Itemset-Based Variable Construction in Multi-relational Supervised Learning

Dhafer Lahbib[1], Marc Boullé[1], and Dominique Laurent[2]

[1] Orange Labs - 2, avenue Pierre Marzin, 23300 Lannion
{dhafer.lahbib,marc.boulle}@orange.com
[2] ETIS-CNRS-Université de Cergy Pontoise-ENSEA, 95000 Cergy Pontoise
dominique.laurent@u-cergy.fr

Abstract. In multi-relational data mining, data are represented in a relational form where the individuals of the target table are potentially related to several records in secondary tables in one-to-many relationship. In this paper, we introduce an itemset based framework for constructing variables in secondary tables and evaluating their conditional information for the supervised classification task. We introduce a space of itemset based models in the secondary table and conditional density estimation of the related constructed variables. A prior distribution is defined on this model space, resulting in a parameter-free criterion to assess the relevance of the constructed variables. A greedy algorithm is then proposed in order to explore the space of the considered itemsets. Experiments on multi-relationalal datasets confirm the advantage of the approach.

Keywords: Supervised Learning, Multi-Relational Data Mining, one-to-many relationship, variable selection, variable construction.

1 Introduction

Most of existing data mining algorithms are based on an attribute-value representation. In this flat format, each record represents an individual and the columns represent variables describing these individuals. In real life applications, data usually present an intrinsic structure which is hard to express in a tabular format. This structure may be naturally described using the relational formalism where each object (target table record) refers to one or more records in other tables (secondary tables) through a foreign key.

Example 1. In the context of a Customer Relationship Management (CRM) problem, Figure 1 shows an extract of a virtual CRM relational database schema. The table *Customer* is the target table, whereas *Call detail record (CDR)* is a secondary table related to *Customer* through the foreign key CID. The problem may be, for instance, to identify the customers likely to be interested in a certain product. This problem turns into a classification problem where the target variable is the variable *Appetency*, which denotes whether the customer is likely to order a particular product.

F. Riguzzi and F. Železný (Eds.): ILP 2012, LNAI 7842, pp. 130–150, 2013.

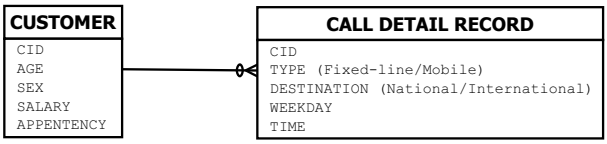

Fig. 1. Relational schema of a CRM database

Learning from relational data has recently received increasing attention in the literature. The term Multi-Relational Data Mining (MRDM) was initially introduced by [1] to address novel knowledge discovery techniques from multiple relational tables. The common point between these techniques is that they need to transform the relational representation. In Inductive Logic Programming (ILP) [2], data is recoded as logic formulas. Other methods known as propositionalisation [3] flatten the relational data by creating new variables.

Our goal in this article is to directly exploit the informativeness of secondary variables w.r.t. the target variable. We propose a multivariate pre-processing of secondary variables in order to construct multivariate itemsets from secondary tables. This pre-processing consists in a discretization in the numerical case and a value grouping in the categorical case. To the best of our knowledge, only few studies have considered the variable pre-processing problem within the multi-relational setting with one-to-many relationship (in particular, discretization and variables selection) [4,5,6]. In these approaches, secondary variables are considered independently from each other, which does not make it possible to take into account their correlation to predict the class label.

In our approach, the considered itemsets are in secondary tables while the class labels are in the target one. In order to evaluate these itemsets and exploit their information for classification, we construct new binary variables in the secondary tables. We propose a conditional density estimation of the constructed variables in order to extend the Naive Bayes classifier to multi-relational data, and we define a prior distribution on the itemsets model space. As a result, we obtain a parameter-free relevance criterion for the constructed variables.

Example 2. Given our CRM example, let us consider the following itemset π in the CDR secondary table: $(WeekDay \in \{Saturday\}) \wedge (10:00:00 \leq Time < 11:30:00) \wedge (Destination \in \{International\})$ where $WeekDay$ and $Destination$ are categorical variables and $Time$ is a numerical variable. This itemset allows constructing a new binary variable in the secondary table, according to whether the secondary records are covered or not by the itemset. For example, the secondary table record ("$C901$", "$Mobile$", "$International$", "$Saturday$", "10 : 30 : 00") is covered by π and therefore the value of A_π for that record is "1".

Computing features as a pre-processing step is a classical solution in MRDM in order to be able to use a propositional classifier. In the 1BC system, [7] compute a set of conjunctive patterns consisting of first-order conditions which are used as features in a classical Naive Bayes classifier. 1BC2 [8] and Mr-SBC [9]

extend this approach with more accurate estimation of conditional probabilities and with improved results. It is worthy of mention that our method is not a propositionalisation approach [10]. The new binary variables are created in the secondary table, not in the target one. Their conditional probability is estimated directly by using the multi-relational approach introduced in [11].

The remaininder of this paper is organized as follows. In the next section, the present work is motivated and related to alternative approaches. Section 3 recalls the method [11] exploited to estimate the conditional probability of a binary secondary variable. Section 4 introduces the space of constructed itemset-based secondary variables and presents their evaluation criterion. This section also gives a heuristic algorithm in order to explore the itemset space. In Section 5, an experimental evaluation of the proposed approach on real-world multi-relational datasets is reported. Finally, Section 6 gives a summary and discusses future work.

2 Motivation and Related Work

Classifying data scattered over the multiple tables of a relational database has recently received a growing attention within the data mining community. In this paper we are interested in classifying individuals contained in a target table with a one-to-many relationship with secondary tables.

The novelty of this multi-relational setting, compared to classical attribute-value methods, consists in exploiting the predictive power of secondary variables belonging to secondary tables. The difficulty when dealing with these variables arises from the presence of one-to-many associations. In the attribute-value single table case, each individual has a single value per variable, while in multiple table setting, for a secondary variable, an individual may have a set of values (possibly empty) of varying size.

2.1 Motivation

The idea behind using itemsets on non target tables is to discover multivariate patterns between secondary variables in order to detect significant differences between individuals of distinct classes. In this paper we propose to use a multivariate approach. Instead of considering only one variable at a time, we introduce itemsets of secondary variables in order to take into account correlations between these variables w.r.t. to the target variable. In some problems, a single secondary variable may not be relevant to predict the class label, whereas several secondary variables considered jointly may help predicting the target variable.

Example 3. Let us consider the Digits[1] dataset where the task is to recognize handwritten digits (classes are digits from 0 to 9) [12]. In its original version, this dataset has an attribute-value tabular format where each line represents an image with 28×28 pixels. This database can be represented in a relational format

[1] Available at `http://yann.lecun.com/exdb/mnist/`

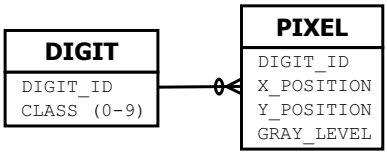

Fig. 2. Relational schema of the Digits database

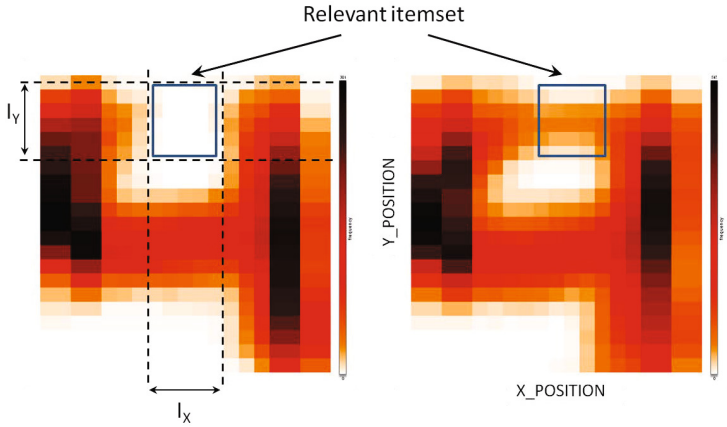

Fig. 3. Relevant itemset in the table Pixel of the Digits database

composed of two tables: a target table Digit as well as a secondary table Pixel which describes pixels composing each image. Figure 2 shows the corresponding relational schema. A handwritten digit is then associated with 784 lines in the table Pixel. A pixel is described by three secondary variables: *Gray_level* as well as *X_position* and *Y_position* which represent the position of the pixel in the original image.

The problem turns into a multi-relational classification problem. It is clear from Figure 3 that identifying the digit represented by the handwritten character (for example whether it is a 9 or a 4) requires taking into account simultaneously the values of the three secondary variables: *Gray_level* , *X_position* and *Y_position*. The itemset

$$\pi : (X\_position \in I_X) \wedge (Y\_position \in I_Y) \wedge (Gray\_level > 0)$$

gives a discriminant pattern characterizing a 9 or a 4 digit. The binary secondary variable which denotes whether a pixel is covered or not by this itemset can accurately predict the class.

2.2 Related Work

Mining itemsets of secondary variables can be seen as a descriptive task which joins many studies on association rules and frequent patterns. Classical solutions

assume that data are stored in a single attribute-value data table. But many attempts have been proposed recently to deal with relational data.

First multi-relational association rules and frequent patterns are based on ILP in order to discover frequent Prolog queries ([13,14]) or frequent predicates ([15]). These methods need to transform the initial relational database into a deductive one. Furthermore, they have a high complexity and the mined patterns may be difficult to understand [16]. Some other approaches use a classical attribute-value association rule algorithm on a single table obtained by joining all the tables in order to generate a cross table [16] or by propagating the target variable to the secondary tables [17]. Such transformations may lead to statistical skews since individuals with a large number of related records in a secondary table will be overestimated thereby causing overfitting.

Beside descriptive tasks [18,19], association rules have been proposed for a classification purpose. [20] investigated the use of logical association rules in order to classify spatial data. Discrimininant features are generated based on these rules and are exploited for propositionalization and to propose an extension of the Naive Bayes classifier. The proposed approach is limited to spatial relational data described by a hierarchy of concepts.

[21] used emergent patterns over multiple tables in order to extend the Naive Bayes classifier to relational data. The considered emergent patterns are conjunctions of logical predicates modeling properties of the relational objects and associations between them. Using these patterns, the authors propose a decomposition of the posterior probability based on the naive Bayes assumption to simplify the probability estimation problem. The problem of this approach is that it suffers from a high number of considered emerging patterns and scalability limits since it is based on logical inference.

We notice that in this article we restrict ourselves to secondary variables located in tables with direct one-to-many relationships with the target table. More generally, we consider a star schema with a central target table related to secondary tables. The second level of one-to-many relations is intrinsically challenging and is left for future work.

3 Evaluation of Binary Secondary Variables

In this section, we summarize the method introduced in [11] to evaluate the relevance of a binary secondary variable A with values a and b.

3.1 Binary Secondary Variable Evaluation

In this case, each individual of the target table is described by a bag of secondary values among a and b, and summarized without loss of information by the numbers n_a of a and n_b of b. Thus, the whole information about the initial secondary variable can be captured by considering jointly the pair (n_a, n_b) of primary variables. We emphasize that the two variables are considered jointly so that information is preserved, as illustrated in Figure 4.

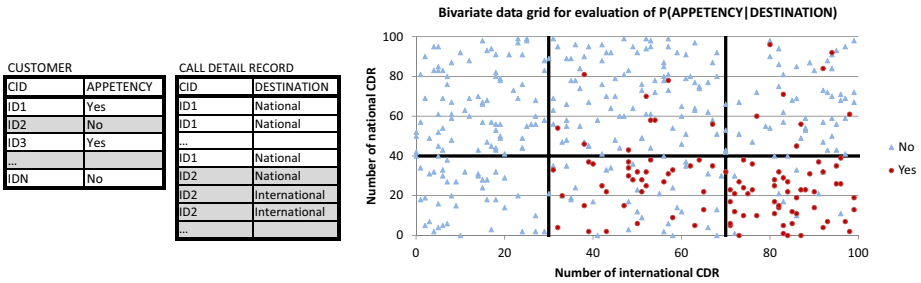

Fig. 4. Evaluation of the secondary variable *Destination* for the prediction of the target variable *Appetency*. Here, customers with a small number of national CDR and a large number of international CDR are likely to have the value *Yes* of *Appetency*.

By doing so, the conditional probability $P(Y \mid A)$ is equivalent to the probability $P(Y \mid n_a, n_b)$. Bivariate data grid models [22] are used to qualify the information contained in the pair (n_a, n_b). The couple of numeric variables are partitioned jointly into intervals. Individuals are then partitioned into a data grid whose cells are defined by interval pairs, and the target variable distribution is defined locally in each cell. Therefore, the purpose is to find the optimal bivariate discretization which maximizes the class distribution, in other words, the purpose is to obtain the optimal grid with homogeneous cells according to the class values. Applying a Bayesian model selection approach, a criterion $c_e(A)$ is obtained to assess the relevance of a secondary binary variable.

Notation 1.

- N : *number of individuals (number of target table records)*
- J : *number of target values (classes),*
- I_a,I_b : *number of discretization intervals respectively for* n_a *and* n_b
- $N_{i_a..}$: *number of individuals in the interval* i_a $(1 \leq i_a \leq I_a)$ *for variable* n_a
- $N_{.i_b.}$: *number of individuals in the interval* i_b $(1 \leq i_b \leq I_b)$ *for variable* n_b
- $N_{i_a i_b.}$: *number of individuals in the cell* (i_a, i_b)
- $N_{i_a i_b j}$: *number of individuals in the cell* (i_a, i_b) *for the target value* j

$$c_e(A) = \log N + \log N + \log \binom{N + I_a - 1}{I_a - 1} + \log \binom{N + I_b - 1}{I_b - 1} \tag{1}$$
$$+ \sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} \log \binom{N_{i_a i_b.} + J - 1}{J - 1} + \sum_{i_a=1}^{I_a} \sum_{i_b=1}^{I_b} \log \frac{N_{i_a i_b.}!}{N_{i_a i_b 1}! N_{i_a i_b 2}! \ldots N_{i_a i_b J}!}$$

The first five terms of Formula 1 stand for the prior probability: choosing the numbers of intervals, their frequencies and the distribution parameters for the target values in each grid cell[2]. The last term represents the conditional likelihood of the data given the model. The bivariate discretization criterion is

[2] Notation $\binom{n}{k}$ represents the binomial coefficient: number of k-combinations of n elements.

optimized starting from an initial random solution, using a bottom-up greedy heuristic. Pre and post-optimization steps are employed based on alternating partial optimizations per variable. The overall complexity of the algorithm is $O(JN \log(N))$ [22]. Details regarding this criterion and the optimization algorithm can be found in [22]. Beyond the evaluation of binary secondary variables, our goal is to extend the method to numerical and categorical secondary variables while capturing the potential correlations that may exist between secondary variables. In the next section, we introduce a similar criterion for itemset-based models defined over sets of secondary variables, in order to take into account this multivariate correlation.

3.2 Naive Bayes Extension

The constructed secondary variable is used to build a Naive Bayes classifier which aims to classify an object o by maximizing the posterior probability $P(Y_j \mid o)$ that o is of class Y_j. This probability can be reformulated by applying the Bayes rule:

$$P(Y_j \mid o) = \frac{P(Y_j)\, P(o \mid Y_j)}{P(o)} \tag{2}$$

$P(Y_j)$ is the prior probability of the class Y_j and the term $P(o \mid Y_j)$ is estimated by using the Naive Bayes assumption: if X_k are descriptive variables, then $P(o \mid Y_j) = P(X_1, X_2, \cdots, X_K \mid Y = j) = \prod_{k=1}^{K} P(X_k \mid Y_j)$.

The above formulation of the Naive Bayesian classifier is clearly limited to the attribute-value representation. In order to take into account the multi-table data, in particular the secondary variables, we need to assume that the secondary variables are independent given the target variable, and then estimate the conditional probabilities $P(X_k \mid Y_j)$ where X_k are secondary variables.

Estimating $P(X_k \mid Y_j)$ is equivalent to evaluating $P(n_a, n_b \mid Y_j)$ which is performed by simple counting locally in each cell of the optimal bivariate data grid. More explicitly, if for an object o in test, the corresponding cell is (i_a, i_b), and if N_j denotes the number of objects of class j, $P(X_k \mid Y_j)$ can be estimated as follows :

$$P(X_k \mid Y_j) = \frac{N_{i_a i_b j}}{N_j}, \tag{3}$$

where $N_{i_a i_b j}$ stands for the number of objects in the cell (i_a, i_b) for the target value j.

4 Itemset Based Variable Construction

In this section, we introduce a method to construct new binary variables in secondary tables, based on itemset models. We first introduce a model of itemset based secondary variables, then present a criterion to evaluate these constructed variables, and finally propose an algorithm to construct itemset based variables and evaluate them.

4.1 Variable Construction Model

Let us first introduce a model of itemset based variable construction in a secondary table. We exploit the model of [23] to define itemsets with numerical or categorical variables, where the itemset is defined by a conjunction of intervals in the numerical case and sets of values in the categorical one, for a secondary table with a one-to-many path from the target table.

Definition 1 (Itemset Based Construction Model). *An (IBCM) itemset π, at the basis of a secondary Boolean constructed variable A_π, is defined by:*

- *the secondary table with a one-to-many path from the target table,*
- *the constituent variables of the itemset,*
- *the group of values involved in the itemset, for each categorical variable of the itemset,*
- *the interval involved in the itemset, for each numerical variable of the itemset,*

where the value of A_π is true for secondary records that are covered by the itemset, false otherwise.

An example of itemset is provided in Example 2.

In [24] we considered any interval and groups of values for the constituents of itemsets. In the numerical case, a number of discretization intervals is chosen, then the bounds of the interval, and finally, the index of the interval belonging to the itemset. In the categorical case, a number of groups of values is chosen, then the partition of the values into groups and finally the index of the group belonging to the itemset. In this paper, we focus on quantile partitions for each secondary variable. Given a partition, a quantile is defined solely by an index, whereas an interval requires two bounds. This allows us to consider itemset models which are both more interpretable and parsimonious, and enables efficient optimization heuristics. An itemset is then defined by a choice of a quantile part for each constituent variable, where the quantile parts are themselves defined solely by a size of partition and a part index. In definitions 2 and 3, we precisely define quantile partitions both for numerical and categorical variables.

Definition 2 (Numerical quantile partition). *Let D be a dataset of N instances and X a numerical variable. Let $x_1, x_2, \ldots, x_N$ be the N sorted values of X in dataset D. For a given number of parts P, the dataset is divided into P equal frequency intervals $]-\infty, x_{\lfloor 1+\frac{N}{P} \rfloor}[$, $[x_{\lfloor 1+\frac{N}{P} \rfloor}, x_{\lfloor 1+2\frac{N}{P} \rfloor}[$, $\ldots$, $[x_{\lfloor 1+i\frac{N}{P} \rfloor}, x_{\lfloor 1+(i+1)\frac{N}{P} \rfloor}[$, $\ldots$, $[x_{\lfloor 1+(P-1)\frac{N}{P} \rfloor}, +\infty[$.*

Definition 3 (Categorical quantile partition). *Let D be a dataset of N instances and X a categorical variable with V values. For a given number of parts P, let $N_P = \lceil \frac{N}{P} \rceil$ be the expected minimum frequency per part. The categorical quantile partition into (at most) P parts is defined by singleton parts for each value of X with frequency beyond the threshold frequency N_P and a "garbage" part consisting of all values of X below the threshold frequency.*

Now, we can give a formal definition of IBCM itemset models, using the following notations.

Notation 2.

- $\mathcal{T} = \{T_1, T_2, \ldots\}$*: set of secondary tables having a one-to-many relation with the target table*
- $|\mathcal{T}|$*: number of secondary tables*
- $T \in \mathcal{T}$*: secondary table containing the variables which compose the itemset*
- N_s *: number of records in the secondary table* T
- m*: number of variables in the secondary table* T
- $X = \{x_1, \ldots, x_k\}$*: set of* k *variables of* T *which compose the itemset*
- I_x*: size of the quantile partition of variable* x
- i_x*: index of the quantile part of variable* x *involved in the itemset*

An IBCM itemset model π is then defined by the secondary table T, the set X of variables of T which compose the itemset, and for each constituent variable by the size I_x of the quantile partition and the index i_x of the quantile part involved in the itemset.

4.2 Evaluation of Constructed Variables

The new variable A_π that we have built is seen as a binary secondary variable which can be evaluated using the cost $c_e(A_\pi)$ of Formula 1. Let $c_e(\emptyset)$ be the null cost when no input variable is used estimate the target variable. This corresponds to a bivariate data grid with one single cell, whose cost is

$$c_e(\emptyset) = 2 \log N + \log \frac{N!}{N_1! N_2! \ldots N_J!}, \tag{4}$$

$$= N Ent(Y) + O(\log N), \tag{5}$$

where $Ent(Y)$ is the Shannon entropy of the target variable Y (cf. Formula 1 and using [25]). Therefore, any constructed variable with an evaluation cost beyond the null cost can be discarded, as being less informative than the target variable alone. When the number of constructed variables increases, the risk of wrong detection of informative variables grows. In order to prevent this risk of overfitting, we suggest to introduce a prior distribution over itemset based constructed variables, so as to get a construction cost $c_c(A_\pi)$ derived from a Bayesian approach. We then evaluate the overall relevance $c_r(A_\pi)$ of A_π by taking into account the construction cost $c_c(A_\pi)$ as well as the evaluation cost $c_e(A_\pi)$:

$$c_r(A_\pi) = c_c(A_\pi) + c_e(A_\pi). \tag{6}$$

4.3 Prior Distribution on Itemset Models

To apply the Bayesian approach, we need to define a prior distribution on the itemset based construction model space. We apply the following principles in order to guide the choice of the prior:

1. the prior is as flat as possible, in order to minimize the bias,
2. the prior exploits the hierarchy of the itemset models.

MODL hierarchical prior. We use the following distribution prior on IBCM itemsets, called the MODL hierarchical prior. Notice that a uniform distribution is used at each stage[3] of the parameters hierarchy of the IBCM models:

1. the itemset table T is uniformly distributed among the tables of $\mathcal{T}$

$$p(T) = \frac{1}{|\mathcal{T}|}. \tag{7}$$

2. the number of variables k in the itemset ($k \geq 0$) is distributed according to the universal prior for integers [4] [26]

$$p(k) = 2^{-L(k+1)}. \tag{8}$$

3. for a given number k of variables, every set of k constituent variables of the itemset is equiprobable, given a drawing with replacement. The number of such sets is given by $\binom{m+k-1}{k}$. We obtain

$$p(X|k) = \frac{1}{\binom{m+k-1}{k}}. \tag{9}$$

4. for each constituent variable x, the size I_x of the quantile partition is necessarily greater or equal to 2, and distributed according to the universal prior for integers

$$p(I_x) = 2^{-L(I_x-1)}. \tag{10}$$

5. for each constituent variable x_k, the index i_x of the quantile part is uniformly distributed between 1 and I_x

$$p(I_{i_x|I_x}) = \frac{1}{I_x}. \tag{11}$$

Given the definition of the model space and its prior distribution, we can now express the prior probabilities of an IBCM model.

[3] It does not mean that the hierarchical prior is a uniform prior over the itemset space, which would be equivalent to a maximum likelihood approach.

[4] This universal prior is defined so that the small integers are more probable than the large integers, and the rate of decay is taken to be as small as possible. The code length of the universal prior for integers is given by

$$L(n) = \log_2(c_0) + \log_2^*(n) = \log_2(c_0) + \sum_{j>1} max(\log_2^{(j)}(n), 0),$$

where $\log_2^{(j)}(n)$ is the j^{th} composition of $\log_2$ ($\log_2^{(1)}(n) = \log_2(n)$, $\log_2^{(2)}(n) = \log_2(\log_2(n)), \ldots$) and $c_0 = \sum_{n>1} 2^{-log_2^*(n)} = 2.865\ldots$ The universal prior for integers is then $p(n) = 2^{-L(n)}$.

Construction cost of an IBCM *variable.* We now have an analytical formula for the construction cost $c_c(A_\pi)$ of a secondary variable A_π constructed from an IBCM itemset π:

$$c_c(A_\pi) = \log|\mathcal{T}| + L(k+1)\log 2 + \log\binom{m+k-1}{k} + \sum_{x\in X}(L(I_x-1)\log 2 + \log I_x). \tag{12}$$

The cost of an IBCM variable is the negative logarithm of probabilities which is no other than a coding length according to Shannon [27]. Here, $c_c(A_\pi)$ may be interpreted as a variable construction cost, that is the encoding cost of the itemset π. The first line in Formula 12 stands for the choice of the itemset table, the number of variables and the variables involved in the itemset. The second line is related to the choice of the size of the quantile partition and the quantile part for each variable involved in the itemset.

In Formula 6, the construction cost $c_c(A_\pi)$ acts as a regularization term. Constructed variables based on complex itemsets, with multiple constituent variables or with fine-grained constituent parts in the itemset, are penalized compared to simple constructed variables.

4.4 Variable Construction Algorithm

The objective is to construct a set of itemset based secondary variables in order to obtain a data representation suitable for supervised classification. The space of IBCM variables is so large that exhaustive search is not possible. We propose a greedy Algorithm 1 that constructs all potential variables based on quantile partitions of power of 2 sizes, given a maximum number Max_k of constituent variables in a secondary table T, a maximum size Max_s of partitions and a maximum number Max_c of constructed variables. For clarity purpose, this algorithm is described for one single secondary table T. It just has to be applied in a loop over the tables of $\mathcal{T}$ to construct itemset-based variables for all secondary tables of $\mathcal{T}$.

Filtering Actual Quantile Partitions. Let us first notice that definitions 2 and 3 relate to formal descriptions of quantile partitions. Actual partitions may contain empty parts or fine grained parts that are redundant with coarse grained parts. This is the case when the partition size is greater than the number of values, especially when the number of values is below the number of instances in the dataset. To illustrate this, let us consider a variable with only three values 1, 2 and 3 and a dataset of 10 instances, with the following sorted instances values: $1, 1, 1, 2, 2, 2, 3, 3, 3, 3$. According to Definition 2, the 2-quantile partition is $\{\,]-\infty, 2[,\ [2, +\infty[\,\}$, the 3-quantile partition is $\{\,]-\infty, 2[,\ [2, 3[,\ [3, +\infty[\,\}$ and the 4-quantile partition is $\{\,]-\infty, 1[,\ [1, 2[,\ [2, 3[,\ [3, +\infty[\,\}$. In the 4-quantile partition, the first part $]-\infty, 1[$ is empty while the last two ones are redundant with those of the 3-quantile partition. Overall, we can filter the quantile partitions

Algorithm 1. Greedy construction of itemset based variables

Require: T {Input secondary table}
Require: Max_k {Maximum number of constituent variables}
Require: Max_s {Maximum size of quantile distribution of constituent variables}
Require: Max_c {Maximum number of constructed variables}
Ensure: $\mathcal{A}_\pi = \{A_\pi, c_c(A_\pi) + c_e(A_\pi) < c_e(\emptyset)\}$ {Set of relevant constructed variables}
{**Step 1: Compute quantile partitions for power of 2 sizes**}
Read secondary table T
for $x \in T$ **do**
  for all $s = 2^i, 1 \leq i \leq \log_2 Max_s$ **do**
    Compute quantile partition of size s for variable x {cf. definitions 2 and 3}
  end for
end for

{**Step 2: Construct itemset based variables**}
{Exhaustive construction by increasing number of variables and partition size}
$\mathcal{A}_\pi \leftarrow \emptyset, varNb \leftarrow 0$
for $k = 0$ to $\max(Max_k, m)$ **do**
  $subsetNb \leftarrow \frac{m!}{k!(m-k)!}$ {Number of subset of k variables among m}
  if $varNb + subsetNb * 2^k \leq Max_c$ **then**
    $maxSize_k \leftarrow \arg\max_{s \in \{2,4,8,\ldots\}} (varNb + subsetNb * (2s-2)^k \leq Max_c)$
    $varNb \leftarrow varNb + subsetNb * (2 * maxSize_k - 2)^k$
    for all A_π with k variables and parts in partition of size $2, 4, 8, \ldots, maxSize_k$ **do**
      {Construct only new variables by avoiding missing or redundant parts}
      if A_π contains only non empty and non redundant parts **then**
        $\mathcal{A}_\pi \leftarrow \mathcal{A}_\pi \cup \{A_\pi\}$
      end if
    end for
  end if
end for

{**Step 3: Evaluate constructed variables and keep relevant variables only**}
Read secondary table T
for all $A_\pi \in \mathcal{A}_\pi$ **do**
  Compute values of constructed variable A_π
  Evaluate A_π according to $c_r(A_\pi) = c_c(A_\pi) + c_e(A_\pi)$
  if $c_r(A_\pi) > c_e(\emptyset)$ **then**
    $\mathcal{A}_\pi \leftarrow \mathcal{A}_\pi - \{A_\pi\}$
  end if
end for

by keeping $\{[2,3[,\ [3,+\infty[\}$ for the 3-quantile partition and $\{[1,2[\}$ for the 4-quantile partition.

Now, we can detail the variable construction Algorithm 1, that consists in three steps.

1. In the first step (line 1), Algorithm 1 reads the N_s records of table T and computes all quantile partitions for power of 2 sizes up to Max_s, according to definitions 2 and 3. This step requires sorting the records for each secondary variable (among m), then processing them for partition sizes $2, 4, 8, \ldots, \min(Max_s, N_s)$, that is at most $\log_2 N_s$ times. Empty or redundant parts in actual quantile partitions are removed at this step, and the overall number of parts per variable that need to be stored is less than or equal to the number N_s of records.

   Overall, the first step of the algorithm requires $O(m\, N_s \log N_s)$ time and $O(N_s\, m)$ space.
2. In the second step (line 9), Algorithm 1 iterates on itemsets by increasing number of constituent variables, and for each number of variables, by increasing the size of partitions, considering only power of 2 sizes. For a given number k of constituent variables, the number of subsets of variables is $subsetNb = \frac{m!}{k!(m-k)!}$. For a given maximum size of partition $maxSize = 2^i, i \geq 1$, the total number of usable parts is $2 * maxSize - 2 = 2 + 4 + 8 + \ldots + maxSize$. The total number of potential itemsets with k constituent variables and part from quantile partitions with power of 2 sizes less than or equal to $maxSize$ is $subsetNb * (2 * maxSize)^k$. In line 15, Algorithm 1 computes the maximum size $maxSize_k$ of quantile partitions that can be considered to build all related itemset based variables, while not exceeding the maximum number of requested constructed variables Max_c. In line 19, Algorithm 1 exploits the actual quantile partitions obtained after the first step, so as to filter the itemset based constructed variables. Any constructed variable involving an empty or a redundant part is removed, since the same records will be covered by a simpler itemset, with a lower construction cost. The variable construction step is similar to a breadth first tree search of the space of itemsets, constrained by a maximum size of quantile partitions, of number of variable s and of total constructed variables. Overall, this step requires $O(Max_c\, Max_k)$ time and $O(Max_c\, Max_k)$ space, since at most Max_c variables are constructed, each involving at most Max_k constituent variables in the itemsets.
3. In the third step (line 26), Algorithm 1 reads all the dataset (N instances of the target table and N_s records of the secondary table) to compute the values of all constructed variables, that is new binary values in secondary tables. To evaluate these binary secondary variables, the method described in Section 3 needs two count variables per itemset-based constructed binary variable in the target table: the number of secondary records covered or not by the itemset. The evaluation algorithm (see Section 3) requires $O(N \log N)$ time to evaluate a binary secondary variable A_π. In line 31 of Algorithm 1,

a comparison between the relevance criterion $c_r(A\pi)$ and the null cost $c_e(\emptyset)$ allows to filter the constructed variables and to keep only the relevant ones. Overall, this step requires $O(Max_c\, Max_k\, Ns)$ time to compute the values of the binary itemset based constructed variables and $O(Max_c\, N)$ space to keep the count values in the target table. Evaluating the relevance of all variables requires $O(Max_c\, N \log N)$ time and $O(Max_c\, N)$ space.

Overall, Algorithm 1 needs to read the whole dataset twice, one in the first step to build the actual quantile partitions and one in the third step to compute the values of all constructed variables. The time complexity is $O(m\, N_s \log N_s + Max_c\, (Max_k\, Ns{+}N \log N))$ and the space complexity is $O(N_s\, m{+}Max_c\, N)$. For itemsets involving few constituent variables, it is approximatively super-linear with the number of instances in the target table, of records in the secondary table, of secondary variables and of constructed variables.

5 Evaluation

We evaluate the proposed method by focusing on the following aspects: ability to generate large numbers of variables without combinatorial explosion, resistance to overfitting and contribution for the prediction task.

5.1 Evaluation on 20 Benchmark Datasets

In this first evaluation, we use 20 datasets from the multi-relational data mining community. Since we are interested in secondary variables, we ignore those of the target table. We focus on the itemsets-based variable construction method presented in Section 4.

After the variable construction step, we exploit the extension of the Naive Bayes classifier to secondary variables described in section 3.2. In this article, we use the Selective Naive Bayes (SNB) classifier [28]. It is a variant of the Naive Bayes with variable selection and model averaging, which is robust and efficient in the case of very large numbers of variables.

In order to have a baseline of comparison, we consider the method Relaggs [10], based on the following propositionalisation rules:

- for each secondary numerical variable: Mean, Median, Min, Max, StdDev, Sum
- for each categorical secondary variable: Mode, CountDistinct (number of distinct values) and the number of occurrences per value.
- the number of records in the secondary table.

The classifier used after propositionalisation is also the SNB classifier.

The used multi-relational datasets[5] belong to different domains : image processing domain (datasets Elephant, Fox, Tiger [29], and the Miml dataset [30],

[5] Miml: `http://lamda.nju.edu.cn/data_MIMLimage.ashx`, Fox, Elephant, Tiger, Mutagenesis, Musk1, Musk2: `http://www.uco.es/grupos/kdis/mil/dataset.html`, Diterpenses: `http://cui.unige.ch/~woznica/rel_weka/`, Stulong: `http://euromise.vse.cz/challenge2003`

with target variables Desert, Mountains, Sea, Sunset, Trees), molecular chemistry domain (Diterpenses [31], Musk1, Musk2 [32], and Mutagenesis [33] with three representations), health domain (Stulong[6] [34], with target variables Cholrisk, Htrisk, Kourisk, Obezrisk and Rarisk), game domain (TicTacToe [35], considered as multi-relational with the nine cells of the game in the secondary table). These datasets have a small size, containing from 100 to 2000 individuals. A description of the datasets is provided in Table 1

In all experiments, Algorithm 1 is used with at most five variables per itemset, and quantiles partition of at most 100 parts. By using Algorithm 1, we are able to control the size of the representation by generating 1, 10, 100, 1,000, 10,000 and 100,000 variables per dataset in the training samples of the 10-folds stratified cross-validation process, which leads to almost 20 million constructed variables.

Interpretability. To see an example of how we can interpret an itemset on real world data, let us consider the Stulong dataset [34]. It is medical database composed of two tables in a one-to-many relationship: (i) the target table Entry contains patients, and (ii) the table Control describes results of clinical examinations for each patient. We consider the target variable CHOLRISK (with two values: Normal and Risky) which denotes whether the patient presents a cholesterol risk. The task is to predict the value of this class by considering the secondary variables in table Control. We give here an example of a relevant itemset proposed by our approach: $\pi : HYPCHL \in \{1\}$. π contains only one secondary variable HYPCHL. The corresponding binary constructed variable A_π means whether the control performed by the patient presents or not a hypercholesterolemia. Figure 5 depicts the optimal bivariate data grid related to A_π. It can be seen that we obtain a contrast of the target values (Normal and Risky) in each cell of this grid. For example the top-left cell gives an interpretable rule: if the patient has at most one control with hypercholesterolemia and at least one control without hypercholesterolemia then he is not likely to have a cholesterol risk (i.e. CHOLRISK=normal) in 90% of cases.

Performance evaluation. In a first analysis, we collect the average test accuracy for each number of generated variables. The method Relaggs, which relies on variable construction by applying systematic aggregation rules, cannot control the combinatorial number of generated variables which varies from a dataset to

[6] The study (STULONG) was realized at the 2nd Department of Medicine, 1st Faculty of Medicine of Charles University and Charles University Hospital, U nemocnice 2, Prague 2 (head. Prof. M. Aschermann, MD, SDr, FESC), under the supervision of Prof. F. Boudík, MD, ScD, with collaboration of M. Tomečková, MD, PhD and Ass. Prof. J. Bultas, MD, PhD. The data were transferred to the electronic form by the European Centre of Medical Informatics, Statistics and Epidemiology of Charles University and Academy of Sciences (head. Prof. RNDr. J. Zvárová, DrSc). The data resource is on the web pages `http://euromise.vse.cz/challenge2004`. At present time the data analysis is supported by the grant of the Ministry of Education CR Nr LN 00B 107.

Table 1. Description of the datasets

	Target Table				Secondary table		
Dataset	Name	(# classes)	% Majority class	# records	Name	# variables	# records
Diterpenses	Compound	23	29.80	1503	Spectrum		
Elephant	Elephant-Image	2	50	200	Regions	230	1391
Fox	Fox-Image	2	50	200	Regions	230	1320
Miml_Desert	Desert-Image	2	79.55	2000	Regions	15	18000
Miml_Mountains	Mountains-Image	2	77.10	2000	Regions	15	18000
Miml_Sea	Sea-Image	2	71	2000	Regions	15	18000
Miml_Sunset	Sunset-Image	2	76.75	2000	Regions	15	18000
Miml_Trees	Trees-Image	2	72	2000	Regions	15	18000
TicTacToe	TicTacToe	2	65.34	958	TicTacToeCell	3	8622
Musk1	Molecule	2	51.08	92	Conformations	166	476
Musk2	Molecule	2	61.76	102	Conformations	166	6575
MutagenesisAtoms	Molecule	2	66.48	188	Mutagenesis Atoms	3	1618
MutagenesisBonds	Molecule	2	66.48	188	Mutagenesis Bonds	6	3995
MutagenesisChains	Molecule	2	66.48	188	Mutagenesis Chains	11	5349
Stulong_Cholrisk	Entry	3	72.05	1417	Control	65	10572
Stulong_Htrisk	Entry	3	72.61	1417	Control	65	10572
Stulong_Kourrisk	Entry	3	56.17	1417	Control	65	10572
Stulong_Obezrisk	Entry	3	77.91	1417	Control	65	10572
Stulong_Rarisk	Entry	3	81.72	1417	Control	65	10572
ImagesTiger	Tiger-Images	2	50	200	Regions	230	1220

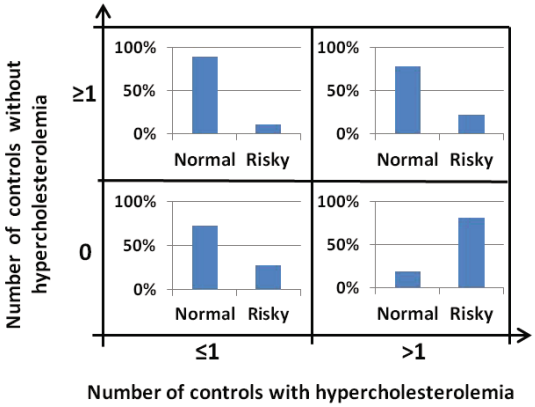

Fig. 5. Example of itemset interpretation on Stulong dataset

another (from about ten variables to 1,400 variables). It is still applicable in the case of small bases and provides here a competitive baseline performance.

The results depicted in Figure 6 show that the performance of our approach systematically increases with the number of constructed variables, and reaches or exceeds the performance of Relaggs over 15 of the 20 datasets when the number of the constructed variables is sufficient. For five datasets (Fox, Musk1, Musk2, MutagenesisAtoms and Tiger), the performance of our approach is significantly worse than Relaggs. These five datasets are either very noisy (Fox with $Acc = 0.63 \pm 0.10$ for Relaggs), very small (Musk1, Musk2 and MutagenesisAtoms with less than 200 instances) or with large variance in the results for Relaggs (from 0.07 to 0.14 of standard deviation in accuracy, eg. 0.9 for Tiger). For these small datasets, there is not sufficient number of instances to reliably recognize a potential pattern. In this case, the regularization (criterion 6) eliminates most of the constructed variables (cf. Figure 7), which brings down the performance. Another explanation that can be provided is that for certain datasets, the pattern in the secondary variables may be easily expressed with an aggregate function. In the case of such a favorable bias, Relaggs is likely to perform better than any other approach.

In order to see the ability of the regularization cost of criterion 6 to eliminate non relevant secondary variables, we report in Figure 7 the number of selected variables with respect to the number of the constructed ones. The results show that criterion 6 significantly prunes the space of the constructed variables. Only a very small number among these variables are considered to be relevant. For example, for 100,000 constructed variables, the proportion of relevant variables is inferior to about 1% for most of the datasets. For some datasets (Elephant, Miml-Sea, TicTacToe, Stulong-Obezrisk and Tiger), only about 10 variables are selected among the 100,000.

Robustness Evaluation. In a second analysis, the experiment is performed after a random reassignment of classes for each dataset in order to assess the robustness

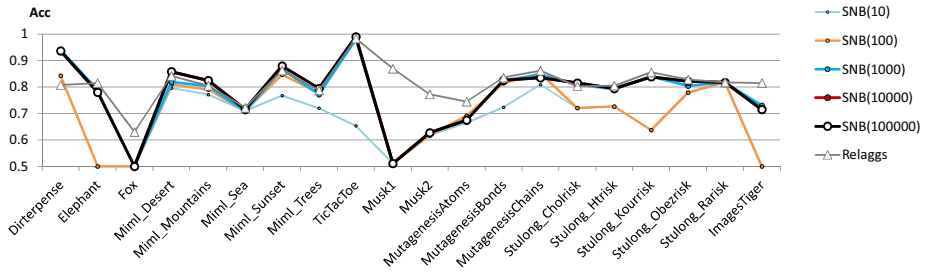

Fig. 6. Test accuracy with respect to the number of constructed variables

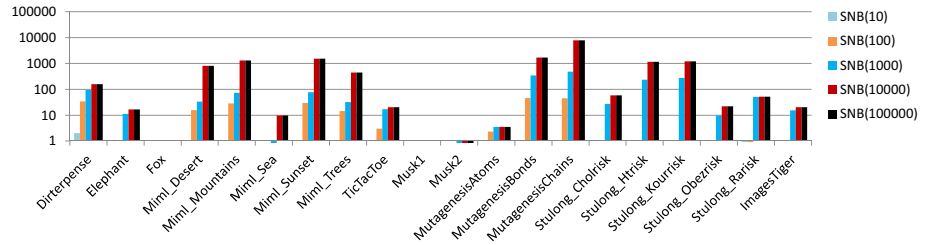

Fig. 7. Number of selected variables with respect to the number of constructed variables

of the approach. We collect the number of selected variables, taking into account the construction cost of the variables according to criterion 6. The method is extremely robust since all 20 million generated variables are identified as non-informative, without exception.

5.2 Handwritten Digits Dataset

In a second evaluation, we use the Digits dataset mentioned earlier in Example 3. In Figure 8, results on the Digits dataset are reported. We used a train and test evaluation with 60,000 instances in training and 10,000 in testing. It is a relatively large dataset with about 50 million records in the secondary table. We report the test accuracy results of the SNB classifier with 1, 10, 100, 1,000, 10,000 and 100,000 constructed variables. These results are compared to those obtained with Relaggs as well as with an SNB on the initial tabular attribute-value representation. We report in the same graphic the number of relevant and filtered variables according to criterion 6 for the different numbers of constructed variables.

The first observation is that the problem is difficult in its initial tabular representation for the SNB classifier which obtains 87.5% of test accuracy. Relaggs only gets 22.4%. In a second observation, Figure 8 shows that the performance of our approach increases with the number of constructed variables, and significantly exceeds that of the SNB classifier obtained on the initial tabular representation. Our rapproach reaches 89.8% with 1,144 relevant variables among

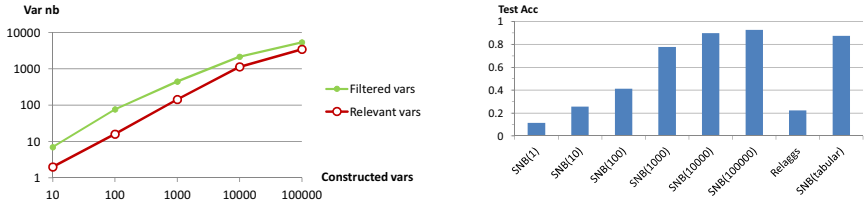

Fig. 8. On the left, number of filtered variables and of relevant variables per number of constructed variables; on the right, test accuracy obtained with Relaggs, SNB with the initial tabular format, and SNB with increasing numbers of constructed variables

10,000 constructed variables, and 92.6% with only 3,476 relevant variables among 100,000 constructed variables.

6 Conclusion

In this paper, we have proposed an approach for constructing new variables and assessing their relevance in the context of multi-relational supervised learning. The method consists in defining an itemset in a secondary table, leading to a new secondary variable that collects whether secondary records are covered or not by the itemset. The relevance of this new variable is evaluated using a bivariate supervised data grid model [11], which provides a regularized estimator of the conditional probability of the target variable. To take into account the risk of overfitting that increases with the number of constructed variables, we have applied a Bayesian model selection approach for both the itemset-based construction model and the conditional density evaluation model, and obtained an exact analytical criterion for the posterior probability of any constructed variable.

A greedy algorithm has been proposed in order to explore the itemset space. We evaluated our approach on several real world multi-relational datasets. Obtained classification performance are very promising. The experiments showed also that our approach is able to deal with relatively large datasets and generate an important number of itemsets while controlling the combinatorial explosion. Furthermore, it is a robust approach. Even with a great number of constructed variables, it remains resistant to overfitting. Future works are envisaged to provide improved search heuristics to better explore the space of constructed variables.

References

1. Knobbe, A.J., Blockeel, H., Siebes, A., Van Der Wallen, D.: Multi-Relational Data Mining. In: Proceedings of Benelearn 1999 (1999)
2. Džeroski, S., Lavrač, N.: Relational Data Mining. Springer-Verlag New York, Inc. (2001)

3. Kramer, S., Flach, P.A., Lavrač, N.: Propositionalization approaches to relational data mining. In: Džeroski, S., Lavrač, N. (eds.) Relational Data Mining, pp. 262–286. Springer, New York (2001)
4. Van Laer, W., De Raedt, L., Džeroski, S.: On multi-class problems and discretization in inductive logic programming. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1997. LNCS, vol. 1325, pp. 277–286. Springer, Heidelberg (1997)
5. Knobbe, A.J., Ho, E.K.Y.: Numbers in multi-relational data mining. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721, pp. 544–551. Springer, Heidelberg (2005)
6. Alfred, R.: Discretization Numerical Data for Relational Data with One-to-Many Relations. Journal of Computer Science 5(7), 519–528 (2009)
7. Lachiche, N., Flach, P.A.: A first-order representation for knowledge discovery and Bayesian classification on relational data. In: PKDD 2000 Workshop on Data Mining, Decision Support, Meta-learning and ILP, pp. 49–60 (2000)
8. Flach, P.A., Lachiche, N.: Naive Bayesian Classification of Structured Data. Machine Learning 57(3), 233–269 (2004)
9. Ceci, M., Appice, A., Malerba, D.: Mr-SBC: A Multi-relational Naïve Bayes Classifier. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 95–106. Springer, Heidelberg (2003)
10. Krogel, M.-A., Wrobel, S.: Transformation-based learning using multirelational aggregation. In: Rouveirol, C., Sebag, M. (eds.) ILP 2001. LNCS (LNAI), vol. 2157, pp. 142–155. Springer, Heidelberg (2001)
11. Lahbib, D., Boullé, M., Laurent, D.: Informative variables selection for multi-relational supervised learning. In: Perner, P. (ed.) MLDM 2011. LNCS, vol. 6871, pp. 75–87. Springer, Heidelberg (2011)
12. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE (11), 2278–2324 (1998)
13. De Raedt, L., Dehaspe, L.: Mining Association Rules in Multiple Relations. In: Džeroski, S., Lavrač, N. (eds.) ILP 1997. LNCS, vol. 1297, pp. 125–132. Springer, Heidelberg (1997)
14. Nijssen, S., Kok, J.N.: Faster association rules for multiple relations. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence, vol. (1) (2001)
15. Guo, J., Bian, W., Li, J.: Multi-relational Association Rule Mining with Guidance of User. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), pp. 704–709 (2007)
16. Gu, Y., Liu, H., He, J., Hu, B., Du, X.: MrCAR: A Multi-relational Classification Algorithm Based on Association Rules. In: 2009 International Conference on Web Information Systems and Mining, pp. 256–260 (2009)
17. Crestana-Jensen, V., Soparkar, N.: Frequent itemset counting across multiple tables. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) PAKDD 2000. LNCS, vol. 1805, pp. 49–61. Springer, Heidelberg (2000)
18. Goethals, B., Le Page, W., Mampaey, M.: Mining interesting sets and rules in relational databases. In: Proceedings of the 2010 ACM Symposium on Applied Computing, p. 997 (2010)
19. Goethals, B., Laurent, D., Le Page, W., Dieng, C.T.: Mining frequent conjunctive queries in relational databases through dependency discovery. Knowledge and Information Systems 33(3), 655–684 (2012)
20. Ceci, M., Appice, A.: Spatial associative classification: propositional vs structural approach. Journal of Intelligent Information Systems 27(3), 191–213 (2006)

21. Ceci, M., Appice, A., Malerba, D.: Emerging pattern based classification in relational data mining. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 283–296. Springer, Heidelberg (2008)
22. Boullé, M.: Optimum simultaneous discretization with data grid models in supervised classification A Bayesian model selection approach. Advances in Data Analysis and Classification 3(1), 39–61 (2009)
23. Gay, D., Boullé, M.: A bayesian approach for classification rule mining in quantitative databases. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 243–259. Springer, Heidelberg (2012)
24. Lahbib, D., Boullé, M., Laurent, D.: An evaluation criterion for itemset based variable construction in multi-relational supervised learning. In: Riguzzi, F., Železný, F. (eds.) The 22nd International Conference on Inductive Logic Programming (ILP 2012), Dubrovnik, Croatia (2012)
25. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, New York (1991)
26. Rissanen, J.: A universal prior for integers and estimation by minimum description length. Annals of Statistics 11(2), 416–431 (1983)
27. Shannon, C.: A mathematical theory of communication. Technical report. Bell Systems Technical Journal (1948)
28. Boullé, M.: Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research 8, 1659–1685 (2007)
29. Andrews, S., Tsochantaridis, I., Hofmann, T.: Support vector machines for multiple-instance learning. In: Advances in Neural Information Processing Systems 15, pp. 561–568. MIT Press (2003)
30. Zhou, Z.H., Zhang, M.L.: Multi-instance multi-label learning with application to scene classification. In: Advances in Neural Information Processing Systems (NIPS 2006), Number i, pp. 1609–1616. MIT Press, Cambridge (2007)
31. Džeroski, S., Schulze-Kremer, S., Heidtke, K.R., Siems, K., Wettschereck, D., Blockeel, H.: Diterpene Structure Elucidation From 13C NMR Spectra with Inductive Logic Programming. Applied Artificial Intelligence 12(5), 363–383 (1998)
32. De Raedt, L.: Attribute-Value Learning Versus Inductive Logic Programming: The Missing Links (Extended Abstract). In: Page, D. (ed.) ILP 1998. LNCS, vol. 1446, pp. 1–8. Springer, Heidelberg (1998)
33. Srinivasan, A., Muggleton, S., King, R., Sternberg, M.: Mutagenesis: ILP experiments in a non-determinate biological domain. In: Proceedings of the 4th International Workshop on ILP, pp. 217–232 (1994)
34. Tomečková, M., Rauch, J., Berka, P.: STULONG - Data from a Longitudinal Study of Atherosclerosis Risk Factors. In: ECML/PKDD 2002 Discovery Challenge Workshop Notes (2002)
35. Asuncion, A., Newman, D.: UCI machine learning repository (2007)