techniques is to extract regularities from a large amount of data. For this purpose some efficient techniques have been proposed like the apriori algorithm [2]. However these techniques are closely related to data stored in a flat representation even though more and more structured data (like relational databases) are used in all domains of activity. Thus to deal with algorithms working on flatten data some pre-processing steps are required that unfortunately lead to lose some valuable information. Cios and Moore pointed out some features that make medical data mining unique [6]. We think that beeing able to deal with structured data is also especially important in medicine where databases may contains a large number of tables.

Over the last years, several attempts have been made to extract regularities directly from databases without having to flatten data. That has led to the emergence of an active field of research, called multi-relational Data Mining [8]. For example, the ILP system WARMR [9] defines a generic framework to learn from a datalog representation of a database, keeping the structuring of data. Crestana-Jensen and Soparkar [7] propose an algorithm for mining decentralized data exploiting the inter-tables relationships to extract frequent itemsets on separate tables. In this paper we present a method allowing to extract some knowledge from data structured as trees. Trees are natural candidates for representing structured information of multi-relational databases. Mining some probabilistic knowledge in this context leads to mining tree patterns that respect the statistical distribution observed over the data. A tree pattern can be

viewed as an abstraction from trees and thus provides an interesting way to represent regularities observed in large amount of data. The concept of tree patterns has received a lot of interest during the past two years. In the machine learning field, learning of tree patterns has been studied for example by Amoth, Cull and Tadepalli in [3] or by Goldman and Kwek in [13]. More recently, data mining approaches have been proposed to extract these patterns [4, 10, 17, 16, 18]. In this paper we are interested in statistical learning methods which may improve the Data Mining task. Indeed these methods are interesting for Data Mining because they provide a quantitative approach to weighting the evidence supporting alternative hypotheses. They are known to perform well with noisy data and may discover some knowledge from positive examples only.

The method we present follows several steps. First we have to define the table containing data we want to focus on. From the rows of this table we generate a set of trees. For each row, a tree is generated recursively using the relationships defined between various tables by foreign keys. Using some probabilistic grammatical inference techniques we learn a probabilistic tree grammar on the database. The tree grammar is represented with a stochastic tree automaton. Then we use a level wise search approach to generalize transition rules of the automaton. These generalized rules are finally used to produce a set of probabilistic tree patterns.

Each step of our method will be described in the next two sections. Then, in section 4 we show how our system may be applied to the discovery of knowledge about chronic hepatitis by extracting probabilistic tree patterns from a relational database. We have focused our work on the relations between the level of liver fibrosis and laboratory examinations made on patients. The level of liver fibrosis is often closely related to the stage of hepatitis C, a disease which may lead to develop liver cirrhosis or hepatocarnocima. The final objective is to point out laboratory examinations that can predict the level of liver fibrosis.

The work presented in this paper can be view as a preliminary work on the extraction of probabilistic tree patterns in relational database on medical data. The objective is to see how these patterns can be useful in multi-relational data mining tasks and how they can model structured data in relational databases.

2 Stochastic tree automata

A tree automaton [12] defines a regular tree language as a finite automaton defines a regular language on strings. Stochastic tree automata are an extension of tree automata and define a statistical distribution on the tree language recognized by the automata. Learning tree automata has received attention for some years. For example Garcia and Oncina [11] or Knuutila and Steinby [15] have proposed some algorithms for learning tree automata. In the probabilistic framework, Carrasco, Oncina and Calera [5] proposed an efficient algorithm to learn stochastic tree automata ; Abe and Mamitsuka [1] dealt with learning stochastic tree grammars to predict protein secondary structure. In this section we mainly describe stochastic tree automata, which are the core of the system.

In fact we consider an extension of stochastic tree automata which takes sorts into account. Thus we consider many-sorted stochastic tree automata. We first define the concept of signature, which represents the set of constructible trees.

Definition 1 *A signature Σ is a 4-tuple (S, X, α, σ). S is a finite set whose elements are called sorts. X is a finite set whose elements are called function symbols. α is a mapping from X into $\mathbb{N}$. $\alpha(f)$ will be called the arity of f. σ is a mapping from X into S. $\sigma(s)$ will be called the sort of s.*

Definition 2 *A stochastic many-sorted tree automaton (SMTA) is a 5-tuple $(\Sigma, Q, r, \delta, p)$. Σ is a signature (S, X, α, σ). $Q = \cup_{s \in S} Q^s$ is a finite set of states, each state having a sort in S. $r : Q \longrightarrow [0, 1]$ is the probability for the state to be an accepting state. $\delta : X \times Q^* \longrightarrow Q$ is the transition function. $p : X \times Q^* \longrightarrow [0, 1]$ is the probability of a transition.*

We denote $R_{A,g,q}$ the set of all rules of the form $g(q_1, \ldots, q_n) \rightarrow q$ of a tree automaton A. Note that we assume we do not allow overloading of function symbols.

A SMTA parses a tree using a bottom-up strategy. A state and a probability are associated with each node of the tree. The labelling of each node is defined by the transition function. The tree is accepted if the probability of its root node is strictly positive. Given a SMTA A, the probability of a tree t is computed as follows:

$$p(t \mid A) = r(\delta(t)) \times \pi(t)$$

where $\pi(f(t_1, \ldots, t_n))$ is recursively computed by:

$$\pi(t) = p(f, \delta(t_1), \ldots, \delta(t_n)) \times \pi(t_1) \times \cdots \times \pi(t_n)$$

Example The automaton A defined by the following transition rules is able to recognize, for instance, the tree $g(f(c, b, a))$ with an associated probability of 0.15.

1.0 : `a` → `q1`	1.0 : `b` → `q2`	1.0 `c` → `q5`
0.6 : `f(q1,q2,q4)` → `q3`	0.15 : `f(q5,q2,q1)` → `q3`	0.15 : `f(q1,q4,q3)` → `q3`
0.1 : `f(q1,q2,q3)` → `q3`	1.0 : `g(q3)` → `q4`	Final state : $r(q4) = 1.0$

The inference of a tree automata is made from a sample of trees defining a dataset. We do not detail the procedure here, the interested reader may consult [5, 14]. The structure of the automaton is iteratively constructed using a state merging procedure. The probabilities are then computed from the train sample, taking into account the distribution of the data. We generalize rules of the automaton, looking for frequent regularities in the database, relatively to the distribution of the dataset. This step of generalization allows to generate probabilistic tree patterns, modeling concepts stored in the database.

3 Generalization of a SMTA

We have proposed in [14] a technique to generalize SMTA relatively to a threshold γ. The idea is to generalize the transition rules of the automaton using a level wise algorithm. The generalization algorithm is local, that is it considers generalization of a SMTA locally to a given arrival state and a given symbol.

The generalization process considers rules containing variables, these rules are called generalized rules. The score of a generalized rule is computed by adding probabilities of the transition rules of the SMTA subsumed by the generalized rule. The algorithm looks for the most specific generalized rules having a score greater than γ. Algorithm presented in [14] extracts generalized rules from all the sets $R_{A,f,q}$ definable in the SMTA.

Definition 3 *Let V a set of variables and Q a set of states. Let r_1 and r_2 be two rules: $r_1 = f(x_1, \ldots, x_n) \rightarrow q$ and $r_2 = f(x'_1, \ldots, x'_n) \rightarrow q'$ such that $x_i \in Q \cup V$ and $x'_i \in Q \cup V$. r_1 is more general than r_2 (we note $r_1 > r_2$) if and only if there exists a substitution θ such that $f(x_1, \ldots, x_n)\,\theta = f(x'_1, \ldots, x'_n)$. We say that r_1 subsumes r_2.*

Definition 4 *A rule r is called a generalized rule if there exists a rule r' in $R_{A,f,q}$ such that $r > r'$.*

For example, consider the following set of rules $R_{A,f,q3}$ from a SMTA A:

$$0.10 : f(q1, q2, q3) \rightarrow q3 \qquad 0.15 : f(q1, q4, q3) \rightarrow q3$$
$$0.60 : f(q1, q2, q4) \rightarrow q3 \qquad 0.15 : f(q5, q2, q3) \rightarrow q3$$

If we fix the γ parameter to 0.6, the generalization algorithm will extract the following generalized rules:

$$0.6 : f(q1, \_, q4), \quad 0.7 : f(q1, q2, \_) \text{ and } 0.6 : f(\_, q2, q4)$$

In this work we are interested in extracting probabilistic tree patterns. These tree patterns give a probabilistic representation of the database. Informally a tree pattern is a tree whose leaves can be either variables or symbols of arity zero. Formally we define tree patterns as terms in first order logic.

Definition 5 *Let $\Sigma = (S, X, \alpha, \sigma)$ a signature and V a set of variables. A tree pattern, on $\Sigma \cup V$, is defined recursively as follows:*

- *a symbol $s \in X$, such that $\alpha(s) = 0$, is a tree pattern*
- *a variable $v \in V$ is a tree pattern*
- *if $t_1, \ldots, t_n$ are tree patterns, then $f(t_1, \ldots, t_n)$ is a tree pattern for any symbol $f \in X$ such that $\alpha(f) = n$*

Definition 6 *A probabilistic tree pattern is a couple (t, p) where t is a tree pattern and p a probability $(0 \leq p \leq 1)$.*

To extract these patterns we use a depth first search on the generalized rules of the SMTA. The idea is to start from the final states of the generalized SMTA (that is states with a probability greater than zero to be a final state) and to construct the tree patterns recognized by the automaton. This process is recursive using the generalized rules of the automaton. The rules are chosen in function of their arrival state, when many rules can be used, we generated tree patterns for each possibility. The probability of a tree pattern is the product of the probabilities of the rules used in the depth first search process.

4 Mining the Chronic Hepatitis Data

In this section we present the work we have done with our system in order to discover some knowledge about chronic hepatitis. The data we used were prepared in the context of a collaboration between the Shimane Medical University, School of Medicine and Chiba University Hospital. The database stores 771 patients with hepatitis B and C who took examinations in the period 1982-2001. Hepatitis A, B and C are virus infections that affect the liver of the patient. Hepatitis B and C are especially important because they have a potential risk of developing liver cirrhosis or hepatocarcinoma. An indicator that can be used to know the risk of cirrhosis or hepatocarcinoma is fibrosis of hepatocyte. For instance liver cirrhosis is characterized as the terminal stage of liver fibrosis. One way to evaluate the stage of liver fibrosis is to make a biopsy, but this examination is invasive to patients, thus it could be interesting to use laboratory examinations as substitutes for biopsy.

In this work we propose to extract probabilistic tree patterns trying to link the stage of liver fibrosis and in-hospital and out-hospital laboratory examinations. For this purpose we focus on four levels on the five described in biopsy examinations, that are levels F1, F2, F3 and F4 which is the most severe stage. Then we construct a sample for each level, each sample is made up of trees taking into account laboratory examinations. The following section presents the preparation of the data.

4.1 Data preparation

Data are organized in a relational database made up of six tables. One table gives information about patients (PT_E table), another one gives results of biopsy on patients (BIO_E table) while information on interferon therapy are stored in the IFN_E table. Two tables give results of in-hospital and out-hospital examinations (ILAB_E and OLAB_E tables), the last table (LABN_E table) gives information about measurements in in-hospital examinations. We stored the data in a relational database, using the relational database management system PostgreSQL.

Let us recall that our system extracts knowledge from data in the form of trees. Thus, using the database, we have to build those trees. For a comprehensive general presentation of the transformation process, the reader may refer to [14]. In this particular case, for each level of liver fibrosis, we choose the table

PT_E to build the root of each tree and we select tuples corresponding to patients having a biopsy with the considered level of liver fibrosis. In this table we only consider the MID attribute which is the identifier of each patient. This attribute is duplicated to define two foreign keys: one on table $OLAB\_E$, the second one on table $ILAB\_E$. Using table $OLAB\_E$ we continue building the trees by considering the attributes $OLAB\ Exam\_Name$ (name of the out-hospital examination), $OLAB\ Exam\_Result$ (result of the out-hospital examination), $OLAB\ Evaluation$ (evaluation of the result) and $OLAB\ Eval\_SubCode$ (internal subcode of the evaluation items). Finally using table $ILAB\_E$ we build the trees by considering the attributes $ILAB\ Exam\_Name$ (name of the in-hospital examination) and $ILAB\ Exam\_Result$ (result of the in-hospital examination). To construct the subtrees corresponding to the foreign keys on tables $ILAB\_E$ and $OLAB\_E$, we consider only records so that the time between the date of examination and the date of the biopsy of the patient doesn't exceed two days.

Values of attributes $ILAB\ Exam\_Result$ and $OLAB\ Exam\_Result$ are discretized taking into account information of table $LABN\_E$. If an examination has an entry in this table, we consider the value as normal if it is between the upper and lower bound, otherwise we measure its level in function of a discretized step. For other examinations, we compute a mean to define lower and upper bounds, then we apply the same policy.

The tree of figure 1 is an example of tree produced for a patient at level F2. This patient has no out-hospital examinations, and has two in-hospital examinations: one on activated partial thromboplastin time (APTT) which has a normal value and one on prothrombin time (PT) which has also a normal value.

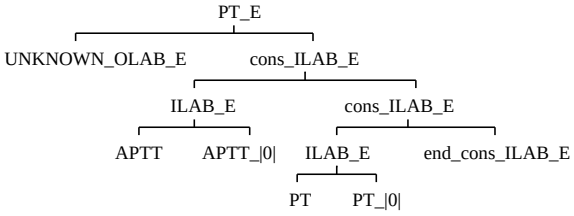

Fig. 1. Tree associated with patient with MID=414

4.2 Experimentation on level of liver fibrosis

We generate a sample of trees for each level of liver fibrosis (F1 to F4). Then, for each sample we learn a generalized SMTA with different levels of generalization. Let us recall that the level of generalization depends on the γ parameter ($0 \leq \gamma \leq 1$) and the higher this parameter is, the more we generalize. Thus, if this parameter is high we extract very few patterns which are often too general. On the other hand, if it is low, a lot of patterns are extracted which are very specific.

For example, we extract 768320 patterns with $\gamma = 0.15$ and only 4 with $\gamma = 0.85$ on patients at level F3. So this parameter may be used to define the specialization of the extracted knowledge. For this purpose we learned generalized SMTA using the γ parameter from 0.1 to 0.85. Then for each generalized model, we extract probabilistic tree patterns.

To illustrate the extracted knowledge, we give some examples of tree patterns and their probabilities. As discussed in a previous section, some leaves of trees correspond to discretized attributes of the database, and thus the number of the interval of values for such attributes is denoted between two vertical lines. Understanding tree patterns is not always easy and it may requires some joint work between data miners and medical experts. It could be useful to design a tool that would convert tree patterns into some natural language sentences but this is not the aim of this paper to discuss such a tool.

Figure 2 shows a general pattern, of probability 0.66, extracted with $\gamma = 0.7$ from patients at level F2. Informally this pattern says that patients, at level F2, may have at least one in-hospital examination with probability 0.66. Figure 3

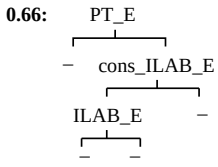

Fig. 2. Tree pattern of level F2 extracted with $\gamma = 0.7$

shows a less general pattern saying that 20% of patients at level F1 generally have at least two in-hospital examinations with one albumin (ALB).

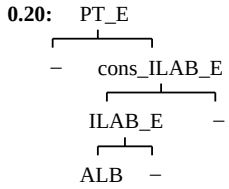

Fig. 3. Tree pattern of level F1 with extracted with $\gamma = 0.55$

Figure 4 shows a very specific pattern of probability 0.0004. This pattern says that 0.04% patients at level F3 often have at least one out-hospital examination on "D inshi" with no evaluation and a subcode equals to F504 and at least one in-hospital examination on albumin (ALB).

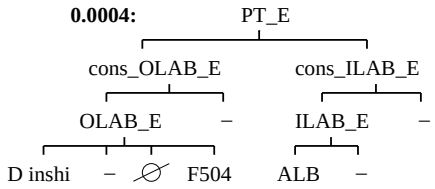

Fig. 4. Tree pattern of level F3 extracted with $\gamma = 0.1$

Figure 5 shows a less specific pattern. It says that 6.9% of patients at level F4 may have one in-hospital examination on albumin (ALB) with value between 0 and 3.9 g/dl. We now sum up on figure 6 of the influence of the γ parameter

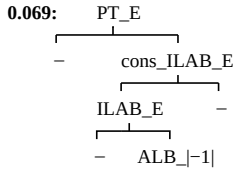

Fig. 5. Tree pattern of level F4 extracted with $\gamma = 0.1$

on the number of extracted tree patterns. For small values of γ, we expect a weak generalization and thus a high number of patterns with a priori small probabilities. When γ tends to 1, we expect overfitting and so a small number of patterns. Note that if too general patterns are not very interesting, patterns with low probabilities can be useful because they can point out rare events.

We now give a summary of the probabilistic tree patterns extracted by our system. Note that in our work we only consider examinations made in the same period of the biopsy examination. Thus a larger study seems necessary to interpret these results.

- patients at level F1: For in-hospital examinations we extract that 20% of patients have ALB examinations. We also extract patterns saying that 17% of patients at level F1 have normal values for this examination. For out-hospital examinations, the examination "ABO shiki ketsuekigata kensa" concerns about 50% of the patients. 20% of them have "O kata" and 15% have "B kata" as result for this examination.
- patients at level F2: The out-hospital examination "ABO shiki ketsuekigata kensa" is present for 30% of patients. For in-hospital examinations we notice the presence of albumin (ALB) (20%) and alkaline phosphatase (ALP) (20%).

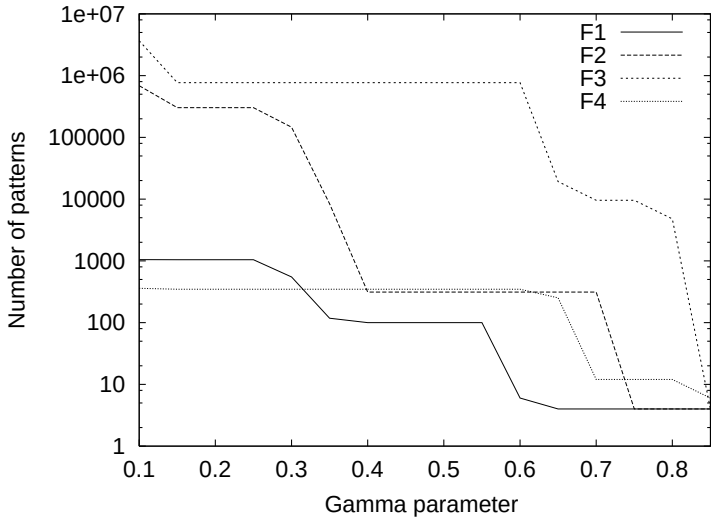

Fig. 6. Number of patterns relatively to γ

- patients at level F3: The out-hospital examination "ABO shiki ketsuekigata kensa" is also present for 18% of patients. We also notice the presence of "D inshi" examination for 17% of patients. These two examinations are both linked with the following subcodes (uniformally distributed): E499, E500, E501, E502, F503, F504 and F505. The albumin (ALB) examination is also present for in-hospital tests for 26% of patients.
- patients at level F4: We notice the examination albumin (ALB) for 20% of the patients, direct bilirubin (D-BIL) and cholinesterase (CHE) for 7% of patients in in-hospital examinations. Note that for albumin exams, the patients have a probability of 12% to have a result lower than the normal range.

5 Conclusion

In this paper we have experimented a method to extract probabilistic tree patterns from a medical database. This method is based on a representation of the database by a set of trees and the inductive phase consists in first learning a SMTA and generalizing this model relatively to a parameter. In the context of a database about chronic hepatitis, we are able to link data from many relations of the database, like out-hospital and in-hospital examinations.

One interesting perspective of this work would be to define a language bias with medical experts to embed all the relevant information in trees. Then we may try to use the method to extract other kind of knowledge. Another perspective would be to work on the way probabilistic tree patterns could be used as a condensed representation of the database.

References

1. N. Abe and H. Mamitsuka. Predicting protein secondary structure using stochastic tree grammars. *Machine Learning*, 29:275–301, 1997.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.
3. T. R. Amoth, P. Cull, and P. Tadepalli. On exact learning of unordered tree patterns. *Machine Learning*, 44(3):211, 2001.
4. H. Arimura, H. Sakamoto, and S. Arikawa. Efficient learning of semi-structured data from queries. In *12th International Conference on Algorithmic Learning Theory*, volume 2225 of *Lecture Notes in Computer Science*, pages 315–331, 2001.
5. R.C. Carrasco, J. Oncina, and J. Calera. Stochastic Inference of Regular Tree Languages. *Machine Learning*, 44(1/2):185–197, 2001.
6. K.J. Cios and G.W. Moore. Uniqueness of medical data mining. *Artificial Intelligence in Medicine*, 26:1–24, 2002.
7. V. Crestana-Jensen and N. Soparkar. Frequent itemset counting across multiple tables. In *4th Pacific-Asian conference on Knowledge Discovery and Data Mining (PAKDD 2000)*, pages 49–61, April 2000.
8. L. De Raedt. Data mining in multi-relational databases. In *4th European Conference on Principles and Practice of Knowledge*, 2000. Invited talk.
9. L. Dehaspe and H. Toivonen. Discovery of frequent DATALOG patterns. *Data Mining and Knowledge Discovery*, 3(1):7–36, 1999.
10. J. Ganascia. Extraction of recurrent patterns from stratified ordered trees. In *12th European Conference on Machine Learning (ECML'01)*, volume 2167 of *LNCS*, pages 167–178, Freiburg, Germany, 2001. Springer.
11. P. García and J. Oncína. Inference of recognizable tree sets. Research Report DSIC - II/47/93, Departamento de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia, 1993.
12. F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984.
13. S. A. Goldman and S. S. Kwek. On learning unions of pattern languages and tree patterns. In *10th Algorithmic Learning Theory conference*, volume 1720 of *Lecture Notes in Artificial Intelligence*, pages 347–363, 1999.
14. A. Habrard, M. Bernard, and F. Jacquenet. Generalized stochastic tree automata for multi-relational data mining. In *Proceedings of the Sixth International Colloquium on Grammatical Inference (ICGI 2002)*, volume 2484 of *LNCS*, pages 120–133. Springer, 2002.
15. T. Knuutila and M. Steinby. Inference of tree languages from a finite sample: an algebraic approach. *Theoretical Computer Science*, 129:337–367, 1994.
16. R. Kosala, J. Bussche, M. Bruynooghe, and H. Blockeel. Information extraction in structured documents using tree automata induction. In *6th European Conference on Principles and Practise of Knowledge Discovery in Databases (PKDD'02)*, volume 2431 of *LNCS*, pages 299–310. Springer, 2002.
17. T. Miyahara, Y. Suzuki, T. Shoudai, T. Uchida, K. Takahashi, and H. Ueda. Discovery of frequent tag tree patterns in semistructured web documents. In *sixth Pacific Asia Conference on Knowledge Discovery and Data mining (PAKDD 2002)*, Taipei, Taiwan, May 2002.
18. M. Zaki. Efficiently mining frequent trees in a forest. In *Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Edmonton,Alberta, Canada, July 2002.