A Bayesian approach for classification rule mining in quantitative databases

Dominique Gay and Marc Boullé

Orange Labs
2, avenue Pierre Marzin, F-22307 Lannion Cedex, France
firstname.name@orange.com

Abstract. We suggest a new framework for classification rule mining in quantitative data sets founded on Bayes theory – without univariate preprocessing of attributes. We introduce a space of rule models and a prior distribution defined on this model space. As a result, we obtain the definition of a parameter-free criterion for classification rules. We show that the new criterion identifies interesting classification rules while being highly resilient to spurious patterns. We develop a new parameter-free algorithm to mine locally optimal classification rules efficiently. The mined rules are directly used as new features in a classification process based on a selective naive Bayes classifier. The resulting classifier demonstrates higher inductive performance than state-of-the-art rule-based classifiers.

1 Introduction

The popularity of association rules [1] is probably due to their simple and interpretable form. That is why they received a lot of attention in the recent decades. E.g., when considering Boolean datasets, an association rule is an expression of the form $\pi : X \rightarrow Y$ where the body X and the consequent Y are subsets of Boolean attributes. It can be interpreted as: "*when attributes of X are observed, then attributes of Y are often observed*". The strength of a rule pattern lies in its inductive inference power: from now on, if we observe the attributes of X then we may rely on observing attributes of Y. When Y is a class attribute, we talk about classification rules (like $X \rightarrow c$) which seems to be helpful for classification tasks – indeed, "if an object is described by attributes of X then it probably belongs to class c". A lot of efforts have been devoted to this area in the past years and have given rise to several rule-based classifiers (see pioneering work: "Classification Based on Associations" (CBA [22]). Nowadays, there exist numerous CBA-like classifiers which process may be roughly summarized in two steps: *(i)* mining a rule set w.r.t. an interestingness measure, *(ii)* building a classifier with a selected subset of the mined rules (see [8] for a recent well-structured survey). Another research stream exploits a rule induction scheme: each rule is greedily extended using various heuristics (like e.g. information gain) and the rule set is built using a sequential database covering strategy. Following this framework, several rule-induction-based classification algorithms have been

proposed; e.g. [10, 24, 32]. Rule mining and rule-based classification are still ongoing research topics. To motivate our approach, we highlight some weaknesses of existing methods.

Motivation. Real-world data sets are made of quantitative attributes (i.e. numerical and/or categorical). Usually, each numerical attribute is discretized using a supervised univariate method and each resulting interval is then mapped to a Boolean attribute (see section 5 for further related work about mining quantitative data sets). The simple XOR example (see figure 1) shows the limit of such preprocessing. Indeed, it seems there is no valuable univariate discretization for attribute X (resp. Y), thus both attributes might be pruned during preprocessing. If X and Y are individually non-informative, their combination could be class-discriminant: e.g., the rule $(X < x_b) \wedge (Y < y_b) \rightarrow c_2$ is clearly an interesting pattern. Notice that univariate preprocessing of a categorical attribute, like supervised value grouping, is subject to the same drawback.

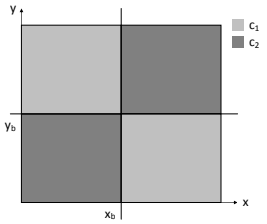

Fig. 1. 2-class XOR data

Another weakness of CBA-like classifiers is parameter tuning. Most of the methods works with parameters: e.g., an interestingness measure threshold for the rules to be mined (sometimes coupled with a frequency threshold), the number of mined rules to use for building the final rule set for classification, etc. The performance of CBA-like classifiers strongly depends on parameter tuning. The choice of parameters is thus crucial but not trivial – each data set may require its own parameter settings. If tuning one parameter could be difficult, a common end-user could be quickly drowned into the tuning of several parameters.

These drawbacks suggest *(i)*, processing quantitative and categorical attributes directly (on the fly) in the mining process in order to catch multivariate correlations that are unreachable with univariate preprocessing and *(ii)* designing an interestingness measure with no need for any wise threshold tuning and a parameter-free method.

Contributions & organization. In this paper, we suggest a new quantitative classification rule mining framework founded on a Bayesian approach. Our method draws its inspiration from the MODL approach (Minimum Optimized Description Length [6]) whose main concepts are recalled in the next section. In section 2, we instantiate the generic MODL approach for the case of classification rules; then, step by step, we build a Bayesian criterion which plays the role of an interestingness measure (with no need for thresholding) for classification rules and we discuss some of its fair properties. In section 3, we also suggest a new efficient parameter-free mining algorithm for the extraction of locally optimal classification rules. A classifier is then built following a simple and intuitive feature construction process based on the mined rules. The resulting classifier shows competitive results when compared with state-of-the-art rule-based classifiers on both real-world and large-scale challenge data sets – showing the added-value of the method (see section 4). Further related work is discussed in section 5 before concluding.

2 Towards MODL rules

MODL principle. The MODL approach is based on a Bayesian approach. Let us consider the univariate supervised discretization task as an example. From the MODL point of view, the problem of discretizing a numerical attribute is formulated as a model selection problem.

Firstly, a space $\mathcal{M}$ of discretization models M is defined. In order to choose the "best" discretization model M , a Bayesian "Maximum A Posteriori" approach (MAP) is used: the probability $p(M|D)$ is to be maximized over $\mathcal{M}$ (i.e., the posterior probability of a given discretization model M given the data D). Using Bayes rule and considering that $p(D)$ is constant in the current optimization problem, it consists in maximizing the expression $p(M) \times p(D|M)$. The prior $p(M)$ and the conditional probability $p(D|M)$ called the likelihood are both computed with the parameters of a specific discretization which is uniquely identified by the number of intervals, the bound of the intervals and the class frequencies in each interval. Notice that the prior exploits the hierarchy of parameters and is uniform at each stage of the hierarchy. The evaluation criterion is based on the negative logarithm of $p(M \mid D)$ and is called the cost of the model M: $c(M) = -\log(p(M) \times p(D \mid M))$. The optimal model M is then the one with the least cost c (see original work [6] for explicit expression of $p(M)$ and $p(D|M)$ and for the optimization algorithm). The generic MODL approach has also already been successfully applied to supervised value grouping [5] and decision tree construction [28]. In each instantiation, the MODL method promotes a trade-off between (1) the fineness of the predictive information provided by the model and (2) the robustness in order to obtain a good generalization of the model. Next, MODL approach is instantiated for the case of classification rules.

Basic notations and definitions. Let $r = \{\mathcal{T}, \mathcal{I}, \mathcal{C}\}$ be a labeled transactional data set, where $\mathcal{T} = \{t_1, \ldots, t_N\}$ is the set of objects, $\mathcal{I} = \{x_1, \ldots, x_m\}$ is a set of attributes (numerical or categorical) and $dom(x_j)$ the domain of an attribute x_j $(1 \leq j \leq m)$ and $\mathcal{C} = \{c_1, \ldots, c_J\}$ is the set of J mutually exclusive classes of a class attribute y. An object t is a vector $t = \langle v_1, \ldots, v_m, c\rangle$ where $v_j \in dom(x_j)$ $(1 \leq j \leq m)$ and $c \in \mathcal{C}$. An item for a numerical attribute x is an interval of the form $x[l_x, u_x]$ where $l_x, u_x \in dom(x)$ and $l_x \leq u_x$. We say that an object $t \in \mathcal{T}$ satisfies an interval item $x[l_x, u_x]$ when $l_x \leq t(x) \leq u_x$. For a categorical attribute, an item is a value group of the form $x\{v_x^1, \ldots, v_x^s\}$ where $v_x^j \in dom(x)$ $(1 \leq j \leq s)$. We say that an object $t \in \mathcal{T}$ satisfies a value group item $x\{v_x^1, \ldots, v_x^s\}$ when $t(x) \in \{v_x^1, \ldots, v_x^s\}$. An itemset X is just a set of items and an object t supports X if t satisfies all items of X. A classification rule π on r is an expression of the form $\pi : X \to c$ where c is a class value and X is an itemset. Notice that a categorical attribute involved in the rule body is partitioned into two value groups: the body item (or group) and the outside item; whereas a numerical attribute, due to the intrinsic order of its values, is discretized into either two or three intervals: the body item (or interval) and the outside item(s) (see example below).

Let us recall that, from a MODL point of view, the problem of mining a classification rule π is formulated as a model selection problem. To choose the

best rule from the rule space we use a Bayesian approach: we look for maximizing $p(\pi|r)$. As explained in previous section, it consists in minimizing the cost of the rule defined as:

$$c(\pi) = -\log(p(\pi) \times p(r \mid \pi))$$

In order to compute the prior $p(\pi)$, we suggest another definition of classification rule based on hierarchy of parameters that uniquely identifies a given rule:

Definition 1 (Standard Classification Rule Model). *A* MODL *rule π, also called* standard classification rule model *(*SCRM*), is defined by:*

- *the constituent attributes of the rule body*
- *the group involved in the rule body, for each categorical attribute of the rule body*
- *the interval involved in the rule body, for each numerical attribute of the rule body*
- *the class distribution inside and outside of the body*

Notice that SCRM definition slightly differs from classical classification rule. The last key point meets the concept of distribution rule [17]. The consequent of a SCRM is an empirical distribution over the classes as illustrated in the following example:

Example of a SCRM. Let us consider the SCRM $\pi : (x_1 \in \{v^1_{x_1}, v^3_{x_1}, v^4_{x_1}\}) \wedge (1.2 < x_2 \leq 3.1) \wedge (x_4 \geq 100) \rightarrow (p_{c_1} = 0.9, p_{c_2} = 0.1)$ where x_1 is a categorical attribute and x_2, x_4 are numerical attributes. The value group $\{v^1_{x_1}, v^3_{x_1}, v^4_{x_1}\}$ and the intervals $]1.2; 3.1]$ and $[100; +\infty[$ are those items involved *in* the rule body. The complementary part (i.e. the negation of their conjunction) constitutes the *outside* part of the rule body. $(p_{c_1} = 0.9, p_{c_2} = 0.1)$ is the empirical class distribution for the objects covered by the rule body (inside part) and the class distribution for the outside part of the body may be deduced easily.

Our working model space is thus the space of all SCRM rules. To apply the Bayesian approach, we first need to define a prior distribution on the SCRM space; and we will need the following notations.

Notations. Let r be a data set with N objects, m attributes (categorical or numerical) and J classes. For a SCRM, $\pi : X \rightarrow (P_{c_1}, P_{c_2}, \ldots, P_{c_J})$ such that $|X| = k \leq m$, we will use the following notations:

– $X = \{x_1, \ldots, x_k\}$: the set of k constituent attributes of the rule body ($k \leq m$)
– $X_{cat} \cup X_{num} = X$: the sets of categorical and numerical attributes of the rule body
– $V_x = |dom(x)|$: the number of values of a categorical attribute x
– I_x: the number of intervals (resp. groups) of a numerical (resp. categorical) attribute x
– $\{i(v_x)\}_{v_x \in dom(x)}$: the indexes of groups to which v_x are affected (one index per value, either 1 or 2 for inside or outside of the rule body)
– $\{N_{i(x).}\}_{1 \leq i \leq I_x}$: the number of objects in interval i of numerical attribute x
– $i_{x_1}, \ldots, i_{x_k}$: the indexes of groups of categorical attributes (or intervals of numerical attributes) involved in the rule body
– $N_X = N_{i_{x_1} \ldots i_{x_k}}$: the number of objects in the body $i_{x_1} \ldots i_{x_k}$
– $N_{\neg X} = N_{\neg i_{x_1} \ldots i_{x_k}}$: the number of objects outside of the body $i_{x_1} \ldots i_{x_k}$
– $N_{Xj} = N_{i_{x_1} \ldots i_{x_k} j}$: the number of objects of class j in the body $i_{x_1} \ldots i_{x_k}$
– $N_{\neg Xj} = N_{\neg i_{x_1} \ldots i_{x_k} j}$: the number of objects of class j outside of the body $i_{x_1} \ldots i_{x_k}$

MODL hierarchical prior. We use the following distribution prior on SCRM models, called the MODL hierarchical prior. Notice that a uniform distribution is used at each stage[1] of the parameters hierarchy of the SCRM models:

- *(i)* the number of attributes in the rule body is uniformly distributed between 0 and m
- *(ii)* for a given number k of attributes, every set of k constituent attributes of the rule body is equiprobable
- *(iii)* for a given categorical attribute in the body, the number of groups is necessarily 2
- *(iv)* for a given numerical attribute in the body, the number of intervals is either 2 or 3 (with equiprobability)
- *(v)* for a given categorical (or numerical) attribute, for a given number of groups (or intervals), every partition of the attribute into groups (or intervals) is equiprobable
- *(vi)* for a given categorical attribute, for a value group of this attribute, belonging to the body or not are equiprobable
- *(vii)* for a given numerical attribute with 2 intervals, for an interval of this attribute, belonging to the body or not are equiprobable. When there are 3 intervals, the body interval is necessarily the middle one.
- *(viii)* every distribution of the class values is equiprobable, in and outside of the body
- *(ix)* the distributions of class values in and outside of the body are independent

Thanks to the definition of the model space and its prior distribution, we can now express the prior probabilities of the model and the probability of the data given the model (i.e., $p(\pi)$ and $p(r \mid \pi)$).

Prior probability. The prior probability $p(\pi)$ of the rule model is:

$$\begin{aligned} p(\pi) = & \, p(X) \\ & \times \prod_{x \in X_{cat}} p(I_x) \times p(\{i(v_x)\}|I_x) \times p(i_x|\{i(v_x)\}, I_x) \\ & \times \prod_{x \in X_{num}} p(I_x) \times p(\{N_{i(x).}\}|I_x) \times p(i_x|\{N_{i(x).}\}, I_x) \\ & \times p(\{N_{Xj}\}\{N_{\neg Xj}\} \mid N_X, N_{\neg X}) \end{aligned}$$

Firstly, we consider $p(X)$ (the probability of having the attributes of X in the rule body). The first hypothesis of the hierarchical prior is the uniform distribution of the number of constituent attributes between 0 and m. Furthermore, the second hypothesis says that every set of k constituent attributes of the rule body is equiprobable. The number of combinations $\binom{m}{k}$ could be a natural way to compute this prior; however, it is symmetric. Beyond $m/2$, adding new attributes makes the selection more probable. Thus, adding irrelevant variables is favored, provided that this has an insignificant impact on the likelihood of the model. As we prefer simpler models, we suggest to use the number of combinations with replacement $\binom{m+k-1}{k}$. Using the two first hypothesis, we have:

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

$$p(X) = \frac{1}{m+1} \cdot \frac{1}{\binom{m+k-1}{k}}$$

For each categorical attribute x, the number of partitions of V_x values into 2 groups is $\mathcal{S}(V_x, 2)$ (where $\mathcal{S}$ stands for Stirling number of the second kind). Considering hypotheses *(iii)*, *(v)*, *(vi)*, we have:

$$p(I_x) = 1 \quad ; \quad p(\{i(v_x)\}|I_x) = \frac{1}{\mathcal{S}(V_x, 2)} \quad ; \quad p(i_x|\{i(v_x)\}, I_x) = \frac{1}{2}$$

For each numerical attribute x, the number of intervals is either 2 or 3. Computing the number of partitions of the (ranked) values into intervals turns into a combinatorial problem. Notice that, when $I_x = 3$ the interval involved in the rule body is necessarily the second one; when $I_x = 2$, it is either the first or the second with equiprobability. Considering hypotheses *(iv)*, *(v)*, *(vii)*, we get:

$$p(I_x) = \frac{1}{2} \quad ; \quad p(\{N_{i.}\}|I_x) = \frac{1}{\binom{N-1}{I_x-1}} \quad ; \quad p(i_x|\{N_{i.}\}, I_x) = \frac{1}{1 + \mathbb{1}_{\{2\}}(I_x)}$$

where $\mathbb{1}_{\{2\}}$ is the indicator function of set $\{2\}$ such that $\mathbb{1}_{\{2\}}(a) = 1$ if $a = 2$, 0 otherwise.

Using the hypotheses *(viii)* and *(ix)*, computing the probabilities of distributions of the J classes inside and outside of the rule body turns into a multinomial problem. Therefore, we have:

$$p(\{N_{Xj}\} \mid N_X, N_{\neg X}) = \frac{1}{\binom{N_X+J-1}{J-1}} \quad ; \quad p(\{N_{\neg Xj}\} \mid N_X, N_{\neg X}) = \frac{1}{\binom{N_{\neg X}+J-1}{J-1}}$$

The likelihood. Now, focusing on the likelihood term $p(r \mid \pi)$, the probability of the data given the rule model is the probability of observing the data inside and outside of the rule body (w.r.t. to N_X and $N_{\neg X}$ objects) given the multinomial distribution defined for N_X and $N_{\neg X}$. Thus, we have:

$$p(r \mid \pi) = \frac{1}{\frac{N_X!}{\prod_{j=1}^{J} N_{Xj}!}} \cdot \frac{1}{\frac{N_{\neg X}!}{\prod_{j=1}^{J} N_{\neg Xj}!}}$$

Cost of a SCRM. We now have a complete and exact definition of the cost of a SCRM π:

$$c(\pi) = \log(m+1) + \log\binom{m+k-1}{k} \tag{1}$$

$$+ \sum_{x \in X_{cat}} \log \mathcal{S}(V_x, 2) + \log 2 \tag{2}$$

$$+ \sum_{x \in X_{num}} \log 2 + \log\binom{N-1}{I_x-1} + \log(1 + \mathbb{1}_{\{2\}}(I_x)) \tag{3}$$

$$+ \log\binom{N_X+J-1}{J-1} + \log\binom{N_{\neg X}+J-1}{J-1} \tag{4}$$

$$+ \left(\log N_X! - \sum_{j=1}^{J} \log N_{Xj}!\right) + \left(\log N_{\neg X}! - \sum_{j=1}^{J} \log N_{\neg Xj}!\right) \tag{5}$$

The cost of a SCRM is the negative logarithm of probabilities which is no other than a coding length according to Shannon [25]. Here, $c(\pi)$ may be interpreted as the ability of a SCRM π to encode the classes given the attributes. Line (1) stands for the choice of the number of attributes and the attributes involved in the rule body. Line (2) is related to the choice of the groups and the values involved in the rule body for categorical attributes; line (3) is for the choice of the number of intervals, their bounds and the one involved in the rule body for numerical attributes. Line (4) corresponds to the class distribution in and outside of the rule body. Finally, line (5) stands for the likelihood.

Since the magnitude of the cost depends on the size of the data set (N and m), we defined a normalized criterion, called *level* and which plays the role of interestingness measure to compare two SCRM.

Definition 2 (Level: interestingness of SCRM). *The level of a* SCRM *is:*

$$level(\pi) = 1 - \frac{c(\pi)}{c(\pi_\emptyset)}$$

where $c(\pi_\emptyset)$ is the cost of the null model (i.e. default rule with empty body). Intuitively, $c(\pi_\emptyset)$ is the coding length of the classes when no predictive information is used from the attributes. The cost of the default rule $\pi_\emptyset$ is formally:

$$c(\pi_\emptyset) = \log(m+1) + \log \binom{N+J-1}{J-1} + \left(\log N! - \sum_{j=1}^{J} \log N_j! \right)$$

The *level* naturally draws the frontier between the interesting patterns and the irrelevant ones. Indeed, rules π such that $level(\pi) \leq 0$, are not more probable than the default rule $\pi_\emptyset$; then using them to explain the data is more costly than using $\pi_\emptyset$ – such rules are considered irrelevant. Rules such that $0 < level(\pi) \leq 1$ highlight the interesting patterns π. Indeed, rules with lowest cost (highest *level*) are the most probable and show correlations between the rule body and the class attribute. In terms of coding length, the *level* may also be interpreted as a compression rate. Notice also that $c(\pi)$ is smaller for lower k values (cf. line (1)), i.e. rules with shorter bodies are more probable thus preferable – which meets the consensus: "Simpler models are more probable and preferable". This idea is translated in the following proposition (the proof is almost direct):

Proposition 1. *Given two* SCRM *π and π' resp. with bodies X and X', such that $X \subseteq X'$ and sharing the same contingency table (i.e., $N_X = N'_X$, $N_{\neg X} = N_{\neg X'}$, $N_{Xj} = N_{X'j}$, $N_{\neg Xj} = N_{\neg X'j}$), then we have: $c(\pi) < c(\pi')$ and π is preferable.*

Asymptotic behavior. The predominant term of the cost function is the likelihood term (eq.(5)) that indicates how accurate the model is. The others terms behave as regularization terms, penalizing complex models (e.g., with too many attributes involved in the rule body) and preventing from over-fitting. The following two theorems show that the regularization terms are negligible when the number of objects N of the problem is very high and that the

cost function is linked with Shannon class-entropy [11] (due to space limitations, full proofs are not given, but the key relies on the Stirling approximation: $\log n! = n(\log n - 1) + O(\log n)$).

Theorem 1. *The cost of the default rule $\pi_\emptyset$ for a data set made of N objects is asymptotically N times the Shannon class-entropy of the whole data set when $N \to \infty$, i.e. $H(y) = -\sum_{j=1}^{J} p(c_j) \log p(c_j)$.*

$$\lim_{N\to\infty} \frac{c(\pi_\emptyset)}{N} = -\sum_{j=1}^{J} \frac{N_j}{N} \log \frac{N_j}{N}$$

Theorem 2. *The cost of a rule π for a data set made of N objects is asymptotically N times the Shannon conditional class-entropy when $N \to \infty$, i.e. $H(y|x) = -\sum_{x\in\{X,\neg X\}} p(x) \sum_{j=1}^{J} p(c_j|x) \log p(c_j|x)$.*

$$\lim_{N\to\infty} \frac{c(\pi)}{N} = \frac{N_X}{N}\left(\sum_{j=1}^{J} -\frac{N_{Xj}}{N_X} \log \frac{N_{Xj}}{N_X}\right) + \frac{N_{\neg X}}{N}\left(\sum_{j=1}^{J} -\frac{N_{\neg Xj}}{N_{\neg X}} \log \frac{N_{\neg Xj}}{N_{\neg X}}\right)$$

The asymptotic equivalence between the coding length of the default rule $\pi_\emptyset$ and the class-entropy of the data confirms that "rules such that $level \leq 0$ identify patterns that are not statistically significant" and links the MODL approach with the notion of incompressibility of Kolmogorov [21] – which defines randomness as the impossibility of compressing the data shorter than its raw form.
The asymptotic behavior of the cost function (for a given rule π) confirms that high *level* values highlight the most probable rules that characterize classes, since high *level* value means high class-entropy ratio between π and the default rule. In terms of compression, rules with $level > 0$ correspond to a coding with better compression rate than the default rule; thus, they identify patterns that do not arise from randomness. Here, we meet the adversarial notions of spurious and significant patterns as mentioned and studied in [31]. Conjecture 1 illustrates this idea and we bring some empirical proof to support it in Section 4:

Conjecture 1. Given a classification problem, for a random distribution of the class values, there exist no SCRM with $level > 0$ (asymptotically according to N, almost surely).

Problem formulation. Given the MODL method framework instantiated for classification rules, an ambitious problem formulation would have been: "*Mining the whole set of* SCRM *with level* > 0" (or the set of K-top *level* SCRM). However, the model space is huge, considering all possibilities of combinations of attributes, attribute discretization and value grouping: the complexity of the problem is $O((2^{V_c})^{m_c}(N^2)^{m_n})$ where m_c is the number of categorical attributes with V_c values and m_n the number of numerical attributes. Contrary to some standard approaches for classification rule mining, exhaustive extraction is not an option. Our objective is to sample the posterior distribution of rules using a randomization strategy, starting from rules (randomly) initialized according to

their prior. Therefore, we opt for a simpler formulation of the problem: "*Mining a set of locally optimal* SCRM *with level* > 0". In the following, we describe our mining algorithm and its sub-routines for answering the problem.

3 MODL rule mining

This section describes our strategy and algorithm for mining a set of locally optimal SCRM (see algorithm 1) and how we use it in a classification system.

Algorithm 1: MACATIA: The MODL-rule miner

```
   Input  : r = {T, I, C} a labeled data set
   Output: Γ a set of locally optimal SCRM
 1 Γ ← ∅;
 2 while ¬ StoppingCondition do
 3 |   t ←chooseRandomObject(T);
 4 |   X ←chooseRandomAttributes(I);
 5 |   π ←initRandomBodyRule(X,t);
 6 |   currentLevel ← computeRuleLevel(π,r);
 7 |   repeat
 8 |   |   minLevel ← currentLevel;
 9 |   |   randomizeOrder(X);
10 |   |   for x ∈ X do
11 |   |   |   OptimizeAttribute(t, x, X);
12 |   |   deleteNonInformativeAttributes(X);
13 |   |   currentLevel ← computeRuleLevel(π,r);
14 |   until noMoreLevelImprovement;
15 |   if currentLevel > 0 then
16 |   |   Γ ← Γ ∪ {π};
17 return Γ
```

The MODL rule miner. We adopt an instance-based randomized strategy for mining rules in given allowed time. The stopping condition (l.2) is the time that the end-user grants to the mining process. At each iteration of the main loop (l.2-16), a locally optimal SCRM is built – when time is up, the process ends and the current rule set is output. Firstly (l.3-5), a random object t and a random set of k attributes are chosen from the data set; then, a SCRM π is randomly initialized such that the body of π is made of a random itemset based on attributes X and t supports the rule body (to simplify notations, X and the body itemset based on X own the same notation). The inner loop (l.7-14) optimizes the current rule while preserving the constraint "t supports body itemset X". We are looking for *level* improvement: a loop of optimization consists in randomizing the order of the body attributes, optimizing each item (attribute) sequentially – the intervals or groups of an attribute are optimized while the other body attributes are fixed (see specific instantiations of OptimizeAttribute in sub-routines algorithms 2 and 3), then removing non-informative attributes from the rule body (i.e., attributes with only one interval or only one value group). Optimization phase ends when there is no more improvement. Finally, the optimized rule is added to the rule set if its *level* is positive.

Attribute optimization. Let us remind that, while optimizing a rule, each rule attribute (item) is optimized sequentially while the others are fixed.

For a numerical attribute x (see algorithm 2), we are looking for the best bounds

for its body interval containing $t(x)$ (i.e. the bounds that provide the best *level* value for the current SCRM while other attributes are fixed). If there are two intervals (l.3-4), only one bound is to be set and the best one is chosen among all the possible ones. When there are three intervals (l.1-2), the lower bound and the upper bound of the body interval are to be set. Each bound is set sequentially and in random order (again, the best one is chosen while the other is fixed). Since an interval might be empty at the end of this procedure, we remove empty intervals (l.5) – the current attribute might be deleted from the rule body by the main algorithm if only one interval is remaining.

Algorithm 2: OPTIMIZEATTRIBUTE: Numerical attribute optimization

Input : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{C}\}$ a transactional labeled data set, $\pi : X \to (p_{c_J}, \ldots, p_{c_J})$ a SCRM covering an object $t \in \mathcal{T}$, $x \in X$ a numerical attribute of the rule body
Output: x an optimized numerical attribute

```
1  if x.IntervalsNumber == 3 then
2      (x.LB,x.UB) ← CHOOSEBESTBOUNDS(t, x, π, r);
3  if x.IntervalsNumber == 2 then
4      x.B ← CHOOSEBESTBOUND(t, x, π, r);
5  CLEANEMPTYINTERVALS();
6  return x
```

Algorithm 3: OPTIMIZEATTRIBUTE: Categorical attribute optimization

Input : $r = \{\mathcal{T}, \mathcal{I}, \mathcal{C}\}$ a transactional labeled data set, $\pi : X \to (p_{c_J}, \ldots, p_{c_J})$ a SCRM covering an object $t \in \mathcal{T}$, $x \in X$ a categorical attribute of the rule body
Output: x an optimized categorical attribute

```
1  minLevel ← COMPUTERULELEVEL(π,r);
2  currentLevel ← COMPUTERULELEVEL(π,r);
3  SHUFFLE(x.allValues);
4  for value ∈ {x.allValues \{t(x)}} do
5      CHANGEGROUP(value);
6      currentLevel ← COMPUTERULELEVEL(π,r);
7      if currentLevel > minLevel then
8          minLevel ← currentLevel;
9      else
10         CHANGEGROUP(value);
11 CLEANEMPTYGROUPS();
12 return x
```

For a categorical attribute x (see algorithm 3), we are looking for a partition of the value set into two value groups (i.e. the value groups that provide the best *level* value for the current SCRM while other attributes are fixed). First (l.3), the values of the current categorical attribute are shuffled. Then (l.5-10), we try to transfer each value (except for $t(x)$ staying in the body group) from its origin group to the other: the transfer is performed if the *level* is improved. Once again we clean possible empty value group at the end of the procedure (necessarily the out-body group) – the current attribute might be deleted from the rule body by the main algorithm if only one group is remaining.

About local optimality. Our MODL rule miner (and its sub-routines) bet on a trade-off between optimality and efficiency. In the main algorithm, the strategy of optimizing an attribute while the other are fixed leads us to a local optimum (so do the strategies of optimizing interval and value group items). This trade-off allows us to mine *one* rule in time complexity $O(kN \log N)$ using efficient implementation structures and algorithms. Due to space limitation, we cannot

give details about the implementation.

About mining with diversity. Randomization is present at each stage of our algorithm. Notice that the randomization is processed according the defined and motivated hierarchical prior (except for object choice). As said above, we are not looking for an exhaustive extraction but we want to sample the posterior distribution of SCRM rules. This randomization facet of our method (plus the optimization phase) allows us mine interesting rules ($level > 0$) with diversity.

Classification system. We adopt a simple and intuitive feature construction process to build a classification system based on a set of locally optimal SCRM. For each mined rule π, a new Boolean attribute (feature) is created: the value of this new feature for a training object t of the data set r is (1) true if t supports the body of π, (0) false otherwise. This feature construction process is certainly the most straightforward but has also shown good predictive performance in several studies [9]. To provide predictions for new incoming (test) objects, we use a Selective Naive Bayes classifier (SNB) on the recoded data set. This choice is motivated by the good performances of SNB on benchmark data sets as well as on large-scale challenge data sets (see [7]). Moreover, SNB is Bayesian-based and parameter-free, agreeing with the characteristics of our method.

4 Experimental validation

In this section, we present our empirical evaluation of the classification system (noted KRSNB). Our classification system has been developed in C++ and is using a JAVA-based user interface and existing libraries from KHIOPS [2]. The experiments are performed to discuss the following questions:

- $\mathcal{Q}_1$ The main algorithm is controlled by a running time constraint. In a given allowed time, a certain number of rules might be mined: how do the performance of the classifier system evolve w.r.t. the number of rules? And, what about the time-efficiency of the process?
- $\mathcal{Q}_2$ Does our method suffer over-fitting? What about spurious patterns? We will also bring an empirical validation of conjecture 1.
- $\mathcal{Q}_3$ Does the new feature space (made of locally optimal rules) improve the predictive performance of SNB?
- $\mathcal{Q}_4$ Are the performance of the classification system comparable with state-of-the-art `CBA`-like classifiers?

For our experiments, we use 29 UCI data sets commonly used in the literature (australian, breast, crx, german, glass, heart, hepatitis, horsecolic, hypothyroid ionosphere, iris, LED, LED17, letter, mushroom, pendigits, pima, satimage, segmentation, sickeuthyroid, sonar, spam, thyroid, tictactoe, vehicle, waveform and its noisy version, wine and yeast) and which show a wide variety in number of objects, attributes and classes, in the type of attributes and the class distribution (see [2] for a full description). All performance results reported in the following are obtained with stratified 10-fold cross validation. Notice that, the feature construction step is performed *only on the training set* and the new learned features are reported on the test set for each fold of the validation.

[2] `http://www.khiops.com`

Evolution of performance w.r.t. the number of rules. In figure 2, we plot the performance in terms of accuracy and AUC of KRSNB based on ρ rules ($\rho = 2^n, n \in [0, 10]$). The details per data set is not as important as the general behavior: we see that, generally, the predictive performance (accuracy and AUC) increases with the number of rules. Then, the performance reaches a plateau for most of the data sets: with about a few hundreds of rules for accuracy and a few rules for AUC.

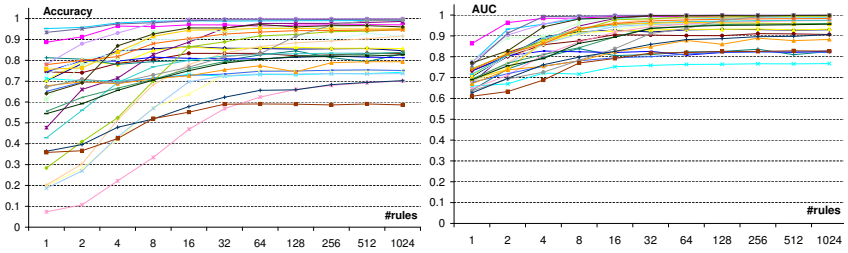

Fig. 2. Accuracy and AUC results (per data set) w.r.t. number of rules mined.

Running time report. Due to our mining strategy, running time grows linearly with the number of rules to be mined. For most of the data sets, mining a thousand rules is managed in less than 500s. In fig. 3, for each of the 29 data sets, we report the processing time of KRSNB based on 1024 rules w.r.t. the size of the data set – plotted with logarithmic scales. It appears that the MODL-rule miner roughly runs in linear time according to $N \times m$. The analysis of performance evolution and running time w.r.t. the number of rules shows that KRSNB reaches its top performance in reasonable time using a few hundreds of rules. In the following, to facilitate the presentation, we will experiment our classifier with 512 rules.

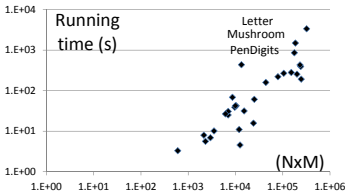

Fig. 3. Running time for mining 1024 rules w.r.t. the size of the data set ($N \times m$).

About spurious patterns and robustness of our method. As mentioned in [31], "*Empirical studies demonstrate that standard pattern discovery techniques can discover numerous spurious patterns when applied to random data and when applied to real-world data result in large numbers of patterns that are rejected when subjected to sound statistical validation*". Proposition 1 states that in a data set with random class distribution, there should not exist any SCRM with $level > 0$ (i.e. no interesting rule). To support this proposition, we lead the following empirical study: *(i)* for each of the 29 benchmark data sets, we randomly assign a class label $c \in \mathcal{C}$ to the objects; *(ii)* we run KRSNB on the data sets with random labels. The result is strong: all rule optimizations during the process end with a default rule with $level \leq 0$. This study shows that our method is robust, discovers no spurious patterns and thus avoids over-fitting.

Added-value of the new feature space. We process a comparative study of the performance of SNB and KRSNB to demonstrate the added-value of the new feature space. Due to space limitations, we skip results on individual data sets and only Area Under ROC Curve (AUC) and accuracy average results of each method are reported in table 1. We are aware of the problems arising from averaging results over various data sets, therefore Win-Tie-Loss (WTL) and average rank results are also reported. A raw analysis of the results gives advantage to KRSNB (in all dimensions: average accuracy and AUC, rank and WTL results). Concerning the Win-Tie-Loss results (WTL) at significance level $\alpha = 0.05$, the critical value for the two-tailed sign test is 20 (for 29 data sets). Thus, even if we cannot assert a significant difference of AUC performance between the two approaches, WTL AUC results of KRSNB vs SNB is close to the critical value ($17 < 20$) – which is a promising result. Moreover, the new feature space made of locally optimal SCRM is clearly a plus when considering WTL accuracy results.

	Accuracy		AUC	
Algorithms	avg	rank	avg	rank
SNB	83.58	1.72	92.43	1.60
KRSNB	84.80	1.28	93.48	1.39
WTL KR VS. SNB	**21**/0/8		**17**/2/10	

Table 1. Comparison of SNB and KRSNB performance results.

Algorithms	avg.acc	avg.rank	KR-WTL
KRSNB	84.80	2.17	-
HARMONY	83.31	3.53	19/1/9
KRIMP	83.31	3.64	23/1/5
RIPPER	84.38	2.83	19/1/9
PART	84.19	2.83	18/1/10

Table 2. Comparison of KRSNB with state-of-the-art methods.

Comparisons with state-of-the-art

Let us first notice that, for the tiny XOR case shown in introduction, KRSNB easily finds the four obvious 2-dimensional patterns characterizing the classes – this finding is unreachable for CBA-like methods using univariate pre-processing. We now compare the performance of KRSNB with four state-of-the-art competitive rule-based classifiers: two recent pattern-based classifiers: HARMONY [29] an instance-based classifier and KRIMP [20] a compression-based method; and two induction-rule-based approaches RIPPER [10] and PART [13] available from the WEKA platform [16] with default parameters. The choice of accuracy for performance comparisons is mainly motivated by the fact that competitors (HARMONY and KRIMP) provide only accuracy results. Since HARMONY and KRIMP are restricted to Boolean (or categorical) data sets, we preprocess the data using a MDL-based univariate supervised discretization [16] for these methods. We also run experiments with parameters as indicated in the original papers. Once again, only average results are reported in table 2. A first analysis of the raw results shows that KRSNB is highly competitive. Again, average accuracy, Win-Tie-Loss and average rank results give advantage to KRSNB. We also applied the Friedman test coupled with a post-hoc Nemenyi test as suggested by [12] for multiple comparisons (at significance level $\alpha = 0.05$ for both tests). The null-hypothesis was rejected, which indicates that the compared classifiers are not equivalent in terms of accuracy. The result of the Nemenyi test is represented by the critical difference chart shown in figure 4 with $CD \simeq 1.133$. First of all, we observe that there is no critical difference of performance between

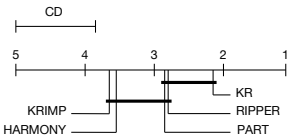

Fig. 4. Critical difference of performance between KRSNB and state-of-the-art rule-based classifiers.

the four competitors of KRSNB. Secondly, even if KRSNB is not statistically singled out, it gets a significant advantage on HARMONY and KRIMP – whereas PART and RIPPER do not get this advantage.

Results on challenge data sets

We also experiment KRSNB on recent large-scale challenge data sets (Neurotech challenges at PAKDD'09-10 and Orange *small* challenge at KDD'09)[3]. Each data set involves 50K instances, from tens to hundreds quantitative attributes, and two highly imbalanced classes – recognized as a difficult task. We experiment KRSNB and its competitors in a 70%train-30%test setting and report AUC results in table 3. As univariate pre-processing of quantitative attributes generate thousands of variables, we were not able to obtain any results with KRIMP and HARMONY. Thus, a first victory for KRSNB is its ability to mine rules from large-scale data. Secondly, it appears that the class-imbalance facet of the tasks severely harms the predictive performance of RIPPER and PART; there, KRSNB outperforms its competitors.

	NEUROTECH-PAKDD		ORANGE-KDD'09		
	2009	2010	APPET.	CHURN	UPSELL.
KRSNB	66.31	62.27	82.02	70.59	86.46
RIPPER	51.90	50.70	50.00	50.00	71.80
PART	59.40	59.20	76.40	64.70	83.50

Table 3. Comparison of AUC results for challenge data sets.

5 Discussion & Related Work

As mentioned in sec. 2, the MODL method and its extension to classification rules are at the crossroads of Bayes theory and Kolmogorov complexity [21]. Our approach is also related to Minimum Description Length principle (MDL [15]) since the cost of a rule is similar to a coding length.

About MDL, information theory and related. Some traditional rule learning methods integrate MDL principle in their mining algorithms *(i)* as a stopping criterion when growing a rule and/or *(ii)* as a selection criterion for choosing the final rule set (see e.g. [10, 23]).

The MODL method is similar to practical MDL (also called crude MDL) which aims at coding the parameters of models M and data D given the models by minimizing the total coding length $l(M) + l(D|M)$. In [20], authors develop a MDL-based pattern mining approach (KRIMP) and its extension for classification purpose. The main divergences with our work are: *(i)* the MODL hierarchical prior induces a different way of coding information; *(ii)* KRIMP is designed for Boolean data sets and works with parameters when MODL is parameter-free and handles quantitative data sets. Also related to information theory, based on recently introduced maximum entropy models, [19] suggest the ratio of Shannon information content over the description length of a tile (i.e. an itemset coupled with its support) as an interestingness measure for binary tiles in an exploratory framework.

About mining quantitative data sets. The need for handling quantitative attributes in pattern mining tasks is not new. Srikant & Agrawal [26] develop a method for mining association rule in quantitative data sets: they start from a fine partitioning of the values of quantitative attributes, then combine adjacent partitions when interesting. After pioneering work, the literature became abun-

[3] `http://sede.neurotech.com.br/PAKDD2009/` ; `http://sede.neurotech.com.br/PAKDD2010/` ; `http://www.kddcup-orange.com/`

dant; see e.g., [30, 18]. The main differences with our work come from *(i)* the way of dealing with numerical attributes *(ii)* the mining strategy. Many methods start from a fine-granularity partition of the values and then try to merge or combine them – we design on-the-fly optimized intervals and groups when mining rules. Moreover, they inherit from classical association rule framework in which parameters are to be set.

About mining strategy and sampling methods. Exhaustive search might be inefficient on large-scale binary data (with many attributes). When facing quantitative attributes, the task is much more complicated. Separate-and-conquer (or covering) strategies [14] greedily extend one rule at once and follows a sequential data coverage scheme to produce the rule set; these strategies can tackle with large data with quantitative attributes. Our randomized strategy promotes diversity by sampling the posterior distribution of SCRMs. However, we are aware of very recent pattern mining algorithms for *binary* data using advanced sampling methods like Markov chains Monte Carlo methods (see e.g. [3, 4, 27]). Notice that our method, coupling randomized sampling with instance-based strategy, may generate similar rules. As SNB is quasi-insensitive to redundant features [7], it does not echo in the predictive performance of the classification system. We did not focus on the redundancy and sampling issues in this first study, but they are planned for future work.

6 Conclusion

We have suggested a novel framework for classification rule mining in quantitative data sets. Our method stems from the generic MODL approach. The present instantiation has lead us to several significant contributions to the field: *(i)* we have designed a new interestingness measure (*level*) that allows us to naturally mark out interesting and robust classification rules; *(ii)* we have developed a randomized algorithm that efficiently mines interesting and robust rules with diversity; *(iii)* the resulting classification system is parameter-free, deals with quantitative attributes without pre-processing and demonstrates highly competitive inductive performance compared with state-of-the-art rule-based classifiers while being highly resilient to spurious patterns. The genericity of the MODL approach and its present successful instantiation to classification rules call for other intuitive extensions, e.g., for regression rules or for other pattern type in an exploratory framework (such as descriptive rule or sequence mining).

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: ACM SIGMOD'93. pp. 207–216 (1993)
2. Asuncion, A., Newman, D.: UCI machine learning repository (2010), `http://archive.ics.uci.edu/ml/`
3. Boley, M., Gärtner, T., Grosskreutz, H.: Formal concept sampling for counting and threshold-free local pattern mining. In: SIAM DM'10. pp. 177–188 (2010)
4. Boley, M., Lucchese, C., Paurat, D., Gärtner, T.: Direct local pattern sampling by efficient two-step random procedures. In: KDD'11. pp. 582–590 (2011)

5. Boullé, M.: A bayes optimal approach for partitioning the values of categorical attributes. Journal of Machine Learning Research 6, 1431–1452 (2005)
6. Boullé, M.: MODL: A bayes optimal discretization method for continuous attributes. Machine Learning 65(1), 131–165 (2006)
7. Boullé, M.: Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research 8, 1659–1685 (2007)
8. Bringmann, B., Nijssen, S., Zimmermann, A.: Pattern-based classification: A unifying perspective. In: LeGo'09 workshop @ EMCL/PKDD'09 (2009)
9. Cheng, H., Yan, X., Han, J., Hsu, C.W.: Discriminative frequent pattern analysis for effective classification. In: Proceedings ICDE'07. pp. 716–725 (2007)
10. Cohen, W.W.: Fast effective rule induction. In: ICML'95. pp. 115–123 (1995)
11. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley (2006)
12. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
13. Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: ICML'98. pp. 144–151 (1998)
14. Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Revue 13(1), 3–54 (1999)
15. Grünwald, P.: The minimum description length principle. MIT Press (2007)
16. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Expl' 11(1), 10–18 (2009)
17. Jorge, A.M., Azevedo, P.J., Pereira, F.: Distribution rules with numeric attributes of interest. In: PKDD'06. pp. 247–258 (2006)
18. Ke, Y., Cheng, J., Ng, W.: Correlated pattern mining in quantitative databases. ACM Transactions on Database Systems 33(3) (2008)
19. Kontonasios, K.N., de Bie, T.: An information-theoretic approach to finding informative noisy tiles in binary databases. In: SIAM DM'10. pp. 153–164 (2010)
20. van Leeuwen, M., Vreeken, J., Siebes, A.: Compression picks item sets that matter. In: PKDD'06. pp. 585–592 (2006)
21. Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity and Its Applications. Springer (2008)
22. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings KDD'98. pp. 80–86 (1998)
23. Pfahringer, B.: A new MDL measure for robust rule induction. In: ECML'95. pp. 331–334 (1995)
24. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A midterm report. In: ECML'93. pp. 3–20 (1993)
25. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal (1948)
26. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: SIGMOD'96. pp. 1–12 (1996)
27. Tatti, N.: Probably the best itemsets. In: KDD'10. pp. 293–302 (2010)
28. Voisine, N., Boullé, M., Hue, C.: A bayes evaluation criterion for decision trees. In: Advances in Knowledge Discovery & Management, pp. 21–38. Springer (2010)
29. Wang, J., Karypis, G.: HARMONY : efficiently mining the best rules for classification. In: Proceedings SIAM DM'05. pp. 34–43 (2005)
30. Webb, G.I.: Discovering associations with numeric variables. In: KDD'01. pp. 383–388 (2001)
31. Webb, G.I.: Discovering significant patterns. Machine Learning 68(1), 1–33 (2007)
32. Yin, X., Han, J.: CPAR : Classification based on predictive association rules. In: Proceedings SIAM DM'03. pp. 369–376 (2003)