

# Optimum simultaneous discretization with data grid models in supervised classification

## A Bayesian model selection approach

Marc Boullé

The date of receipt and acceptance will be inserted by the editor

**Abstract** In the domain of data preparation for supervised classification, filter methods for variable ranking are time efficient. However, their intrinsic univariate limitation prevents them from detecting redundancies or constructive interactions between variables. This paper introduces a new method to automatically, rapidly and reliably extract the classificatory information of a pair of input variables. It is based on a simultaneous partitioning of the domains of each input variable, into intervals in the numerical case and into groups of categories in the categorical case. The resulting input data grid allows to quantify the joint information between the two input variables and the output variable. The best joint partitioning is searched by maximizing a Bayesian model selection criterion. Intensive experiments demonstrate the benefits of the approach, especially the significant improvement of accuracy for classification tasks.

**Keywords** Data preparation · Discretization · Feature selection · Model selection · Supervised classification

**Subject classification** AMS 62H17, AMS 62H20, AMS 62H30

## 1 Introduction

In a data mining project, the data preparation phase aims at constructing a data table for the modeling phase (Pyle, 1999; Chapman et al., 2000). The data preparation is both time consuming and critical for the quality of the mining results. It mainly consists in a search of an efficient data representation, based on variable selection. In this paper we will concentrate on a

---

Marc Boullé  
Orange Labs, 2, avenue Pierre Marzin,  
22300 Lannion, France  
E-mail: marc.boullé@orange-ftgroup.com

---

supervised classification problem with  $p$  categorical or numerical input variables and one output categorical variable with  $J$  classes (output category) and assume that a pre-specified classifier is to be used. The basic problem is to select or transform the variables in a way that maintains as much their classificatory information as possible. In this situation, the purpose of variable selection is three-fold: to improve the classifier accuracy, to reduce the training and deployment time, and to ease the comprehensibility of the classifier (Guyon and Elisseeff, 2003; Guyon et al., 2006). Two main approaches, filter and wrapper (Kohavi and John, 1997), have been studied in the literature. Filter methods consider the correlation between the input variables and the output variable as a pre-processing step, independently of the chosen classifier. Wrapper methods search the best subset of variables for a given classification technique, used as a black box. Wrapper methods, which are time consuming, are restricted to the modeling phase of data mining, as a post-optimization of a classifier. Filter methods are better suited to the data preparation phase, since they can be combined with any data modeling approach. In this paper, we focus on the filter approach.

### 1.1 Ranking methods and one-dimensional discretization

Univariate filter methods, also called ranking methods, select informative variables from a large set of candidate variables by ranking them individually according to a specified criterion, and then choosing the set of “best” ones for classification purposes, e.g., those where the criterion exceeds a given threshold. The simplest way to determine this threshold is to keep as many variables as the classification technique (often constrained by scalability issues) can handle. Another classical approach is to estimate the parameters of the classifier with several subsets of variables of increasing size. The best subset is chosen according to a trade-off between the accuracy of the classifier and the size of the subset.

The most commonly used ranking methods are based on statistical tests (Saporta, 1990) that consider the correlation between an input variable and the classificatory output variable, such as the chi-square test for categorical input variables, or Student or Fisher-Snedecor tests for numerical input variables. These statistical tests are easy to apply, but they suffer from serious limitations. They are restricted to a strong dichotomy between dependent and independent variables, which does not provide a reliable ranking of the input variables. They are also subject to strong constraints (minimum expected frequency in each cell of the contingency table for categorical variable, Gaussian distribution for numerical variables). Many alternative measures of associations between two variables have been studied in the context of decision trees (Kass, 1980; Breiman et al., 1984; Quinlan, 1993; Zighed and Rakotomalala, 2000). These criteria are based on a partition of the domain of the input variable and the consideration of the dependence between the corresponding discretized input variable and the output variable. Supervised discretization

---

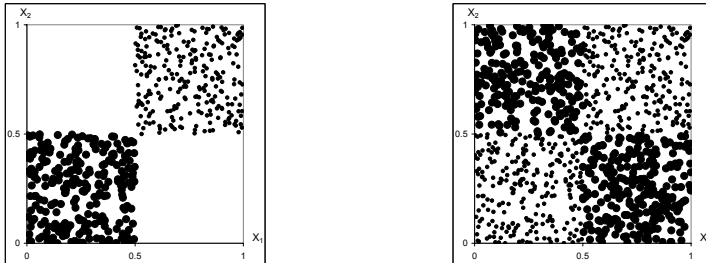
methods split the numerical domain into a set of intervals and supervised grouping methods partition the input categories into groups. Fine grained partitions allow an accurate discrimination of the output classes, whereas coarse grained partitions tend to be more reliable.

Many discretization criteria have been studied in the literature. Error based criteria (Holte, 1993; Maass, 1994; Kurgan and Cios, 2004) aim at minimizing the classification error and need a user or internal parameter to constrain the number of intervals. They focus only on the most represented output class in each interval and ignore the distribution of the output classes, which limits their classification performance (Kohavi and Sahami, 1996). Chi-square based criteria have been exploited in a top-down (Bertier and Bouroche, 1981) or bottom-up (Kerber, 1992) discretization algorithm to produce multi-interval discretizations from elementary binary split or merge decisions. In (Boullé, 2004), the confidence level related to the chi-square statistics is directly exploited to provide a multi-interval discretization criterion. These criteria suffer from limitations of the statistical test procedure, e.g. the minimum frequency per interval (Cochran, 1954; Connor-Linton, 2003). Entropy based criteria consider the distribution of the output classes. In (Quinlan, 1986, 1993), the method is confined to binary discretization. To extend the approach to multi-interval discretization, the BalancedGain method (Kononenko et al., 1984) penalizes the entropy by the number of intervals, while the Fusinter method (Zighed et al., 1998) penalizes intervals with small frequencies. A more principled approach based on the minimum description length principle (Rissanen, 1978) is applied in the MDLPC (Minimum Description Length Principal Cut) method (Fayyad and Irani, 1992). In the case of a binary discretization, to decide whether to split an interval, or not, the MDLPC criterion encodes the description length of the model (cut or not cut) plus the description length of the data given the model. When the number of intervals of the discretization is a free parameter, the trade-off between information and robustness is an issue. In the MODL (Minimum Optimized Description Length) approach, supervised discretization (Boullé, 2006) (or grouping (Boullé, 2005)) is treated as a nonparametric model of conditional probability of the output classificatory variable given an input variable. The best partition is defined and constructed by using a Bayesian model selection approach, and the posterior probability of this best partition provides a measure of association that is both accurate and reliable.

## 1.2 Limits of ranking methods

Ranking methods suffer from their univariate limitation, being unable to reveal interactions between input variables. For example, they are not able to detect if one of two variables is redundant and insofar only one of them should be used for classification purposes. On the other hand, input variables might be uninformative when considered alone and strongly informative when considered simultaneously. These two cases are illustrated for a two-class problem

in Fig. 1, using two-dimensional scatterplots where the points are drawn in different sizes according to their class membership. The left diagram shows the case of two redundant variables. The right diagram corresponds to an XOR pattern: the distribution of each input variable taken alone is a mixture of the two-class specific distribution within the classes, whereas a joint consideration of the two variables allows a perfect discrimination of the output classes.



**Fig. 1** Multiple scatterplots for two input variables  $X_1$  and  $X_2$ , and two output classes (small and large circles). The left diagram shows the case of two redundant variables and the right diagram the case of two jointly informative variables

In the case of two numerical input variables, multiple scatterplots are a popular visualization technique to detect interactions between the input and output variables. Scatterplot matrices (Carr et al., 1987) extend this technique to sets of more than two input variables and allow to show all pairwise interactions between the variables. These methods are widely used in exploratory data analysis, but they do not provide a measure of the joint information contained in the variable pairs. Furthermore, these methods do not apply in the case of large numbers of variables: 100 input variables lead to  $4950 = 100 * 99/2$  scatterplots, which cannot be managed by the data analyst. An automatic method for estimating interactions between variables is needed due to the increasing number of variables in datasets.

### 1.3 Our contribution

Our goal is to provide an efficient method to discretize in an optimum way pairs of input variables in the context of data preparation for supervised classification. In this paper, we extend the MODL approach to the bivariate case for any pair of input variables, numerical, categorical or mixed types. Each input variable is partitioned, into intervals in the numerical case and into groups of categories in the categorical case. This joint partitioning defines a distribution of the instances in a bi-dimensional input data grid. The correlation between the cells of this data grid and the output classes allows to quantify the joint classificatory information. The trade-off between accuracy and robustness is established using a Bayesian model selection approach. This provides a criterion for any simultaneous partitioning of the input variables. Several optimization

heuristics, including pre-optimization and post-optimization are proposed to search the best possible simultaneous partitioning in a super-linear computation time.

Our method combines several interesting properties. It is able to manage both numerical and categorical input variables. It is nonparametric and non asymptotic. It is regularized in order to tackle the sparseness problem and optimally balance between the accuracy and the robustness of the discretization. The optimization process is computationally efficient and results in super-linear computation time. Finally, it also provides a filter criterion for the ranking of pairs of variables and it builds easily understandable models.

The paper is organized as follows. Section 2 summarizes the MODL method in the univariate case. Section 3 introduces the extension of the approach to the bivariate case and presents the resulting criterion. Section 4 summarizes the optimization algorithms, which are detailed in (Boullé, 2008). Section 5 demonstrates the benefits of the approach on real datasets, both for the data preparation and data modeling phases of data mining. Finally, Section 6 gives a summary.

#### 1.4 Related work

Many criteria such as Pearson's chi-square, Tschuprow's  $t$  or Cramer's  $v$  have been studied in the literature (Olszak and Ritschard, 1995; Ritschard and Nicoloyannis, 2000) to measure the association between variables. It is noteworthy that these association measures deal with two variables of a cross-table, while our approach considers the association between a pair of input variables and one output variable, which involves three variables.

Multivariate discretization or similar techniques have already been proposed in various contexts. For example, the joint partitioning of the lines and rows of contingency table has been studied in the general case (Nadif and Govaert, 2005) for data exploration, or in the case of decision trees for the joint partitioning of one input variable and the output variable (Zighed et al., 2005). Multivariate discretization has also been developed in the case of association rule mining (Bay, 2001), learning the structure of Bayesian network (Steck and Jaakkola, 2004) or for decision rule induction (Kwedlo and Kretowski, 1999).

The main differences with these approaches are the choice of our family of models, our Bayesian approach for model selection and our optimization algorithm with super-linear computation time.

## 2 The MODL univariate supervised partitioning methods

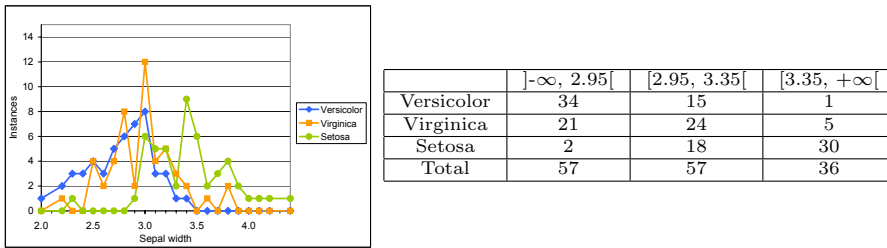
This section summarizes the MODL approach for discretization (Boullé, 2006) and grouping (Boullé, 2005). Extensive comparative experiments have demonstrated that the univariate MODL preprocessing methods significantly outperforms alternative state of the art methods. The approach is quickly presented

in this section, and applied and extended to the bivariate case with extensive explanations in Section 3.

## 2.1 The MODL discretization method for continuous variables

Let us consider a classification problem with one single numerical input variable  $X$ , a categorical output variable  $Y$  with  $J$  classes and  $D = \{(x_n, y_n); 1 \leq n \leq N\}$  a data sample of  $(X, Y)$  of size  $N$ .

The objective of supervised discretization is to induce a list of intervals which splits the numerical domain of  $X$ , while keeping the classificatory information relative to the output variable. A compromise must be found between accuracy (sufficient number of intervals to precisely estimate the conditional distribution of the output variable) and robustness (sufficient sample size in every interval to ensure generalization). For example, we present on the left of Fig. 2 the number of instances of each output class of the Iris dataset (Blake and Merz, 1996) w.r.t. the sepal width variable. The problem is to find the partition of the numerical domain of sepal length into intervals which maintains optimally the classificatory information about the three Iris classes.



**Fig. 2** MODL discretization of the Sepal Width variable for the classification of the Iris dataset in 3 classes Versicolor, Virginica and Setosa

In the MODL approach (Boullé, 2006), the discretization is turned into a model selection problem. Instead of defining the intervals by their boundaries in the numerical domain of  $X$ , they are defined by their frequencies, which makes the approach invariant w.r.t. any monotone transformation of the input variable and robust w.r.t. atypical values. The parameters of a discretization model  $M$  are the number of intervals  $I$ , the frequencies of the intervals  $\{N_i\}_{1 \leq i \leq I}$  and the frequencies of the output classes  $\{N_{ij}\}_{1 \leq i \leq I, 1 \leq j \leq J}$  in each interval. A prior distribution is proposed on this model space. This prior exploits the hierarchy of the parameters: the number of intervals is first chosen, then the frequencies of the intervals and finally the output frequencies. The choice is uniform at each stage of the hierarchy. Finally, we assume that the parameters of the multinomial distributions of the output classes in each interval are independent from each other. A Bayesian approach is applied to select the best discretization model, which is found by maximizing

the probability  $p(M|D)$  of the model given the data. Using the Bayes formula  $P(M, D) = P(M)P(D|M) = P(D)P(M|D)$  and since the probability  $p(D)$  is the same for all discretization models, this is equivalent to maximizing  $p(M)p(D|M)$ . Taking the negative log of the probabilities, we can obtain the following expression.

$$\log N + \log \binom{N + I - 1}{I - 1} + \sum_{i=1}^I \log \binom{N_i + J - 1}{J - 1} + \sum_{i=1}^I \log \frac{N_i!}{N_{i1}!N_{i2}!\dots N_{iJ}!} \quad (1)$$

The first term of the criterion stands for the choice of the number of intervals and the second term for the choice of the frequencies of the intervals. The third term corresponds to the choice of the parameters of the multinomial distributions of the output classes in each interval and the last term represents the conditional likelihood of the data given the model. Therefore “complex” models with large numbers of intervals are penalized.

Once the optimality of the criterion is established, the problem is to design a search algorithm in order to find a discretization model that minimizes the criterion. In (Boullé, 2006), a standard greedy bottom-up heuristic is used to find a good discretization. In order to further improve the quality of the solution, the MODL algorithm performs post-optimizations based on hill-climbing search in the neighborhood of a discretization. The neighbors of a discretization are defined by combinations of interval splits and interval merges. Overall, the time complexity of the algorithm is  $O(JN \log N)$ .

The MODL discretization method for classification provides the most probable discretization given the data sample. Extensive comparative experiments report high quality performance. In the Iris example, the three intervals of the MODL discretization are shown on the right of Fig. 2. The contingency table on the right gives us comprehensible rules such as “for a sepal width less than 2.95, the probability of occurrence of the Versicolor class is  $34/57 = 0.60$ ”.

## 2.2 The MODL grouping method for categorical variables

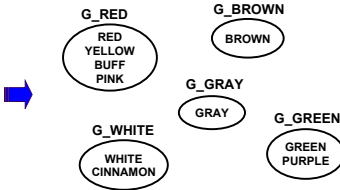
Let us consider a classification problem with one single categorical input variable  $X$  with  $V$  categories, a categorical output variable  $Y$  with  $J$  classes and  $D = \{(x_n, y_n); 1 \leq n \leq N\}$  a data sample of  $(X, Y)$  of size  $N$ .

Categorical input variables are analyzed in a way similar to that of numerical variables, owing to a partitioning model of the input categories. In the numerical case, the input values are constrained to be adjacent and the only considered partitions are the partitions into intervals. In the categorical case, there are no such constraints between the categories and any partition into groups of categories is possible<sup>1</sup>. For instance, Fig. 3 illustrates the grouping of the categories of the Cap Color input variable of the Mushroom dataset

<sup>1</sup> Categorical variables with ordered values (ordinal variables) are treated as numerical variables on the basis of the rank of the values.

(Blake and Merz, 1996) for the classification of mushrooms into two classes: edible or poisonous. The input categories provide a fine grained estimation of conditional probabilities of the two output classes. Grouping the input categories and estimating group-specific conditional probabilities leads to different trade-offs between accuracy and robustness of the estimation: fine grained groupings are more accurate whereas coarse grained groupings are more robust. The problem is obtain a reduced number of groups of categories, while keeping as much classificatory information as possible. Producing a good grouping is harder with large numbers of input categories since the risk of overfitting the data increases. In the extreme situation where the number of input categories is the same as the number of instances, overfitting is obviously so important that efficient grouping methods should produce one single group, leading to the elimination of the variable.

Category	edible	poisonous	Frequency
BROWN	55.2%	44.8%	1610
GRAY	61.2%	38.8%	1458
RED	40.2%	59.8%	1066
YELLOW	38.4%	61.6%	743
WHITE	69.9%	30.1%	711
BUFF	30.3%	69.7%	122
PINK	39.6%	60.4%	101
CINNAMON	71.0%	29.0%	31
GREEN	100.0%	0.0%	13
PURPLE	100.0%	0.0%	10



**Fig. 3** MODL grouping of the categories of the Cap Color input variable for the classification of the Mushroom dataset in two classes edible and poisonous

The parameters of a grouping model  $M$  are the number of groups  $I$ , the partition of the  $V$  input categories into  $I$  groups and the frequencies of the output classes  $\{N_{ij}\}_{1 \leq i \leq I, 1 \leq j \leq J}$  in each group. The frequencies of the groups  $\{N_{i.}\}_{1 \leq i \leq I}$  are derived from the definition of the partition and from the input data. The Bayesian model selection approach is applied like in the discretization case and allows to obtain the expression given in formula (2). This formula has a similar structure as that of formula (1). The two first terms correspond to the prior distribution of the partitions of the input categories, into groups in formula (2) and into intervals in formula (1). The two last terms are the same in both formula.

$$\log V + \log B(V, I) + \sum_{i=1}^I \log \binom{N_{i.} + J - 1}{J - 1} + \sum_{i=1}^I \log \frac{N_{i.}!}{N_{i1}! N_{i2}! \dots N_{iJ}!} \quad (2)$$

$B(V, I)$  is the number of divisions of the  $V$  categories into  $I$  groups (with eventually empty groups). When  $I = V$ ,  $B(V, I)$  is the Bell number. In the general case,  $B(V, I)$  can be written as  $B(V, I) = \sum_{i=1}^I S(i, V)$ , where  $S(i, V)$  is the Stirling number of the second kind (Abramowitz and Stegun, 1970),



which stands for the number of ways of partitioning a set of  $V$  elements into  $i$  nonempty sets.

In (Boullé, 2005), a standard greedy bottom-up heuristic is proposed to find a good grouping of the input categories. Several pre-optimization and post-optimization steps are incorporated, in order to both ensure an algorithmic time complexity of  $O(JN \log(N))$  and to obtain accurate groupings.

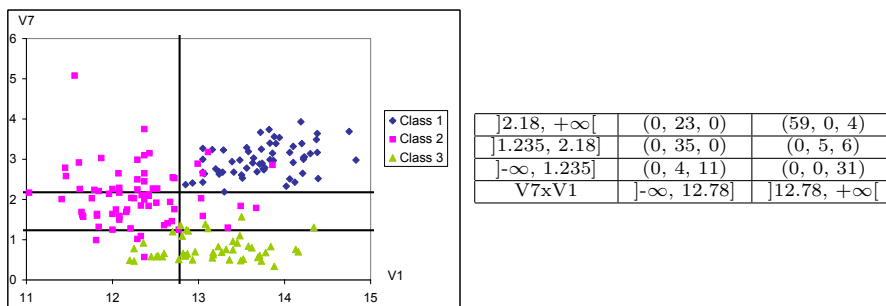
### 3 Extension to supervised bivariate discretization

In this section, we extend the MODL methods to the supervised discretization of pairs of input variables. We first introduce the approach using an illustrative example and then present the bivariate criterion in the case of two numerical variables. We generalize the criterion to any case of pairs of input variables and finally introduce the compression gain, a normalized version of the criteria.

We now consider  $X = (X_1, X_2)$  a pair of input variables,  $Y$  a categorical output variable with  $J$  classes and  $D = \{(x_n, y_n); 1 \leq n \leq N\}$  a data sample of  $(X, Y)$  of size  $N$ .

#### 3.1 Interest of the joint partitioning of two input variables

Fig. 4 draws the multiple scatter plot (per output class) of the input variables V1 and V7 of the Wine dataset (Blake and Merz, 1996). This diagram allows to visualize the conditional probability of the output classes (Class 1, Class 2, Class 3) given the pair of input variables. The V1 variable taken alone cannot separate Class 1 from Class 3 for input values greater than 13. Similarly, the V7 variable is a mixture of Class 1 and Class 2 for input values greater than 2. Taken jointly, the two input variables allow a better separation of the output classes.



**Fig. 4** Multiple scatterplot (per output class) of the input variables V1 and V7 of the Wine dataset. The optimal MODL supervised bivariate partition of the input variables is drawn on the multiple scatterplot, and the triplet of output class frequencies per data grid cell is reported in the right table

Extending the univariate case, we partition the Cartesian product of the two input variables  $V1$  and  $V7$  by considering the empirical distribution of the observed input data pairs and by quantifying the relationship between these pairs and the classificatory output variable. Each input variable is partitioned into a set of *parts* (intervals in the numerical case and groups of categories in the categorical case). The Cartesian product of the univariate input partitions defines a *data grid*, which partitions the instances into a set of *data cells*. Each data cell is defined by a pair of parts. The connection between the input variables and the output variable is estimated owing to the distribution of the output classes in each cell of the data grid. It is noteworthy that the considered data grid partition, which is a direct product of the two univariate input partitions, can be factorized on the input variables.

For instance in Fig. 4, the  $V1$  variable is discretized into 2 intervals (one boundary 12.78) and the  $V7$  variable into 3 intervals (two boundaries 1.235 and 2.18). The instances of the dataset are distributed in the resulting bidimensional data grid. In each cell of the grid, the distribution of the output classes can be estimated by counting. For example, the right table in Fig. 4 shows that the cell defined by the intervals  $]12.78, +\infty[$  on  $V1$  and  $]2.18, +\infty[$  on  $V7$  contains 63 instances. These 63 instances are distributed on 59 instances for Class 1 and 4 instances for Class 3.

*Problem of model selection.* Coarse grained data grids tend to be reliable, whereas fine grained data grids allow a better separation of the output classes. Data grid models are very expressive and selecting the best model is an issue:

- What is the correct number of intervals for each input variable?
- How to chose the frequency of each interval?
- How to characterize the lack of discriminant information?

Moreover, in the context of bivariate preprocessing for data preparation, the following questions arise:

- How to compare two data grid models for a given pair of input variables?
- How to compare the discriminant information for different pairs of input variables?

We will answer to these questions at the end of Section 3.4.

### 3.2 Discretization of pairs of numerical variables

$X = (X_1, X_2)$  is a pair of numerical input variables and  $Y$  a categorical output variable. We are looking for a model of the conditional probability  $P(Y|X)$  of  $Y$  given  $X$ .

The choices 1 and 2 are the basis of the MODL approach.

**Choice 1** *Choice of the ranks.*

*We require that a good estimation of the conditional probability  $P(Y|X)$  should*

be invariant w.r.t. any monotonous transformation of the input variables and robust w.r.t. atypical values (outliers). Given this requirement, we choose to exploit the ranks of the input values in the data sample rather than the values themselves. Our objective is then to describe the distribution of  $Y$  given the rank of  $X_1$  and  $X_2$ , instead of given the value of  $X_1$  and  $X_2$ .

**Choice 2** *Choice of the model precision.*

Given that we have a finite data sample of size  $N$ , it does not look realistic to approximate the true conditional probability with a precision better than  $1/N$ . We thus confine the domain of the model parameters to a finite number of frequencies (rather than continuous distributions in  $[0, 1]$ ), on the basis on instance counts in the data sample.

Although a precision of  $1/N$  might look unnecessarily small, a more classical precision of  $1/\sqrt{N}$  (like in the variance of the binomial distribution) is too coarse to detect fine grain patterns. This choice is justified by theoretical and empirical evidences in (Boullé, 2006), where intervals with very few instances are reliably constructed by the MODL univariate discretization method.

In Definition 1, these modeling choices are exploited to define a family of bivariate partitioning models called data grid models, where the conditional probability  $P(Y|X)$  is assumed to be constant in each cell of the data grid.

**Definition 1** A data grid model is a bivariate partitioning model defined by a partition of each input variable into a set of intervals and by a multinomial distribution of the output classes in each cell of the data grid resulting from the Cartesian product of the univariate partitions.

**Notations.**

- $N$ : number of instances,
- $J$ : number of output classes,
- $I_1, I_2$ : number of intervals for each input variable,
- $N_{i_1..}$ : number of instances in the interval  $i_1$  of variable  $X_1$ ,
- $N_{..i_2}$ : number of instances in the interval  $i_2$  of variable  $X_2$ ,
- $N_{i_1i_2..}$ : number of instances in the input data cell  $(i_1, i_2)$ ,
- $N_{i_1i_2j}$ : number of instances of output class  $j$  in the input data cell  $(i_1, i_2)$ .

A data grid model  $M$  describes the distribution of the output classes given a partition of the Cartesian product of the input variables. It is completely defined by the numbers of intervals  $I_1$  and  $I_2$ , the frequencies of the intervals  $\{N_{i_1..}\}$  and  $\{N_{..i_2}\}$  and the distribution of the output classes  $\{N_{i_1i_2j}\}$  in each cell  $(i_1, i_2)$  of the data grid. It is noteworthy that the numbers of instances per cell  $\{N_{i_1i_2..}\}$  do not belong to the parameters of the data grid models: they are derived from the definition of the two univariate input partitions and from the data sample  $D$ .

The available data  $D = \{(x_n, y_n); 1 \leq n \leq N\}$  consists of input data  $D_X = \{x_n; 1 \leq n \leq N\}$  and output data  $D_Y = \{y_n; 1 \leq n \leq N\}$ .

Any input data is used to define the family of models introduced in Definition 1. The boundaries of the univariate partitions are calculated from the input values and the frequencies of the input data cells are calculated from the data sample and from the definition of the univariate partitions. In that sense, the data grid models are data dependent. What is described in the model is the association between the input variables and the output variable.

The objective is now to select the best model  $M$  on the basis on the available data sample  $D$ . Whereas the input data only is used to define the family of models, the output data is used to select the best model. We apply a Bayesian approach (Robert, 1997; Bernardo and Smith, 2000) to select the maximum a posteriori (MAP) model. Choosing the MAP estimator is justified in the context of data preparation, since an objective function is not always available at this stage of data analysis.

Selecting the MAP model, we have to maximize

$$P(M|D) = \frac{P(M)P(D|M)}{P(D)}.$$

Since the probability  $P(D)$  is constant when varying the model, this is equivalent to maximizing  $P(M)P(D|M)$ . Exploiting the structure of the parameters of a data grid model  $M$  and assuming that the parameters of the discretizations of the two input variables are a priori independent, we obtain

$$\begin{aligned} P(M)P(D|M) &= P(I_1)P(\{N_{i_1..}\}|I_1)P(I_2)P(\{N_{.i_2.}\}|I_2) \\ &P(\{N_{i_1i_2j}\}|I_1, I_2, \{N_{i_1..}\}, \{N_{.i_2.}\})P(D|M). \end{aligned}$$

We now assume that the parameters of the multinomial distributions of the output classes are independent for each cell of data grid. The interest of this assumption is threefold: it improves understandability, owing to a focus on models with discriminating behavior per data grid cell, it provides an analytic criterion for the posterior probability of data grid models and it permits the development of efficient optimization heuristics (see Section 4). It is noteworthy that the assumption of independence per cell is involved by the usual assumption of independently and identically distributed data (i.i.d. assumption).

Denoting  $D_{i_1i_2}$  the subset of  $D$  belonging to the data cell  $(i_1, i_2)$ , we obtain

$$\begin{aligned} P(M)P(D|M) &= P(I_1)P(\{N_{i_1..}\}|I_1)P(I_2)P(\{N_{.i_2.}\}|I_2) \\ &\prod_{i_1=1}^{I_1} \prod_{i_2=1}^{I_2} P(\{N_{i_1i_2j}\}|I_1, I_2, \{N_{i_1..}\}, \{N_{.i_2.}\}) \\ &\prod_{i_1=1}^{I_1} \prod_{i_2=2}^{I_2} P(D_{i_1i_2}|M). \end{aligned} \quad (3)$$

In order to compute the criterion, we introduce in Definition 2 a prior distribution on the parameters of the data grid models. This prior makes explicit the independence assumptions, exploits the hierarchy of the parameters and is uniform at each stage of this hierarchy.

**Definition 2** The hierarchical prior of the parameters of data grid models is defined as follows:

- the numbers of input intervals are independent from each other, and uniformly distributed between 1 and  $N$ ,
- for each input variable and for a given number of intervals, every partition into intervals is equiprobable,
- for each cell of the data grid, all the parameters of the multinomial distribution of the output classes are equiprobable,
- the parameters of the multinomial distributions of the output classes in each cell are independent from each other.

In the Bayesian approach, the choice of the prior is either subjective (Goldstein, 2006), where the choice comes from the prior knowledge of the data analyst, or objective (Berger, 2006), where the aim is to be as uninformative as possible. In the context of data preparation, we have adopted an objective approach inspired from the minimum description length (Rissanen, 1978).

Owing to the definition of the model space and its prior distribution, the Bayes formula is applicable to exactly calculate the prior probabilities of the models and the probability of the data given a model. Theorem 1 introduces the MODL criterion.

**Theorem 1** *The negative log of the posterior probability of a data grid model distributed according to the hierarchical prior is given by the following formula:*

$$c(M) = \log N + \log \binom{N + I_1 - 1}{I_1 - 1} + \log N + \log \binom{N + I_2 - 1}{I_2 - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2} + J - 1}{J - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \dots N_{i_1 i_2 J}!} \quad (4)$$

*Proof* The prior probability of a data grid model  $M$  is that of the model parameters  $\{I_1, I_2, \{N_{i_1 \cdot}\}_{1 \leq i_1 \leq I_1}, \{N_{\cdot i_2}\}_{1 \leq i_2 \leq I_2}, \{N_{i_1 i_2 j}\}_{1 \leq i_1 \leq I_1, 1 \leq i_2 \leq I_2, 1 \leq j \leq J}\}$ .

Using the hierarchical prior and the independence assumptions, we have decomposed the posterior probability of a model in formula (3).

Let us now inspect each elementary term of this formula, exploiting the hypotheses introduced in Definition 2.

The first hypothesis of the prior distribution is that the number of intervals is uniformly distributed between 1 et  $N$  for each input variable. Thus, we get

$$P(I_1) = P(I_2) = \frac{1}{N}.$$

The second hypothesis of the prior distribution is that all the division of  $N$  instances into  $I_1$  (resp.  $I_2$ ) intervals are equiprobable. Computing the

probability of one set of intervals turns into the combinatorial calculation of the number of possible interval sets. Dividing  $N$  instances into  $I_1$  intervals is equivalent to decomposing the natural number  $N$  as the sum of the frequencies  $N_{i_1..}$  of the intervals. Using combinatorics, that number of choices of such any  $\{N_{i_1..}\}_{1 \leq i_1 \leq I_1}$  is equal to  $\binom{N+I_1-1}{I_1-1}$ . Since each set of discretization parameters is equiprobable, we obtain

$$P(\{N_{i_1..}\}|I_1) = \frac{1}{\binom{N+I_1-1}{I_1-1}} \quad \text{and} \quad P(\{N_{i_2..}\}|I_2) = \frac{1}{\binom{N+I_2-1}{I_2-1}}.$$

Given two univariate discretizations of the input variables  $X_1$  and  $X_2$ , the frequency  $N_{i_1 i_2}$  of each cell of the data grid can be derived from the input data sample. According to the third hypothesis of the prior distribution, in each cell  $(i_1, i_2)$ , all the parameters of the multinomial distributions of the  $N_{i_1 i_2}$  instances of the cell on the  $J$  output classes are equiprobable. Calculating the probability of one such set of multinomial parameters is a combinatorial problem, which reduces to computing the number of ways of decomposing a natural number  $N_{i_1 i_2}$  as a sum of  $J$  terms. Since each set of multinomial parameters is equiprobable, we obtain

$$P(\{N_{i_1 i_2 j}\}|I_1, I_2, \{N_{i_1..}\}, \{N_{i_2..}\}) = \frac{1}{\binom{N_{i_1 i_2} + J - 1}{J - 1}}.$$

We now have to calculate the conditional likelihood term in each data grid cell, that is to compute the probability of observing the output classes of a cell given the parameters of the multinomial distribution in this cell. The number of ways of observing  $N_{i_1 i_2}$  instances distributed according to a multinomial distribution is given by the multinomial coefficient  $\frac{N_{i_1 i_2}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \dots N_{i_1 i_2 J}!}$ . The conditional likelihood per cell is thus

$$\frac{1}{\frac{N_{i_1 i_2}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \dots N_{i_1 i_2 J}!}}.$$

Finally, we replace each prior and conditional likelihood term in formula (3) and get

$$P(M)P(D|M) = \frac{1}{N} \frac{1}{\binom{N+I_1-1}{I_1-1}} \frac{1}{N} \frac{1}{\binom{N+I_2-1}{I_2-1}} \prod_{i_1=1}^{I_1} \prod_{i_2=1}^{I_2} \frac{1}{\binom{N_{i_1 i_2} + J - 1}{J - 1}} \\ \prod_{i_1=1}^{I_1} \prod_{i_2=1}^{I_2} \frac{1}{\frac{N_{i_1 i_2}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \dots N_{i_1 i_2 J}!}}.$$

Taking the negative log of the probabilities, the maximization problem turns into the minimization of the claimed criterion.  $\square$

As in the case of univariate discretization (formula (1)), the two first terms in formula (4) correspond to the prior probability of the parameters (number of intervals and choice of the interval frequencies) of the discretization of the input variable  $X_1$ . Similarly, the two following terms correspond to the prior probability of the discretization of the input variable  $X_2$ . The binomial term in the first double sum represents the choice of the multinomial distribution of the output classes in each cell. The multinomial coefficient in the last double sum represents the conditional likelihood of the output classes given the data grid model.

### 3.3 Partitioning of any pair of variable

In the case of two categorical input variables  $X_1$  and  $X_2$  with  $V_1$  and  $V_2$  categories, we apply the same approach. The  $X_1$  variable is partitioned into  $I_1$  groups of categories (instead of intervals in the numerical case) and the  $X_2$  variable into  $I_2$  groups. The distribution of the output classes is described in each cell of the data grid resulting from the joint partitioning of the input variables. Compared to the numerical case, the only change is the prior distribution of each univariate partition. The impact in formula (4) is to replace the terms related to the prior distribution of the partition into intervals (two first terms of the univariate discretization of formula (1)) by the corresponding grouping terms (two first terms of the univariate grouping of formula (2)). We then obtain the following expression in the case of a data grid model  $M$  with two categorical input variables.

$$c(M) = \log V_1 + \log B(V_1, I_1) + \log V_2 + \log B(V_2, I_2) + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2} + J - 1}{J - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \dots N_{i_1 i_2 J}!} \quad (5)$$

In the mixed case of one categorical input variable  $X_1$  with  $V_1$  categories and one numerical input variable  $X_2$ , the first variable is grouped and the second one is discretized, and we obtain the following expression.

$$c(M) = \log V_1 + \log B(V_1, I_1) + \log N + \log \binom{N + I_2 - 1}{I_2 - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \binom{N_{i_1 i_2} + J - 1}{J - 1} + \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \log \frac{N_{i_1 i_2}!}{N_{i_1 i_2 1}! N_{i_1 i_2 2}! \dots N_{i_1 i_2 J}!} \quad (6)$$

### 3.4 Compression gain

The criterion  $c(M)$  given in formulas (4), (5), (6) is related to the probability that a data grid model  $M$  explains the output variable given the input

variables. The criterion  $c(M)$  can also be interpreted as the ability of a data grid model to encode the output classes given the input values, since negative log of probabilities are no other than coding lengths (Shannon, 1948; Weaver and Shannon, 1949). Let  $M_\emptyset$  be the null model with only one part for each univariate partition and one cell in the data grid.  $c(M_\emptyset)$  represents the coding length of the output classes when no input information is used. In the case of bivariate discretizations for example (criterion given in formula (4)), the coding length of the null model  $M_\emptyset$  reduces to

$$c(M_\emptyset) = 2 \log N + \log \binom{N+J-1}{J-1} + \log \frac{N_{11}!}{N_{111}!N_{112}!\dots N_{11J}!}, \quad (7)$$

which corresponds to posterior probability of a multinomial distribution of the output classes, independently of the input variables. Using the Stirling's approximation  $\log N! = N(\log N - 1) + O(\log N)$ , we get

$$\begin{aligned} c(M_\emptyset) &= 2 \log N + \log \binom{N+J-1}{J-1} + \log \frac{N_{11}!}{N_{111}!N_{112}!\dots N_{11J}!}, \\ &= (N+J-1)(\log(N+J-1) - 1) - N(\log N - 1) \\ &\quad + N_{11}(\log N_{11} - 1) - \sum_{j=1}^J N_{11j}(\log N_{11j} - 1) + O(\log N), \\ &= -N_{11} \sum_{j=1}^J \frac{N_{11j}}{N_{11}} \log \frac{N_{11j}}{N_{11}} + O(\log N). \end{aligned}$$

Thus  $c(M_\emptyset)$  is asymptotically equal to  $N$  times the Shannon's entropy of the output variable.

More complex data grid models may better compress the output classes, since the entropy of the output classes is defined locally to each input cell. Fine grained cells allow to identify input regions where the output entropy is low (unbalanced mixture of the output classes), but too complex data grid models with many cells are penalized with an increasing coding length of the model parameters.

*Compression gain.* Given these probabilistic and compression interpretations, we propose to use the criterion  $c(M)$  to build a relevance criterion for each pair of input variables. The variable pairs can be sorted by decreasing probability of explaining the output variable. In order to provide a normalized indicator, we consider the following transformation of  $c(M)$ :

$$g(M) = 1 - \frac{c(M)}{c(M_\emptyset)}. \quad (8)$$

The compression gain  $g(M)$  holds its values between 0 and 1 for models which are better than the null model ( $g(M)$  is negative otherwise). It has value 0 for the null model and is maximal when the best possible explanation of the output classes conditionally to the pair of input variables is achieved.



*Answer to the problem of model selection.* We can now provide an answer to the questions raised at the end of Section 3.1. The numbers of intervals and the frequencies of the intervals are determined by the minimization of the criterion in formula (4). The lack of discriminant information corresponds to the null model  $M_\emptyset$  with only one interval for each univariate discretization and one cell in the data grid, that is to a compression gain of 0.

To compare two data grid models  $M$  and  $M'$  for a given pair of input variables, we can use the probabilistic interpretation of the criterion  $c(M) = -\log P(M) - \log P(D|M)$ . Thus, the ratio of the posterior probabilities of the models is in direct relation to the difference of their criterion according to

$$\frac{P(M|D)}{P(M'|D)} = e^{-(c(M)-c(M'))}.$$

This formula can also be exploited to compare the predictive importance of different pairs of input variables. Actually, the value of the criterion  $c(M^*)$  for the optimal data grid model can be interpreted as the probability that a pair of input variables  $(X_1, X_2)$  explains the output variable  $Y$  on the basis of a data grid model.

#### 4 Optimization algorithms

The space of data grid models is so large that straightforward algorithms almost surely fail to obtain good solutions within a practicable computational time. Given that the MODL criterion is optimal, the design of sophisticated optimization algorithms is both necessary and meaningful. Such algorithms are described in (Boullé, 2008). They finely exploit the sparseness of the data grids and the additivity of the MODL criterion, and allow a deep search in the space of data grid models with  $O(N)$  memory complexity and  $O(N \log N)$  time complexity.

In this section, we give an overview of the data grid optimization algorithms which are fully detailed in (Boullé, 2008). Let us first focus on the case of two numerical input variables. The optimization of a data grid is a combinatorial problem. For each input variable  $X_1$  and  $X_2$ , there are  $2^N$  possible univariate discretizations, which represents  $(2^N)^2$  possible bivariate discretizations. An exhaustive search through the whole space of models is unrealistic. We describe in algorithm 1 a greedy bottom up merge heuristic (GBUM) to optimize the data grids. The method starts with the maximum data grid  $M_{Max}$ , which corresponds to the finest possible univariate discretizations, with singleton intervals. It considers all the merges between adjacent intervals, and performs the best merge if the criterion decreases after the merge. The process is reiterated until no further merge decreases the criterion.

Each evaluation of the criterion for a data grid requires  $O(N^2)$  time, since the initial data grid model  $M_{Max}$  contains  $N^2$  cells (see formula (4)). Each step of the algorithm relies on  $O(N)$  evaluations of interval merges, and there

---

**Algorithm 1** Greedy Bottom Up Merge heuristic (GBUM)
 

---

**Require:**  $M$  {Initial data grid solution}  
**Ensure:**  $M^*, c(M^*) \leq c(M)$  {Final solution with improved cost}  
 1:  $M^* \leftarrow M$   
 2: **while** improved solution **do**  
 3:   **for all** Merge  $m$  between two parts of variable  $X_1$  or  $X_2$  **do**  
 4:      $M' \leftarrow M^* + m$  {Consider merge  $m$  on data grid  $M^*$ }  
 5:     **if**  $c(M') < c(M^*)$  **then**  
 6:        $M^* \leftarrow M'$   
 7:     **end if**  
 8:   **end for**  
 9: **end while**

---

are at most  $O(N)$  steps, since the data grid becomes equal to the null model  $M_\emptyset$  once all the possible merges have been performed. Overall, the time complexity of the algorithm is  $O(N^4)$  using a straightforward implementation of the algorithm. However, the method can be optimized in  $O(N \log N)$  time, as demonstrated in (Boullé, 2008). The optimized algorithm mainly exploits the sparseness of the data and the additivity of the criterion. Although a data grid may contain  $O(N^2)$  cells, at most  $N$  cells are non empty. Thus, each evaluation of a data grid can be performed in  $O(N)$  owing to a specific algorithmic data structure. The additivity of the criterion means that it can be decomposed on the hierarchy of the components of the data grid: variables, parts and cells. Using this additivity property, all the merges between adjacent parts can be evaluated in  $O(N)$  time. Furthermore, when the best merge is performed, the only impacted merges that need to be reevaluated for the next optimization step are the merges that share instances with the best merge. Since the data grid is sparse, the number of reevaluations of data grids is small on average. Sophisticated algorithmic data structures and algorithms are necessary to exploit these optimization principles and guarantee a time complexity of  $O(N \log N)$ .

The optimized version of the greedy heuristic is time efficient, but it may fall into a local optimum. First, the greedy heuristic may stop too soon and produce too many parts for each input variable. Second, the univariate partitions into intervals may be sub-optimal since the merge decisions of the greedy heuristic are never rejected. The post-optimization algorithms described in (Boullé, 2006) in the case of univariate discretization are applied alternatively to each input variable, for a frozen partition of the other input variable.

While post-optimizations may help to refine a good solution, the main heuristic may be unable to obtain such an initial good solution. This problem is tackled using the variable neighborhood search (VNS) meta-heuristic (Hansen and Mladenovic, 2001), which mainly benefits from multiple runs of the algorithms with different random initial solutions.

In the case of categorical variables, the combinatorial problem is still worse for large numbers of input categories  $V$ . The number of possible partitions of  $V$  categories is equal to the Bell number  $B(V) = \frac{1}{e} \sum_{k=1}^{\infty} \frac{k^V}{k!}$  which is far greater than the  $O(2^N)$  possible discretizations. Furthermore, the number of possible merges between adjacent parts is  $O(V^2)$  for categorical variables instead

of  $O(N)$  for numerical variables. Specific pre-processing and post-processing heuristics are necessary to efficiently handle the categorical input variables. Mainly, the number of groups of categories is bounded by  $O(\sqrt{N})$  in the algorithms, and the initial and final groupings are locally improved by exchanges of categories between groups. This allows to keep an  $O(N)$  memory complexity and bound the time complexity by  $O(N\sqrt{N} \log N)$  for categorical variables.

The optimization algorithms summarized above have been extensively evaluated in (Boullé, 2008), using a large variety of artificial datasets, where the true data distribution is known. Overall, the method is both resilient to noise and able to detect complex fine grained patterns. It is able to approximate any conditional data distribution as close as requested, provided that there are enough instances in the train data sample.

## 5 Experiments

This section evaluates the impact of the MODL bivariate partitioning method on supervised classification. The benefits for data preparation have been investigated in (Boullé, 2008). Overall, the bivariate partitioning method is very helpful in the data preparation step of data mining, with reliable ranking of pairs of variables, detection of constructive interactions or of redundancies in the representation space, and easily understandable visualizations of the joint conditional information carried out by each pair of input variables. In this section, we focus on the benefit for data modeling and evaluate the impact on classification accuracy of the data grid models as a preprocessing step for the naive Bayes classifier.

In order to evaluate the intrinsic performance of the MODL bivariate partitioning method, we introduce a new type of classifier called best bivariate (B2). This classifier first searches the best pair of input variables, which maximizes the probability that its partitioning model explains the classificatory output variable. In order to classify a test instance, the input cell related to the instance is retrieved from the trained data grid and the most frequent output class of this cell is used for prediction. In case where this cell was empty in the trained data grid, the most frequent output class on the whole train data sample is used for prediction. For sanity check, we also evaluate the best univariate classifier (B1), which proceeds in the same way on the basis of the MODL univariate analysis, and we present the results of the majority classifier (M) which always predict the most frequent output class and serves as a ground level reference.

In order to analyze the impact of the method on multivariate classifiers, we use the naive Bayes classifier (Langley et al., 1992), on the basis of the univariate preprocessing (NB1) and bivariate preprocessing (NB2). The bivariate preprocessing is basically exploited in the experiments, since each bivariate partitioning is simply managed as a constructed variable which expands the data representation space. In a classification problem with  $p$  input variables,

the NB1 classifier is based on  $p$  preprocessed input variables, while the NB2 classifier uses  $p(p-1)/2$  additional constructed variables corresponding to the preprocessed pairs of input variables. We also exploit the enhancements of the naive Bayes classifier described in (Boullé, 2007)<sup>2</sup>, which incorporates both variable selection and model averaging. This enhanced selective naive Bayes classifier (SNB) is applied using the univariate preprocessing (SNB1) and bivariate preprocessing (SNB2).

To summarize, the evaluated classifiers are:

- M: majority classifier,
- B1: best univariate classifier,
- B2: best bivariate classifier (based on the best preprocessed pair of input variables),
- NB1: naive Bayes classifier,
- NB2: naive Bayes classifier (based on bivariate preprocessing),
- SNB1: selective naive Bayes classifier,
- SNB2: selective naive Bayes classifier (based on bivariate preprocessing).

The experiments are performed on 30 datasets from the UCI repository (Blake and Merz, 1996) described in Table 1. They represent a large variety of domains, numbers of instances, numbers of variables, types of variables (numerical or categorical) and numbers of output classes. The test accuracy is estimated using a stratified ten fold cross-validation. In order to determine whether the performance are significantly different between the SNB2 method and the alternative methods, the t-statistics of the difference of the results is computed, at the 5% confidence level.

The results are summarized in Table 2 with the mean of the test accuracy on all the datasets. The number of significant differences for the SNB2 classifier is also reported, as well as the mean rank of each method. It is noteworthy that the classifier based on one single variable (B1) is as accurate as the best multivariate classifier evaluated in the benchmark (SNB2) in about one quarter of the datasets (no significant differences in 7 datasets out of 30). The classifier that selects the best pair of input variables (B2) obtains the best performance in about one third of datasets (10 datasets out of 30).

In order to analyse the results with deeper details, Fig. 5 presents the accuracy per dataset for the best univariate, best bivariate and naive Bayes classifiers, relatively to the accuracy of the majority classifier. The best bivariate classifier is always more accurate than the best univariate classifier, which confirms the capacity of the bivariate discretization method to efficiently select a predictive pair of variables. However, the best bivariate classifier is significantly dominated by the naive Bayes classifier, which exploits the whole set of variables.

Fig. 6 focuses on the naive Bayes multivariate classifier, and studies the impact of exploiting or not the pairs of variables (NB2 and NB1) and that of variable selection (SNB2 and SNB1). Using the pairs of variables enlarges

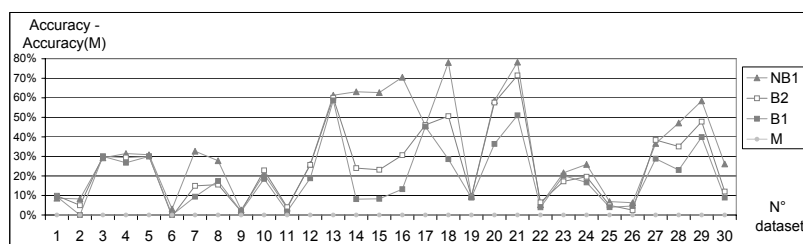
<sup>2</sup> Tool available as a shareware at <http://perso.rd.francetelecom.fr/boulle/>.

**Table 1** UCI Datasets

N°	Name	Instances	Numerical variables	Categorical variables	Classes	Majority accuracy
1	Abalone	4177	7	1	28	16.5
2	Adult	48842	7	8	2	76.1
3	Australian	690	6	8	2	55.5
4	Breast	699	10	0	2	65.5
5	Crx	690	6	9	2	55.5
6	German	1000	24	0	2	70.0
7	Glass	214	9	0	6	35.5
8	Heart	270	10	3	2	55.6
9	Hepatitis	155	6	13	2	79.4
10	HorseColic	368	7	20	2	63.0
11	Hypothyroid	3163	7	18	2	95.2
12	Ionosphere	351	34	0	2	64.1
13	Iris	150	4	0	3	33.3
14	LED	1000	7	0	10	11.4
15	LED17	10000	24	0	10	10.7
16	Letter	20000	16	0	26	04.1
17	Mushroom	8416	0	22	2	53.3
18	PenDigits	7494	16	0	10	10.4
19	Pima	768	8	0	2	65.1
20	Satimage	6435	36	0	6	23.8
21	Segmentation	2310	19	0	7	14.3
22	SickEuthyroid	3163	7	18	2	90.7
23	Sonar	208	60	0	2	53.4
24	Spam	4307	57	0	2	64.7
25	Thyroid	7200	21	0	3	92.6
26	TicTacToe	958	0	9	2	65.3
27	Vehicle	846	18	0	4	25.8
28	Waveform	5000	21	0	3	33.9
29	Wine	178	13	0	3	39.9
30	Yeast	1484	8	1	10	31.2

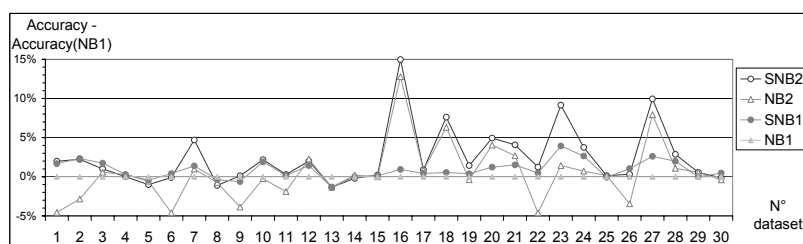
**Table 2** Mean of the test accuracy, number of significant differences (win/draw/loss) for the SNB2 classifier and mean rank of each classifier on 30 UCI datasets

	SNB2	NB2	SNB1	NB1	B2	B1	M
Mean	83.9%	81.9%	82.4%	81.4%	73.4%	67.6%	48.5%
W/D/L		15/15/0	12/18/0	14/16/0	20/10/0	23/7/0	
Mean rank	1.8	3.3	2.3	3.4	4.4	5.4	



**Fig. 5** Difference of accuracy between each evaluated classifier and the majority classifier (M), used as baseline. The evaluated classifiers are the best univariate (B1), best bivariate (B2) and naive Bayes (NB1) classifiers

the representation space, which potentially allows to detect new classificatory information. On the other hand, redundancies in the univariate representation are multiplied in the bivariate representation, which is detrimental to the naive Bayes assumption. Fig. 6 shows that the two effects are observed on the datasets of the experiments, with significant loss of accuracy for datasets 1, 2, 6, 9, 22, 26, and strong gain of accuracy for datasets 16, 18, 20, 21, 27. The variable selection method (Boullé, 2007) used in the SNB1 classifier confirms its beneficial impact on test accuracy, systematic but slight, compared to the NB1 classifier. When efficient variable selection is used together with the pairs of variables preprocessed using data grid models (SNB2), the gain in accuracy becomes both important, with an average improvement of 2.5% (15% for the Letter dataset), and highly significant, with 14 significant wins and 0 loss.



**Fig. 6** Difference of accuracy between each evaluated classifier and the naive Bayes classifier (NB1), used as baseline. The evaluated classifiers are the naive Bayes classifier exploiting all pairs of variables (NB2) and the selective naive Bayes classifiers based on univariate preprocessing (SNB1) or bivariate preprocessing (SNB2)

## 6 Conclusion

The bivariate discretization method introduced in this paper is based on a partitioning model of each input variables, into intervals for numerical variables

and into groups of categories for categorical variables. The Cartesian product of the univariate partitions, called a data grid, allows to quantify the conditional information relative to the output variable. The best data grid model is defined by maximizing a Bayesian model selection criterion and searched in the model space owing to efficient heuristics.

Our method is nonparametric both in the statistical and algorithmic sense : it does not rely on any statistical hypothesis for the data distribution (like Gaussianity for instance) and, as the criterion is regularized, there is no parameter to tune before optimizing it.

The benefit of data grid models for data preparation has been evaluated in (Boullé, 2008). The results demonstrate the ability of the method to detect constructive interactions or, on the opposite, redundancies between the input variables, and highlight the visualization and data understanding capacities of the data grids.

The impact of bivariate preprocessing on classification accuracy is evaluated in this paper through extensive experiments on 30 UCI datasets. The results show that the bivariate discretization method is able to select strongly predictive pairs of variables. However, the average impact on classification accuracy is not conclusive for the naive Bayes classifier when all the pairs of variables are exploited. The problem is that the potential benefit of additional classificatory information extracted from the pairs of variables is balanced by the detrimental effect of increased redundancies in the presentation space. When the naive Bayes classifier is equipped with an efficient variable selection method, the benefit of bivariate preprocessing becomes both systematic and important: the classification accuracy always increases, with significant differences in half of the cases.

## References

- Abramowitz, M. and I. Stegun (1970) Handbook of mathematical functions. New York: Dover Publications Inc.
- Bay, S. (2001) Multivariate discretization for set mining. *Machine Learning* 3(4), 491–512.
- Berger, J. (2006) The case of objective Bayesian analysis. *Bayesian Analysis* 1(3), 385–402.
- Bernardo, J. and A. Smith (2000) Bayesian theory. John Wiley & sons.
- Bertier, P. and J. Bouroche (1981) Analyse des données multidimensionnelles. Presses Universitaires de France.
- Blake, C. and C. Merz (1996) UCI repository of machine learning databases. <http://www.ics.uci.edu/mllearn/MLRepository.html>.
- Boullé, M. (2004) Khiops: a statistical discretization method of continuous attributes. *Machine Learning* 55(1), 53–69.
- Boullé, M. (2005) A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research* 6, 1431–1452.

- Boullé, M. (2006) MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning* 65(1), 131–165.
- Boullé, M. (2007) Compression-Based Averaging of Selective Naive Bayes Classifiers. *Journal of Machine Learning Research* 8, 1659–1685.
- Boullé, M. (2008) Bivariate data grid models for supervised learning. Technical Report NSM/R&D/TECH/EASY/TSI/4/MB, France Telecom R&D. <http://perso.rd.francetelecom.fr/boulle/publications/-BoulleNTTSI4MB08.pdf>.
- Breiman, L., J. Friedman, R. Olshen, and C. Stone (1984) *Classification and Regression Trees*. California: Wadsworth International.
- Carr, D., R. Littlefield, W. Nicholson, and J. Littlefield (1987) Scatterplot matrix techniques for large n. *Journal of the American Statistical Association* 82, 424–436.
- Chapman, P., J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth (2000) CRISP-DM 1.0 : step-by-step data mining guide.
- Cochran, W. (1954) Some methods for strengthening the common chi-squared tests. *Biometrics* 10(4), 417–451.
- Connor-Linton, J. (2003) Chi square tutorial. [http://www.georgetown.edu/-faculty/ballc/webtools/web\\_chi\\_tut.html](http://www.georgetown.edu/-faculty/ballc/webtools/web_chi_tut.html).
- Fayyad, U. and K. Irani (1992) On the handling of continuous-valued attributes in decision tree generation. *Machine Learning* 8, 87–102.
- Goldstein, M. (2006) Subjective Bayesian analysis: principles and practice. *Bayesian Analysis* 1(3), 403–420.
- Guyon, I. and A. Elisseeff (2003) An introduction to variable and feature selection. *Journal of Machine Learning Research* 3, 1157–1182.
- Guyon, I., S. Gunn, A. B. Hur, and G. Dror (2006) Design and analysis of the NIPS2003 challenge. In Guyon, I., Gunn, S., Nikravesh, M. and Zadeh, L., editors, *Feature Extraction: Foundations And Applications*, Chapter 9, pp 237–263. Springer, New York, USA.
- Hansen, P. and N. Mladenovic (2001) Variable neighborhood search: principles and applications. *European Journal of Operational Research* 130, 449–467.
- Holte, R. (1993) Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11, 63–90.
- Kass, G. (1980) An exploratory technique for investigating large quantities of categorical data. *Applied Statistics* 29(2), 119–127.
- Kerber, R. (1992) ChiMerge discretization of numeric attributes. In *Proceedings of the 10th International Conference on Artificial Intelligence*, pp 123–128. MIT Press, Cambridge, Massachusetts.
- Kohavi, R. and G. John (1997) Wrappers for feature selection. *Artificial Intelligence* 97(1-2), 273–324.
- Kohavi, R. and M. Sahami (1996) Error-based and entropy-based discretization of continuous features. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp 114–119. Menlo Park, CA: AAAI Press
- Kononenko, I., I. Bratko, and E. Roskar (1984) Experiments in automatic learning of medical diagnostic rules. Technical report, Ljubljana: Joseph



- 
- Stefan Institute, Faculty of Electrical Engineering and Computer Science.
- Kurgan, L. and J. Cios (2004) CAIM discretization algorithm. *IEEE Transactions on Knowledge and Data Engineering* 16(2), 145–153.
- Kwedlo, W. and M. Kretowski (1999) An evolutionary algorithm using multivariate discretization for decision rule induction. In *Principles of Data Mining and Knowledge Discovery*, LNCS 1704, pp 392–397. Springer Berlin, Heidelberg.
- Langley, P., W. Iba, and K. Thompson (1992) An analysis of Bayesian classifiers. In *10th National Conference on Artificial Intelligence*, pp 223–228. San Jose, CA: AAAI Press.
- Maass, W. (1994) Efficient agnostic pac-learning with simple hypothesis. In *COLT '94: Proceedings of the seventh annual conference on Computational learning theory*, pp 67–75. ACM Press.
- Nadif, M. and G. Govaert (2005) Block clustering of contingency table and mixture model. In *Advances in Intelligent Data Analysis VI*, LNCS 3646, pp 249–259. Springer Berlin, Heidelberg.
- Olszak, M. and G. Ritschard (1995) The behaviour of nominal and ordinal partial association measures. *The Statistician* 44(2), 195–212.
- Pyle, D. (1999) *Data preparation for data mining*. Morgan Kaufmann Publishers, Inc. San Francisco, USA.
- Quinlan, J. (1986) Induction of decision trees. *Machine Learning* 1, 81–106.
- Quinlan, J. (1993) *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Rissanen, J. (1978) Modeling by shortest data description. *Automatica* 14, 465–471.
- Ritschard, G. and N. Nicoloyannis (2000) Aggregation and association in cross tables. In *PKDD '00: Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pp 593–598. Springer-Verlag.
- Robert, C. (1997) *The Bayesian choice: A decision-theoretic motivation*. New York: Springer-Verlag.
- Saporta, G. (1990) *Probabilités analyse des données et statistique*. TECHNIP, Paris.
- Shannon, C. (1948) A mathematical theory of communication. Technical Report 27, Bell systems technical journal.
- Steck, H. and T. Jaakkola (2004) Predictive discretization during model selection. *Pattern Recognition* LNCS 3175, 1–8.
- Weaver, W. and C. Shannon (1949) *The mathematical theory of communication*. Urbana, Illinois: University of Illinois Press.
- Zighed, D., S. Rabaseda, and R. Rakotomalala (1998) Fusinter: a method for discretization of continuous attributes for supervised learning. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6(33), 307–326.
- Zighed, D. and R. Rakotomalala (2000) *Graphes d'induction*. France: Hermes.
- Zighed, D., G. Ritschard, W. Erray, and V. Scuturici (2005) Decision trees with optimal joint partitioning. *International Journal of Intelligent System* 20(7), 693–718.