Multi-relational Structural Bayesian Classifier

Michelangelo Ceci, Annalisa Appice, Donato Malerba, and Vincenzo Colonna

Dipartimento di Informatica, Università degli Studi à di Bari
via Orabona 4, 70126 Bari, Italy
{ceci,appice,malerba}@di.uniba.it
colonna_vin@yahoo.it

Abstract. In the traditional naïve Bayes classification method, training data are represented as a single table (or database relation), where each row corresponds to an example and each column to a predictor variable or a target variable. In this paper we propose a multi-relational extension of the naïve Bayes classification method that is characterized by three aspects: first, an integrated approach in the computation of the posterior probabilities for each class; second, the applicability to both discrete and continuous attributes; third, the consideration of knowledge on the data model embedded in the database schema during the generation of classification rules. The proposed method has been implemented in the new system Mr-SBC and tested on three benchmark tasks. Results on predictive accuracy favour our system for the most complex task. Mr-SBC also proved to be an efficient multi-relational data mining system with a tight dose integration to a relational DBMS.

1 Introduction

In the usual classification setting, data are generated independently and with an identical distribution from an unknown distribution P on some domain $\mathbf{X}$ and are labelled according to an unknown function g. The domain of g is spanned by m independent (or predictor) random variables X_i (both numerical and categorical), that is $\mathbf{X}$=$\mathrm{X}_1 \times \mathrm{X}_2 \times \ldots \times \mathrm{X}_m$, while the range of g is a finite set Y=$\{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_L\}$, where each C_i is a distinct class. An inductive learning algorithm takes a training sample S=$\{(\mathbf{x}, \mathrm{y}) \in \mathbf{X} \times \mathrm{Y} \mid \mathrm{y}=\mathrm{g}(\mathbf{x})\}$ as input, and returns a function f which is hopefully close to g on the domain $\mathbf{X}$.

This problem has received much attention in the literature [15]. A well-known solution is represented by the Naïve Bayesian Classifiers [3], which aim to build a *discriminant function* d_i for each class, such that the value of the real number returned by $d_i(\mathrm{x})$ represents a degree of belief in assigning the observation x to the class C_i. The classification of any $x \in \mathbf{X}$ is the class C_k maximizing this degree of belief, that is:

$$f(\boldsymbol{x}) = arg\ max_i d_i(x)$$

In the Bayesian framework, $d_i(x)$ estimates the *posterior probability* $P(C_i|x)$ that the observation x is of class C_i. This estimation is based on the training set. By applying the Bayes theorem, $d_i(x)$ can be reformulated as follows:

A. Cappelli and F. Turini (Eds.): AI*IA 2003, LNAI 2829, pp. 250–261, 2003.

$$d_i(x) = P(C_i|x) = \frac{P(C_i)P(x|C_i)}{P(x)}$$

The term $P(x|C_i)$ is in turn estimated by means of the *naïve Bayes assumption.*

$P(x|C_i) = P(x_1, x_2, \ldots, x_m|C_i) = P(x_i|C_i) \times P(x_2|C_i) \times \ldots \times P(x_m|C_i)$

This assumption is clearly false if the predictor variables are statistically dependent. However, even in this case, the naïve Bayesian classifier can give good results [3].

Many inductive learning algorithms assume that the training set can be represented as a single table, where each row corresponds to an example and each column to a predictor variable or to the *target* variable Y. This assumption, also known as *single-table assumption* [20], seems quite restrictive in some data mining applications, where data are stored in a database and are organized into several tables for reasons of efficient storage and access. In this context, both predictor variables and the target variable are represented as attributes of distinct tables (relations).

Although in principle it is possible to consider a single relation reconstructed by performing a relational join operation on the tables, this approach is fraught with many difficulties in practice [2][10]. Consequently, the (multi-)relational data mining approach to the classification problem has been receiving considerable attention in the literature [1][9] [13][18][8].

In this paper we present a new approach to the problem of learning classifiers from relational data. It is based on the induction of a set of first-order classification rules in the context of naive Bayesian classification.

Studies on first-order naïve Bayes classifiers have already been reported in the literature. In particular, Pompe and Kononenko [18] proposed a method based on a two-step process. The first step uses the ILP-R system [17] to learn a hypothesis in the form of a set of first-order rules and then, in the second step, the rules are probabilistically analyzed. During the classification phase, the conditional probability distributions of individual rules are combined naïvely according to the naïve Bayesian formula.

Flach and Lachiche proposed a similar two-step method, however, unlike the previous one, there is no learning of first-order rules in the first step. Alternatively, a set of patterns (first-order conditions) is generated that are used afterwards as attributes in a classical attribute-value naive Bayesian classifier [8]. 1BC, the system implementing this method, views individuals as structured objects and distinguishes between *structural* predicates referring to parts of individuals (e.g. atoms within molecules), and *properties* applying to the individual or one or several of its parts (e.g. a bond between two atoms). An *elementary first-order feature* consists of zero or more structural predicates and one property.

An important aspect of the two (*eager*) approaches is that they keep the phases of first-order rules/conditions generation and of probability estimation separate. In particular, Pompe and Kononenko use ILP-R to induce first-order rules [17] while 1BC uses TERTIUS [7] to generate first order features. Then, the probabilities are computed for each first-order rule or feature. In the classification phase, the two approaches are similar to a multiple classifier because

they combine the results of two algorithms. However, most first-order features or rules share some literals and this approach takes into account the related probabilities more than once. To overcome this problem it is necessary to rely on an integrated approach, so that the computation of probabilities on shared literals can be separated from the computation of probabilities on the remaining literals.

Systems implementing one of the two above approaches work on a set of main-memory Prolog facts. In real-world applications, where facts correspond to tuples stored on relational databases, some pre-processing is required in order to transform tuples into facts. However, this has some disadvantages. First, the set of possible joins between relations in the database constrain the hypothesis space H, but for practical reasons after pre-processing only part of the original space Hwill be represented in the transformed space H', so that the best hypothesis in Hmay not be part of H'. Conversely, much of the pre-processing, which is often expensive in terms of computation and storage, may be unnecessary, since that part of the hypothesis space may never be explored, perhaps because of early pruning. Second, in applications where data can frequently change, preprocessing has to be frequently repeated. Finally, database schemas provide the learning system, free of charge, with useful knowledge of a data model that can help to guide the search process. This is an alternative to asking the users to specify a language bias, such as in 1BC.

In this paper the system Mr-SBC (Multi-Relational Structural Bayesian Classifier) is presented. It implements a new learning algorithm based on an integrated approach of first-order classification rules with naive Bayesian classification, in order to separate the computation of probabilities of shared literals from the computation of probabilities for the remaining literals. Moreover, Mr-SBC is tightly integrated with a relational database, and handles categorical as well as numerical data through a discretization method.

The paper is organized as follows. In the next section the problem is introduced and defined. The induction of first-order classification rules is presented in Section 3, the discretization method is explained in Section 4 and the classification model is illustrated in Section 5. Finally, experimental results are reported in Section 6 and some conclusions are drawn.

2 Problem Definition

In traditional classification systems that operate on a single relational table, an observation (or individual) is represented as a tuple of the relational table. Conversely, in Mr-SBC, which induces first-order classifiers from data stored in a set S=$\{\mathrm{T}_0,\mathrm{T}_1,\ldots,\mathrm{T}_h\}$ of tables of a relational database, an individual is a tuple t of a *target* relation T joined with all the tuples in S which are related to t following a foreign key path. Formally, a foreign key path is defined as follows:

Definition 1. *A foreign key path* is an ordered sequence of tables $\vartheta = (T_{i_1}, T_{i_2}, \ldots, T_{i_s})$, where

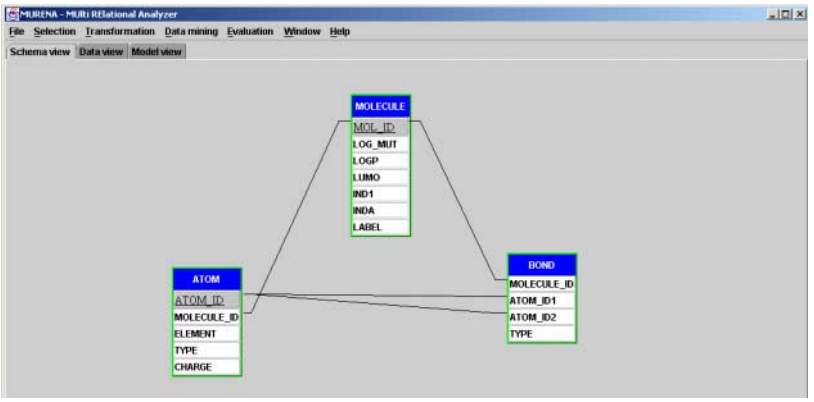

Fig. 1. An example of a relational representation of training data of the Mutagenesis database

– ∀j=1, . . . , s, $T_{i_j} \in S$
– ∀j=1, . . . , s-1, $T_{i_{j+1}}$ has a foreign key to the table T_{i_j}.

In Figure 1 an example of foreign key paths is reported. In this case, S={MOLECULE, ATOM, BOND} and the foreign keys are: A_M_FK, B_M_FK, A_A_FK1, A_A_FK1. If the target relation T is MOLECULE then the possible foreign key paths are:

(MOLECULE)
(MOLECULE,ATOM)
(MOLECULE,BOND)
(MOLECULE, ATOM, BOND)
(MOLECULE, ATOM, BOND)

The last two are equal because the bond table has two foreign keys referencing the table atom.

A formal definition of the learning problem solved by MR-SBC is the following problem:

Given:

– A *training set* represented by means of h relational tables S={T_0,T_1,. . . ,T_h} of a relational database D.
– A set of *primary key constraints* on tables in S.
– A set of *foreign key constraints* on tables in S.
– A *target relation* $T(\mathrm{x}_1,\ldots,\mathrm{x}_n) \in S$
– a *target* discrete *attribute* y in T, different from the primary key of T.

Find:

A naive Bayesian classifier which predicts the value of y for some individual represented as a tuple in T (with possibly UNKNOWN value for y) and related tuples in S according to foreign key paths.

3 Generation of First-Order Rules

Let R' be a set of first-order classification rules for the classes $\{C_1, C_2, \ldots, C_L\}$, and I an individual to be classified and defined as above. The individual can be logically represented as a set of ground facts, the only exception being the fact associated to the target relation T, where the argument corresponding to the target attribute y is a variable Y. A rule $R_j \in R$' *covers* I, if a substitution θ exists, such that $R_j\theta \subseteq I\theta$. The application of the substitution to I is required to ground the only variable Y in I to the same constant as that reported in R_j for the target attribute. Let R be the subset of rules in R' that cover I, that is $R=\{R_j \in R' \mid R_j \text{ covers } I\}$. The first-order naïve Bayes classifier for the individual I, $f(I)$, is defined as follows:

$f(I)= arg\ max_i P(C_i|R) = arg\ max_i \frac{P(C_i)P(R|C_i)}{P(R)}$

The value $P(C_i)$ is the prior probability of the class C_i. Since $P(R)$ is independent of the class C_i, it does not affect $f(I)$, that is,

$$f(I)= arg\ max_i \quad P(C_i)P(R|C_i) \tag{1}$$

The computation of $P(R|C_i)$ depends on the structure of R. Therefore, it is important to clarify how first-order rules are built in order to associate them with a probability measure. As already pointed out, Pompe and Kononenko use the first-order learning system ILP-R to induce the set of rules R'. This approach is very expensive and does not take into account the bias automatically determined by the constraints in the database. On the other hand, Flach and Lachiche use Tertius to determine the structure of *first-order features* on the basis of the structure of the individuals. The system Tertius deals with learning first-order logic rules from data lacking an explicit classification predicate. Consequently, the learned rules are not restricted to predicate definitions as in supervised inductive logic programming. Our solution is similar to that proposed by Flach since the structure of classification rules is determined on the basis of the structure of the individuals. The main difference is that the classification predicate is considered during the generation of the rules.

All predicates in classification rules generated by Mr-SBC are binary and can be of two different types.

Definition 2. A binary predicate p is a *structural* predicate associated to a table $T_i \in S$ if a foreign key FK in T_i exists that references a table $T_{i1} \in S$. The first argument of p represents the primary key of T_{i1} and the second argument represents the primary key of T_i.

Definition 3. A binary predicate p is a *property* predicate associated to a table $T_i \in S$, if the first argument of p represents the primary key of T_i and the second argument represents another attribute in T_i which is neither the primary key of T_i nor a foreign key in T_i.

Definition 4. A first order classification rule associated to the *foreign key path* ϑ is a clause in the form:
$p_0(A_1,y)$:- $p_1(A_1,A_2)$, $p_2(A_2,A_3)$, ..., $p_{s-1}(A_{s-1},A_s)$, $p_s(A_s,c)$.
where

1. p_0 is a property predicate associated to the target table T and to the target attribute y.
2. $\vartheta = (T_{i_1}, T_{i_2}, \ldots, T_{i_s})$ is a *foreign key path* such that for each k=1, ..., s-1: p_k is a structural predicate associated to the table T_{i_k}
3. p_s is a property predicate associated to the table T_{i_s}.

An example of a first-order rule is the following:
molecule_Label(A, active) :- molecule_Atom(A,B), atom_Type(B,'[22..27]').

Mr-SBC searches all possible classification rules by means of a breadth-first strategy and iterates over some refining steps. A refining step is biased by the possible foreign key paths and consists of the addition of a new literal, the unification of two variables and, in the case of a property predicate, the instantiation of a variable.

The search strategy is biased by the structure of the database because each refining step is made only if the generated first-order classification rule can be associated to a foreign key path. However, the number of refinement steps is upper bounded by a user-defined constant MAX_LEN_PATH.

4 Discretization

In Mr-SBC continuous attributes are handled through supervised discretization. Supervised discretization methods utilize the information on the class labels of individuals to partition a numerical interval into bins. The proposed algorithm sorts the observed values of a continuous feature and attempts to greedily divide the domain of the continuous variable into bins, such that each bin contains only instances of one class. Since such a scheme could possibly lead to one bin for each observed real value, the algorithm is constrained to merge bins in a second step. Merging of two contiguous bins is performed when the increase of entropy is lower than a user-defined threshold (MAX_GAIN). This method is a variant of the one-step method 1RD by Holte [12] for the induction of one-level decision trees, that proved to work well with the Naïve Bayes Classifier [4]. It is also different from the one-step method by Fayyad and Irani [5] that recursively splits the initial interval according to the class information entropy measure until a stopping criterion based on the Minimum Description Length (MDL) principle is verified.

5 The Computation of Probabilities

According to the naïve Bayes assumption, the attributes are considered independent. However, this assumption is clearly false for the attributes that are

primary keys or foreign keys. This means that the computation of $P(R|C_i)$ in equation (1) depends on the structures of rules in R.

For instance, if R_1 and R_2 are two rules of class C_i, that share the same structure and differ only for the property predicates in their bodies

R_1: $\beta_{1,0} : -\beta_{1,1}, ..., \beta_{1,K_1-1}, \beta_{1,K_1}$

R_2: $\beta_{2,0} : -\beta_{2,1}, ..., \beta_{2,K_2-1}, \beta_{2,K_2}$

where

$$K_1 = K_2 \text{ and } \beta_{1,1} = \beta_{2,1}, \beta_{1,2} = \beta_{2,2}, ..., \beta_{1,K_1-1} = \beta_{2,K_2-1}$$

then $P(\{R_1, R_2\}|C_i) = P(\beta_{1,0} \cap (\beta_{1,1}, ..., \beta_{1,K_1-1}) \cap \beta_{1,K_1} \cap \beta_{2,K_2}|C_i)=$

$$P(\beta_{1,0} \cap (\beta_{1,1}, ..., \beta_{1,K_1-1})|C_i) \cdot P(\beta_{1,K_1} \cap \beta_{2,K_2}|\beta_{1,0} \cap (\beta_{1,1}, ..., \beta_{1,K_1-1}) \cap C_i)$$

The first term takes into account the structure common to both rules, while the second term refers to the conditional probability of satisfying the property predicates in the rules given the common structure.

The latter probability can be factorized under the naïve Bayes assumption, that is:

$$P(\beta_{1,K_1} \cap \beta_{2,K_2}|\beta_{1,0} \cap (\beta_{1,1}, ..., \beta_{1,K_1-1}) \cap C_i) =$$

$$P(\beta_{1,K_1}|\beta_{1,0} \cap (\beta_{1,1}, ..., \beta_{1,K_1-1}) \cap C_i) \cdot P(\beta_{2,K_2}|\beta_{1,0} \cap (\beta_{1,1}, ..., \beta_{1,K_1-1}) \cap C_i)$$

According to this approach the conditional probability of the structure is computed only once. This approach differs from that proposed in the works of Pompe and Kononenko [18] and Flach [8] where the factorization would multiply the structure probability twice.

By generalizing to a set of classification rules we have:

$$P(C_i)P(R|C_i) = P(C_i)P(structure)\prod_j P(R_j|structure) \tag{2}$$

where the term *structure* takes into account the class C_i and the structures of the rules in R.

If the classification rule $R_j \in R$ is in the form $\beta_{j,0} : -\beta_{j,1}, ..., \beta_{j,K_j-1}, \beta_{j,K_j}$ where $\beta_{j,0}$and $\beta_{j,K_j}$are property predicates and $\beta_{j,1}, \beta_{j,2}, ..., \beta_{j,K_j-1}$are structural predicates, then:

$$P(R_j|structure) = P(\beta_{j,K_j}|\beta_{j,0}, \beta_{j,1}, ..., \beta_{j,K_j-1}) = P(\beta_{j,K_j}|C_i, \beta_{j,1}, ..., \beta_{j,K_j-1})$$

where C_i is the value of the target attribute in the head of the clause ($\beta_{j,0}$). To compute this probability, we use the Laplace estimation:

$$P(\beta_{j,K_j}|C_i, \beta_{j,1}, ..., \beta_{i,K_j-1}) = \frac{\#(\beta_{j,K_j}, C_i, \beta_{j,1}, ..., \beta_{j,K_j-1}) + 1}{\#(C_i, \beta_{j,1}, ..., \beta_{j,K_j-1}) + F}$$

where F is the number of possible values of the attribute to which the β_{j,K_j} property predicate is associated. Laplace's estimate is used in order

to avoid null probabilities in the equation (2). In practice, the value at the nominator is the number of individuals which satisfy that conjunction $\beta_{j,K_j}, C_i, \beta_{j,1}, ..., \beta_{j,K_j-1}$, in other words, the number of individuals covered by the rule $\beta_{j,0} : -\beta_{j,1}, ..., \beta_{j,K_j-1}, \beta_{j,K_j}$. It is determined by a "select count (*)" SQL instruction. The value of the denominator is the number of individuals covered by the rule $\beta_{j,0} : -\beta_{j,1}, ..., \beta_{j,K_j-1}$.

The term *P(structure)* in the equation (2) is computed as follows:
Let B=$\{(\beta_{j,1},\beta_{j,2},\ldots,\beta_{j,t})|\ j$=1..$s$ and t=1, ...,$K_j - 1\}$ the set of all distinct sequences of structural predicates in the rules of R. Then

$$P(structure) = \prod_{seq \in \mathrm{B}} P(seq) \tag{3}$$

To compute *P(seq)* it is necessary to introduce the definition of the probability JP that a join query is satisfied [11]. Let ϑ=(T_{i_1}, T_{i_2},..., T_{i_s}) be a *Foreign Key Path*, then:

$$\mathrm{JP}(\vartheta) = \mathrm{JP}(\mathrm{T}_{i_1}, ..., \mathrm{T}_{i_s}) = \frac{|\triangleright\triangleleft(\mathrm{T}_{i_1} \times ... \times \mathrm{T}_{i_s})|}{|\ \mathrm{T}_{i_1}| \times ... \times |\mathrm{T}_{i_s}|}$$

where $\triangleright\triangleleft(\mathrm{T}_{i_1} \times ... \times \mathrm{T}_{i_s})$ is the result of the join between the tables $\mathrm{T}_{i_1}, ..., \mathrm{T}_{i_s}$.

We must remember that each sequence *seq* is associated to a foreign key path ϑ. If *seq*=($\beta_{j,1}$,$\beta_{j,2}$, ...,$\beta_{j,t}$) there are two possibilities: either a prefix of *seq* is in B or not. By denoting as T_{j_h} the table related to $\beta_{j,h}$, h=1,..., t, the probability $P(seq)$ can be recursively defined as follows:

$$P(seq) = \begin{cases} \mathrm{JP}(\mathrm{T}_{j_1}, ..., \mathrm{T}_{j_t}) \text{ if } seq \text{ has no prefix in B} \\ \frac{\mathrm{JP}(\mathrm{T}_{j_1}, ..., \mathrm{T}_{j_t})}{P(seq')} \quad \text{if } seq' \text{ is the longest prefix of } seq \text{ in B} \end{cases}$$

This formulation is necessary in order to compute the formula (3) considering both dependent and independent events.

Since *P(structure)* takes into account the class, *P(seq)* is computed separately for each class.

6 Experimental Results

MR-SBC has been implemented as a module of the system MURENA and has been empirically evaluated on the Mutagenesis datasets. Mutagenesis datasets have been taken from the MLNET repository (`http://www.mlnet.org/`). These datasets are relative to the problem of identifying the mutagenic compounds [16] and have been extensively used to test both inductive logic programming (ILP) systems and (multi-)relational mining systems. We considered, analogously to related experiments in the literature, the "regression friendly" dataset of 188 elements.

A recent study on this database [19] recognizes five levels of background knowledge for mutagenesis which can provide more detailed descriptions of the

Table 1. Background knowledge for Mutagenesis database

Background	**Description**
BK_0	Consists of those data obtained with the molecular modelling package QUANTA. For each compound it obtains the atoms, bonds, bond types, atom types, and partial charges on atoms.
BK_1	Consists of Definitions in BK_0 plus indicators *ind1*, and *inda* in molecule table.
BK_2	Variables (attributes) *logp*, and *lumo* are added to definitions in BK_1.

examples. In this study we used only the first three levels of background knowledge in order to compare the performance of Mr-SBC with other methods for which experimental results are available in the literature. Table 1 shows the first three sets of background knowledge used in our experiments, where $BK_i \subseteq BK_{i+1}$ for i=0, . . . , 2. The greater the BK, the more complex the learning problem.

The dataset is analyzed by means of a 10-fold cross-validation, that is, the target table is firstly divided into ten blocks of near-equal size and distribution of class values, and then, for every block, a subset of tuples in S related to the tuples in the target table block are extracted. In this way, ten databases are created. Mr-SBC is trained on nine databases and tested on the hold-out database. Mr-SBC has been executed with the following parameters: MAX_LEN_PATH=4 and MAX_GAIN= 0.5.

Experimental results on predictive accuracy are reported in Table 2 for increasing complexity of the models. A comparison to other results reported in the literature is also made. Mr-SBC has the best performance for the most complex task (BK_2) with an accuracy of almost 90%, while it ranks third for the simplest task. It is interesting to note that the predictive accuracy increases with the complexity of the background knowledge.

Table 2. Accuracy comparison on the set of 188 regression friendly elements of Mutagenesis. Results for Progol2, Foil, Tilde are taken from [1]. Results for Progol_1 are taken from [19]. The results for 1BC are taken from [6]. Results for MRDTL are taken from [14]. The values are the results of 10-fold cross-validation

System	textitAccuracy(%)		
	BK_0	BK_1	BK_2
Progol_1	79	86	86
Progol_2	76	81	86
Foil	61	61	83
Tilde	75	79	85
MRDTL	67	**87**	88
1BC	**80.3**	—	87.2
Mr-SBC	76.5	81	**89.9**

Table 3. Time comparison of the set of 188 regression friendly elements of Mutagenesis. Results for Progol2, Foil, Tilde are taken from [1]. Results for Progol_1 are taken from [19]. Results for MRDTL are taken from [14]. The results of MR-SBC have been taken on a PIII WIN2k platform

System	*Time (Secs)*		
	BK_0	BK_1	BK_2
Progol_1	8695	4627	4974
Progol_2	117000	64000	42000
Foil	4950	9138	0.5
Tilde	41	170	142
MRDTL	0.85	170	142
1BC	–	–	–
MR-SBC	36	42	48

Table 4. Number of induced rules for the Mutagenesis database

FOLD	*Number of rules*		
	BK_0	BK_1	BK_2
1	57	61	65
2	56	60	67
3	56	60	64
4	56	60	64
5	57	61	65
6	53	57	61
7	57	61	65
8	57	61	65
9	54	58	68
10	56	60	64
Av. Number	55,9	59,9	64,8

As regards execution time (see Table 3). The time required by Mr-SBC increases with the complexity of the background knowledge. Mr-SBC is generally considerably faster than competing systems, such as Progol, Foil, Tilde and 1BC, that do not operate on data stored in a database. Moreover, except for the task BK_0, Mr-SBC performs better that MRDTL which works on a database. It is noteworthy that the trade-off between accuracy and complexity is in favour of Mr-SBC.

In Table 4 the number of extracted rules for each fold is shown. The number of rules is quite high, although some of them are either redundant or cover very few individuals. Therefore, some additional stopping criteria are required to avoid the generation of these rules and to reduce further the cost complexity of the algorithm.

7 Conclusion

In the paper, a multi-relational extension of the naïve Bayes classification method is described. It presents several differences with respect to related works. First, it is based on an integrated approach, so that the contribution of literals shared by several rules to the posterior probability is computed only once. Second, it works both on discrete and continuous attributes. Third, the generation of rules is based on the knowledge of a data model embedded in the database schema. The proposed method has been implemented in the new system Mr-SBC and tested on four benchmark tasks. Results on predictive accuracy are in favour of our system for the most complex tasks. Mr-SBC also proved to be a very efficient multi-relational data mining system with a tight integration to a relational DBMS.

As future work, we plan to extend the comparison of Mr-SBC to other multi-relational data mining systems on a larger set of benchmark datasets. Moreover, we intend to frame the proposed method in a transduction inference setting where both labelled and unlabelled data are available for training.

Acknowledgements

This work has been supported by the annual Scientific Research Project "Scoperta di conoscenza in basi di dati: metodi e tecniche efficienti e robuste per dati complessi" Year 2002 funded by the University of Bari.

References

[1] H. Blockeel. *Top-down induction of first order logical decision trees*. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, 1998. 251, 258, 259

[2] L. De Raedt. Attribute-value learning versus inductive logic programming: the missing links. In *Proceedings of the 8 th International Conference on Inductive Logic Programming*, pages 128–137. Springer-Verlag, 1998. 251

[3] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997. 250, 251

[4] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*, pages 194–202, 1995. 255

[5] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *In Proc. Of the 13th International Joint Conference on Artificial Intelligence*, pages 1022–1027, 1994. 255

[6] P. Flach and N. Lachiche. First-order bayesian classification with 1BC submitted http://hydria.u-strasbg.fr/ lachiche/1bc.ps.gz, 2000. 258

[7] P. A. Flach and N. Lachiche. Confirmation-guided discovery of first-order rules with tertius. *Machine Learning*, 42(1/2):61–95, 2000. 251

[8] P. A. Flach and N. Lachiche. Decomposing probability distributions on structured individuals. In P. Brito, J. Costa, and Donato Malerba, editors, *Proceedings of the ECML2000 workshop on Dealing with Structured Data in Machine Learning and Statistics*, pages 33–43, Barcelona, Spain, 2000. 251, 256

[9] Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In Morgan Kaufman, editor, *In Proceedings of the 6 th International Joint Conference on Artificial Intelligence*, pages 1300–1309, 1999. 251

[10] L. Getoor. Multi-relational data mining using probabilistic relational models: research summary. In A. J. Knobbe and D. M. G. van der Wallen, editors, *In Proceedings of the First Workshop in Multi-relational Data Mining*, pages 1300–1309, 2001. 251

[11] L. Getoor, D. Koller, and B. Taskar. Statistical models for relational data. In *In Proceedings of the KDD-2002 Workshop on Multi-Relational Data Mining*, pages 36–55, Edmonton, CA, 2002. 257

[12] R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, (11):63–90, 1993. 255

[13] M. Krogel and S. Wrobel. Transformation-based learning using multirelational aggregation. In Céline Rouveirol and Michèle Sebag, editors, *Proceedings of the 11 th International Conference on Inductive Logic Programming LNAI*, volume 2157. Springer-Verlag, 2001. 251

[14] H. A. Leiva. *MRDTL: A multi-relational decision tree learning algorithm.* PhD thesis, Master thesis, University of Iowa, 2002. 258, 259

[15] T. Mitchell. *Machine Learning.* McGraw Hill, 1997. 250

[16] S. H. Muggleton, M. Bain, J. Hayes-Michie, and D. Michie. An experimental comparison of human and machine learning formalisms. In *In Proc. Sixth International Workshop on Machine Learning*, pages 113–118, San Mateo, CA, 1989. Morgan Kaufmann. 257

[17] U. Pompe and I. Kononenko. Linear space induction in first order logic with relief. In R. Viertl. & G. Della Riccia R. Kruse, editor, *CISM Lecture Notes.* Udine, Italy, 1994. 251

[18] U. Pompe and I. Kononenko. Naive bayesian classifier within ILP-R. In L. De Raedt, editor, *In Proc. of the 5th Int. Workshop on Inductive Logic Programming*, pages 417–436, Dept. of Computer Science, Katholieke Universiteit Leuven, 1995. 251, 256

[19] A. Srinivasan and R. D. Kingand S. Muggleton. The role of background knowledge: using a problem from chemistry to examine the performance of an ilp program, 1999. 257, 258, 259

[20] S. Wrobel. *Relational Data Mining S. Džeroski and N. Lavrac*, chapter Inductive logic programming for knowledge discovery in databases, pages 74–101. Springer, 2001. 251