Multivariate Data Grid Models for Supervised and Unsupervised Learning

Note technique

Référence : **Autre référence :** **Version :** **Date d'édition :**	NSM/R&D/TECH/EASY/TSI/5/MB 1.0 15/02/2008	*Vérifié par : Fabrice Clérot* *Affiliation : TECH/EASY* Le : 15/02/2008
Auteurs :	Boullé Marc TECH/EASY	*Approuvé par : Patrice Soyer* *Affiliation : TECH/EASY* Le : 15/02/2008

Résumé :
This paper introduces a new method to automatically, rapidly and reliably evaluate the conditional probability distribution of any subset of variables in supervised learning. It is based on a partitioning of each input variable, into intervals in the numerical case and into groups of values in the categorical case. The cross-product of the univariate partitions forms a multivariate partition of the input representation space into a set of cells. This multivariate partition, called data grid, is a piecewise constant nonparametric estimator of the probability distribution. The best data grid is searched using a Bayesian model selection approach and an efficient combinatorial algorithm.

We have exploited data grid models for data preparation and modelling and study their benefits and limits using artificial and real datasets. In the Agnostic Learning vs. Prior Knowledge Challenge, our method achieved the best performance on two of the datasets. These experiments demonstrate the interest of using data grid models in machine learning tasks, for conditional density estimation, data preparation, classification and rule based explanation.

This paper is divided into two chapters. Chapter 1 introduces the method in the context of supervised learning, and Chapter 2 extends it to unsupervised learning and coclustering.
Mots clés : Data Mining, Data Analysis, Data Preparation, Discretization, Value Grouping, Bayesianism, Model Selection, Supervised Learning, Unsupervised Learning, Coclustering
Thème : 7800 - Intelligence artificielle - IA

Chapter 1

Data Grid Models for Supervised Learning

Multivariate Data Grid Models for Supervised Learning

Marc Boullé

France Télécom R&D Lannion,
marc.boulle@orange-ftgroup.com

Abstract. This paper introduces a new method to automatically, rapidly and reliably evaluate the class conditional information of any subset of variables in supervised learning. It is based on a partitioning of each input variable, into intervals in the numerical case and into groups of values in the categorical case. The cross-product of the univariate partitions forms a multivariate partition of the input representation space into a set of cells. This multivariate partition, called data grid, is a piecewise constant nonparametric estimator of the class conditional probability. The best data grid is searched using a Bayesian model selection approach and an efficient combinatorial algorithm.
We present two classification techniques, exploiting the maximum a posteriori data grid or an ensemble of data grids, and report results in the Agnostic Learning vs. Prior Knowledge Challenge, where our method achieved the best performance on one of the datasets. These experiments demonstrate the interest of using data grid models in machine learning tasks, for conditional density estimation, data preparation, classification and rule based explanation.

1 Introduction

Univariate partitioning methods have been studied extensively in the past, mainly in the context of decision trees [Kas80,BFOS84,Qui93,ZR00]. Supervised discretization methods split the numerical domain into a set of intervals and supervised value grouping methods partition the input values into groups. Fine grained partitions allow an accurate discrimination of the output values, whereas coarse grain partitions tend to be more reliable. When the size of the partition is a free parameter, the trade-off between information and reliability is an issue. In the MODL approach, supervised discretization [Bou06] (or value grouping [Bou05]) is considered as a non-parametric model of dependence between the input and output variables. The best partition is found using a Bayesian model selection approach.

In this paper, we describe an extension of the MODL approach to the bivariate case for pairs of input variables [Bou07b], and introduce its generalization to any subset of variables of any types, numerical, categorical or mixed types. Each input variable is partitioned, into intervals in the numerical case and into groups of values in the categorical case. The cross-product of the univariate partitions

forms a multi-dimensional data grid. The correlation between the cells of this data grid and the output values allows to quantify the joint predictive information. The trade-off between information and reliability is established using a Bayesian model selection approach.

Sophisticated algorithms are necessary to explore the search space of data grid. They have to strike a balance between the quality of the optimization and the computation time. Several optimization heuristics, including greedy search, meta-heuristic and post-optimization are introduced to efficiently search the best possible data grid.

The paper is organized as follows. Section 2 summarizes the MODL method in the univariate discretization and value grouping case. Section 3 extends the approach to the multivariatiate case, by introducing data grid models, and Section 4 presents the optimization algorithms. Section 5 evaluates the data grid models on artificial datasets and studies their limitations. Section 6 reports experiments performed on the agnostic learning vs. prior knowledge challenge datasets [GSDC07] and analyzes their interest for classification and explanation. Finally, Section 7 gives a summary and discusses future work.

2 The MODL Discretization and Value Grouping Methods

This section summarizes the MODL approach in the univariate case, detailed in [Bou06] for supervised discretization, and in [Bou05] for supervised value grouping.

2.1 Discretization

The objective of supervised discretization is to induce a list of intervals which partitions the numerical domain of a continuous input variable, while keeping the information relative to the output variable. A compromise must be found between information quality (homogeneous intervals in regard to the output variable) and statistical quality (sufficient sample size in every interval to ensure generalization).

In the MODL approach, the discretization is turned into a model selection problem. First, a space of discretization models is defined. The parameters of a specific discretization are the number of intervals, the bounds of the intervals and the frequencies of the output values in each interval. Then, 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 bounds of the intervals and finally the frequencies of the output values. The choice is uniform at each stage of the hierarchy. Finally, we assume that the multinomial distributions of the output values 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 $p(Model|Data)$ of the model given the data. Using the Bayes

rule and since the probability $p(Data)$ is constant under varying the model, this is equivalent to maximizing $p(Model)p(Data|Model)$.

Let N be the number of instances, J the number of output values, I the number of input intervals. N_i denotes the number of instances in the interval i, and N_{ij} the number of instances of output value j in the interval i. In the context of supervised classification, the number of instances N and the number of classes J are supposed to be known. A discretization model M is then defined by the parameter set $\left\{I, \{N_i\}_{1\leq i\leq I}, \{N_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}\right\}$.

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. Taking the negative log of the probabilities, this provides the evaluation criterion given in Formula 1.

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

The first term of the criterion stands for the choice of the number of intervals and the second term for the choice of the bounds of the intervals. The third term corresponds to the parameters of the multinomial distribution of the output values in each interval and the last term represents the conditional likelihood of the data given the model, owing to a multionomial term. Therefore "complex" models with large numbers of intervals are penalized.

Once the optimality of the evaluation criterion is established, the problem is to design a search algorithm in order to find a discretization model that minimizes the criterion. In [Bou06], 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 with 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.

2.2 Value Grouping

Categorical variables are analyzed in a way similar to that of numerical variables, owing to a partitioning model of the input values. 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 values and any partition into groups of values is possible. The problem is to improve the reliability of the estimation of the class conditional probabilities owing to a reduced number of groups of values, while keeping the groups as informative as possible. Producing a good grouping is harder with

large numbers of input values since the risk of overfitting the data increases. In the extreme situation where the number of values 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.

Let N be the number of instances, V the number of input values, J the number of output values and I the number of input groups. N_i denotes the number of instances in the group i, and N_{ij} the number of instances of output value j in the group i. The Bayesian model selection approach is applied like in the discretization case and provides the evaluation criterion 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 values, into groups of values in Formula 2 and into intervals in Formula 1. The two last terms are the same in both formula.

$$\log V + \log B\left(V, I\right) + \sum_{i=1}^{I} \log \binom{N_i + J - 1}{J - 1} + \sum_{i=1}^{I} \log \frac{N_i!}{N_{i1}! N_{i2}! \ldots N_{iJ}!} \qquad (2)$$

$B(V, I)$ is the number of divisions of V values 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 $B(V, I) = \sum_{i=1}^{I} S(V, i)$, where $S(V, i)$ is the Stirling number of the second kind [AS70], which stands for the number of ways of partitioning a set of V elements into i nonempty sets.

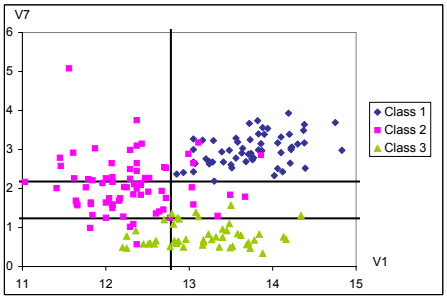
In [Bou05], a standard greedy bottom-up heuristic is proposed to find a good grouping of the input values. Several pre-optimization and post-optimization steps are incorporated, in order to both ensure an algorithmic time complexity of $O(JN \log(N))$ and obtain accurate value groupings.

3 Data Grids Models for any Subset of Variables

In this section, we describe the extension of the MODL approach to pairs of variables introduced in [Bou07b] and generalize it to any subset of variables, in the numerical, categorical and mixed type case. We first introduce the approach using an illustrative example for the case of bivariate discretization, then summarizes the principles of the extension in the general case, and finally present the evaluation criterion of such models.

3.1 Interest of the joint partitioning of two input variables

Figure 1 draws the multiple scatter plot (per class value) of the input variables V1 and V7 of the wine dataset [BM96]. This diagram allows to visualize the conditional probability of the output values 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



]2.18, +∞[	(0, 23, 0)	(59, 0, 4)
]1.235, 2.18]	(0, 35, 0)	(0, 5, 6)
]-∞, 1.235]	(0, 4, 11)	(0, 0, 31)
V7xV1	]-∞, 12.78]	]12.78, +∞[

Fig. 1. Multiple scatterplot (per class value) 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 class frequencies per data grid cell is reported in the right table

Class 2 for input values greater than 2. Taken jointly, the two input variables allow a better separation of the class values.

Extending the univariate case, we partition the dataset on the cross-product of the input variables to quantify the relationship between the input and output variables. Each input variable is partitioned into a set of *parts* (intervals in the numerical case). The cross-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 evaluated using the distribution of the output values in each cell of the data grid. It is noteworthy that the considered partitions can be factorized on the input variables.

For instance in Figure 1, the V1 variable is discretized into 2 intervals (one bound 12.78) and the V7 variable into 3 intervals (two bounds 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 values can be estimated by counting. For example, 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.

Coarse grain data grids tend to be reliable, whereas fine grain data grids allow a better separation of the output values. In our example, the MODL optimal data grid is drawn on the multiple scatter plot on Figure 1.

3.2 Principles of the Extension to Data Grid Models

The MODL approach has been studied in the case of univariate supervised partitioning for numerical variables [Bou06] and categorical variables [Bou05]. The extension to the multivariate case applies the same principles as those described in section 3.1. Each input variable is partitioned, into intervals in the numerical case and into groups of values in the categorical case. Taking the cross-product

of the univariate partitions, we obtain a data grid of input cells, the content of which allows to characterize the distribution of the output values.

The space of multivariate data grid models is very large and prone to overfitting. A Bayesian model selection approach is employed to find the best data grid model given the data. The parameters of the data grid models are precisely defined, and a prior is proposed that exploits the hierarchy of the parameters, is uniform at each stage of the hierarchy, and assumes the independence of the output distribution within each cell. We then obtain an analytic formula that evaluates the posterior probability of each data grid model, and exploit the algorithms described in section 4 to efficiently search the space of data grid models.

3.3 Evaluation Criterion for Supervised Data Grids

We present in Definition 1 a family of multivariate partitioning models and select the best model owing to a Bayesian model selection approach. Compared to the bivariate case, we introduce a new level in the hierarchy of the model parameters, related to variable selection. Indeed, a multivariate data grid model implicitly handles variables selection, where the selected variables which bring predictive information are partitioned in at least two parts. The other variables, the partition of which consists of one single part, can be considered as irrelevant and discarded. We use this variable selection feature explicitly in Definition 1.

Definition 1. *A data grid classification model is defined by a subset of selected input variables, for each selected variable by a univariate partition, into intervals in the numerical case and into groups of values in the categorical case, and by a multinomial distribution of the output values in each cell of the data grid resulting from the cross-product of the univariate partitions.*

Notation.

- Y: output variable,
- $X_1, \ldots, X_K$: input variables,
- N: number of instances,
- J: number of output values,
- K: number of input variables,
- $\mathbb{K}$: set of input variables ($|\mathbb{K}| = K$),
- $\mathbb{K}_n$: subset of numerical input variables,
- $\mathbb{K}_c$: subset of categorical input variables,
- $V_k, k \in \mathbb{K}_c$: number of values of the categorical input variable X_k,
- K_s: number of selected input variables,
- $\mathbb{K}_s$: subset of selected input variables ($|\mathbb{K}_s| = K_s$),
- I_k: number of parts (intervals or groups of values) in the univariate partition of input variable X_k,
- $N_{i_1 i_2 \ldots i_K}$: number of instances in the input data cell $(i_1, i_2, \ldots, i_K)$,
- $N_{i_1 i_2 \ldots i_K j}$: number of instances of output value j in the input data cell $(i_1, i_2, \ldots, i_K)$.

Like in the bivariate case presented in Section 3.1, any input information is used to define the family of the model. For example, the numbers of instances per cell $N_{i_1 i_2 \ldots i_K}$ do not belong to the parameters of the data grid models: they are derived from the definition of the univariate partitions of the selected input variables and from the dataset. These numbers of instances allow to constrain the specification of the multinomial distribution of the output values in each input cell.

We now introduce in Definition 2 a prior distribution on the parameters of the data grid models. Applying the MODL approach, this prior exploits the hierarchy of the parameters and is uniform at each stage of this hierarchy. For the variable selection parameters, we reuse the prior introduced by [Bou07a] in the case of the selective naive Bayes classifier. For the specification of each univariate partition, we reuse the prior presented by [Bou06] for supervised discretization of numerical variables and by [Bou05] for supervised value grouping of categorical variables.

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

- *the number of selected input variables is uniformly distributed between* 1 *and* K*,*
- *for a given number* K_S *of selected input variables, the subsets of* K_S *variables are uniformly distributed (with replacement),*
- *the numbers of input parts, are independent from each other, and uniformly distributed between* 1 *and* N *for numerical variables, between* 1 *and* V_k *for categorical variables,*
- *for each numerical input variable and for a given number of intervals, every partition into intervals is equiprobable,*
- *for each categorical input variable and for a given number of groups, every partition into groups is equiprobable,*
- *for each cell of the input data grid, every distribution of the output values is equiprobable,*
- *the distributions of the output values in each cell are independent from each other.*

We apply the Bayesian model selection approach and obtain the evaluation criterion of a data grid model M in Formula 3.

$$\begin{aligned}
&\log(K+1) + \log \binom{K+K_s-1}{K_s} \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_n} \left(\log N + \log \binom{N+I_k-1}{I_k-1} \right) + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_c} (\log V_k + \log B(V_k, I_k)) \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} \log \binom{N_{i_1 i_2 \ldots i_K} + J - 1}{J-1} \\
&+ \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_K=1}^{I_K} \left(\log N_{i_1 i_2 \ldots i_K}! - \sum_{j=1}^{J} \log N_{i_1 i_2 \ldots i_K j}! \right)
\end{aligned} \tag{3}$$

The first line in Formula 3 corresponds to the prior for variable selection. As in the univariate case, the second line is related to the prior probability of the discretization parameters (like in Formula 1) for the selected numerical input variables and to that of the value grouping parameters (like in Formula 2) for the selected categorical input variables. The binomial terms in the third line represent the choice of the multinomial distribution of the output values in each cell of the input data grid. The multinomial terms in the last line represent the conditional likelihood of the output values given the data grid model.

4 Optimization Algorithm for Multivariate Data Grids

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. In this section, we describe such algorithms. 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 $O(KN)$ memory complexity and $O(N\sqrt{N} \log N \max(K, \log N))$ time complexity.

4.1 Greedy Bottom-Up Heuristic

Let us first focus on the case of numerical input variables. The optimization of a data grid is a combinatorial problem. For each input variable X_k, there are 2^N possible univariate discretizations, which represents $\left(2^N\right)^K$ possible multivariate discretizations. An exhaustive search over 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 partitions, based on single value parts, intervals or groups. It evaluates all the merges between adjacent parts, and performs the best merge if the evaluation criterion decreases after the merge. The process is reiterated until no further merge can decrease the criterion.

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}
$M^* \leftarrow M$
while improved solution **do**
  {Evaluate all the merges between adjacent parts}
  $c^* \leftarrow \infty, m^* \leftarrow \emptyset$
  for all Variable $X_k \in \mathbb{K}$ **do**
    for all Merge m between two adjacent parts of variable X_k **do**
      $M' \leftarrow M^* + m$ {Evaluate merge m on data grid M^*}
      if $c(M') < c^*$ **then**
        $c^* \leftarrow c(M'), m^* \leftarrow m$
      end if
    end for
  end for
  {Perform best merge}
  if $c^* < c(M^*)$ **then**
    $M^* \leftarrow M^* + m^*$
  end if
end while

Each evaluation of a data grid requires $\mathrm{O}(N^K)$ time, since the initial data grid model M_{Max} contains N^K cells. Each step of the algorithm relies on $\mathrm{O}(N)$ evaluations of interval merges times the number K of variables. There are at most $\mathrm{O}(KN)$ steps, since the data grid becomes equal to the null model $M_\emptyset$ (one single cell) once all the possible merges have been performed. Overall, the time complexity of the algorithm is $\mathrm{O}(K^2N^2N^K)$ using a straightforward implementation of the algorithm. However, the GBUM algorithm can be optimized in $\mathrm{O}(K^2N \log N)$ time, as shown in next section and demonstrated in [Bou08] in the bivariate case.

4.2 Optimized Implementation of the Greedy Heuristic

The optimized algorithm mainly exploits the sparseness of the data and the additivity of the evaluation criterion. Although a data grid may contain $\mathrm{O}(N^K)$ cells, at most N cells are non empty. Thus, each evaluation of a data grid can be performed in $\mathrm{O}(N)$ owing to a specific algorithmic data structure.

The additivity of the evaluation criterion means that the criterion can be decomposed according to Definition 3 on the hierarchy of the components of the data grid: grid size, variables, parts and cells.

Definition 3. *An evaluation criterion $c(M)$ of a data grid model M is additive if it can be decomposed as a sum of terms according to*

$$c(M) = c^{(G)}\left(\mathcal{I}\right) + \sum_{k=1}^{K} c^{(V)}\left(X_k, I_k\right) + \sum_{k=1}^{K}\sum_{i_k=1}^{I_k} c^{(P)}\left(P_{i_k}^{(k)}\right) + \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_k=1}^{I_K} c^{(C)}\left(C_{i_1 i_2 \ldots i_K}\right)$$

where

- *the grid criterion $c^{(G)}\left(\mathcal{I}\right)$ relies only on the sizes $\mathcal{I} = \{I_1, I_2, \ldots, I_K\}$ of the univariate partitions of the data grid,*
- *the variable criterion $c^{(V)}\left(X_k, I_k\right)$ relies only on features of the input variable X_k and on the number of parts I_k of its partition,*
- *the part criterion $c^{(P)}\left(P_{i_k}^{(k)}\right)$ for each part $P_{i_k}^{(k)}$ of the univariate partition of the input variable X_k relies only on features of the part,*
- *the cell criterion $c^{(C)}\left(C_{i_1 i_2 \ldots i_K}\right)$ for each cell $C_{i_1 i_2 \ldots i_K}$ of the data grid relies only on features of the cell, and is null for empty cells.*

One can easily check that the evaluation criterion introduced in Formula 3 is an additive criterion. Using this additivity property, all the merges between adjacent parts can be evaluated in $\mathrm{O}(N)$ 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 partial reevaluations of the criterion is limited by the number of instances, not by the number of cells in the data grids. Sophisticated algorithmic data structures and algorithms, detailed in [Bou08], are necessary to exploit these optimization principles and guarantee a time complexity of $\mathrm{O}(K^2 N \log N)$.

4.3 Post-Optimization

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 boundaries of the intervals may be sub-optimal since the merge decisions of the greedy heuristic are never rejected. We propose to reuse the post-optimization algorithms described in [Bou06] in the case of univariate discretization.

In a first stage called *exhaustive merge*, the greedy heuristic merge steps are performed without stopping condition until the data grid consists of one single cell. The best encountered data grid is then memorized. This stage allows escaping local minima with several successive merges and needs $\mathrm{O}(K^2 N \log N)$ time.

In a second stage called *greedy post-optimization*, a hill-climbing search is performed in the neighborhood of the best data grid. This search alternates

the optimization on each input variable. For each given input X_k, we freeze the partition of all the other input variables and optimize the partition of X_k. Since a multivariate additive criterion turns out to be an univariate additive criterion once all except one univariate partitions are frozen, we reuse the post-optimization algorithms described in [Bou06] for univariate discretizations. This process is repeated for all variables until no further improvement can be obtained. This algorithm converges very quickly in practice and requires only a few steps.

We summarize the post-optimization of data grids in Algorithm 2.

Algorithm 2 Post-optimization of a Data Grid

Require: M {Initial data grid solution}
Ensure: $M^*; c(M^*) \leq c(M)$ {Final solution with improved cost}
1: $M^* \leftarrow$ call *exhaustive merge* (M)
2: **while** improved solution **do**
3: **for all** Variable $X_k \in \mathbb{K}$ **do**
4: Freeze the univariate partition of all the variables except X_k
5: $M^* \leftarrow$ call *univariate post-optimization* (M^*) for variable X_k
6: **end for**
7: **end while**

4.4 Meta-Heuristic

Since the GBUM algorithm is time efficient, it is then natural to apply it repeatedly in order to better explore the search space. This is done according to the *variable neighborhood search* (VNS) meta-heuristic introduced by [HM01], which consists in applying the primary heuristic to a random neighbor of the solution. If the new solution is not better, a bigger neighborhood is considered. Otherwise, the algorithm restarts with the new best solution and a minimal size neighborhood. The process is controlled by the maximum length of the series of growing neighborhoods to explore.

For the primary heuristic, we choose the greedy bottom-up heuristic followed by the post-optimization heuristic. In order to "purify" the randomly generated solutions given to the primary heuristic, we also incorporate a pre-optimization heuristic, that exploits the same principle as the post-optimization heuristic.

This meta-heuristic is described in Algorithm 3. According to the level of the neighborhood size l, a new solution M' is generated close to the current best solution. We define the structure of neighborhood by exploiting at most $K_{Max} = \log_2 N$ new variables. For each exploited variable, a random discretization is obtained with the choice of random interval bounds without replacement, with at most $I_{Max} = N^{\frac{1}{K_{Max}}}$ intervals. This heuristic choice for the maximum neighborhood size results from the analysis of Formula 3. In the case of two equidistributed output values, if we have K_{Max} selected variables with I_{Max} intervals per variable and exactly one instance per input cell, the cost of the

model is slightly worse than that of the null model with no selected variable. This means that too sparse data grids are not likely to be informative according to Formula 3.

The VNS meta-heuristic only requires the number of sizes of neighborhood as a parameter. This can easily be turned into an anytime optimization algorithm, by calling iteratively the VNS algorithm with parameters of increasing size and stopping the optimization only when the allocated time is elapsed. In this paper, all the experiments are performed by calling the VNS algorithm with successive values of $1, 2, 4, \ldots, 2^T$ for the parameter $MaxLevel$.

In order to improve the initial solution, we choose to first optimize the univariate partition of each variable and to build the initial solution from a cross-product of the univariate partitions. Although this cannot help in case of strictly bivariate patterns (such as XOR for example), this might be helpful otherwise.

Algorithm 3 VNS meta-heuristic for data grid optimization

Require: M {Initial data grid solution}
Require: $MaxLevel$ {Optimization level}
Ensure: $M^*, c(M^* \leq c(M)$ {Final solution with improved cost}
$Level \leftarrow 1$
while $Level \leq MaxLevel$ **do**
  {Generate a random solution in the neighborhood of M^*}
  $M'' \leftarrow$ random solution with $K_s = \frac{Level}{MaxLevel} \log_2 N$ new selected variables and $\frac{Level}{MaxLevel} N^{\frac{1}{K_s}}$ new intervals per selected variable
  $M' \leftarrow M^* \cup M''$
  {Optimize and evaluate the new solution}
  $M' \leftarrow$ call *Pre-Optimization*(M')
  $M' \leftarrow$ call *Greedy Bottom-Up Merge*(M')
  $M' \leftarrow$ call *Post-Optimization*(M')
  if $c(M') < c(M^*)$ **then**
    $M^* \leftarrow M'$
    $Level \leftarrow 1$
  else
    $Level \leftarrow Level + 1$
  end if
end while

4.5 The case of Categorical Variables

In the case of categorical variables, the combinatorial problem is still worse for large numbers of values V. The number of possible partitions of the values 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 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 values is bounded by $O(\sqrt{N})$ in the algorithms, and the initial and final groupings are locally improved by exchange of values between groups. This allows to keep an $O(N)$ memory complexity per variable and bound the time complexity by $O(N\sqrt{N}\log N)$ per categorical variable, with an overall time complexity of $O(K^2N\sqrt{N}\log N)$ for the complete greedy heuristic.

4.6 Summary of the Optimization Algorithms

The optimization of multivariate data grid models can be summarized as an extension of the univariate discretization and value grouping algorithms to the multivariate case.

The main heuristic is a greedy bottom-up heuristic, which starts from an initial fine grain data grid and iteratively performs the best merges between two adjacent parts of any input variable. Post-optimizations are carried out to improved the best data grid, by exploiting a local neighborhood of the solution. The main optimization heuristic (surrounded by pre-optimization and post-optimization steps) is run from several initial solutions, coming from the exploration of a global neighborhood of the best solution owing to a meta-heuristic.

These algorithms are efficiently implemented, on the basis of two main properties of the problem to optimize: the additivity of the criterion, which consists of a sum of independent terms related to the dimension of the data grid, the variables, the parts and the cells, and the sparseness of the data grids, which contain $O(N^K)$ cells for at most N non empty cells. Furthermore, in the meta-heuristic, we restrict to data grids with at most $K_{Max} = \log_2 N$ variables, which reduces the time complexity of the main greedy heuristic.

Sophisticated algorithms, detailed in [Bou08], are necessary to make the most of these problem properties and to reach the following algorithmic performance:

- $O(KN)$ memory complexity for K variables and N instances,
- $O(KN\log N\max(K,\log N))$ if all the input variables are numerical,
- $O(KN\sqrt{N}\log N\max(K,\log N))$ in the general case of numerical variables and categorical variables having large number of input values ($V \geq \sqrt{N}$).

5 Experiments on Artificial Datasets

In the bivariate case, the data grid models have been intensely experimented on artificial and real datasets in [Bou07b]. In this section, we evaluate the multivariate data grid models on artificial datasets, where the true data distribution is known. Two patterns are considered: noise and multivariate XOR.

5.1 The Noise Pattern

The purpose of the noise pattern experiment is to evaluate the noise resistance of the method, under variation of the sample size and number of input variables. The noise pattern consists of an output variable independent from the

input variables. The expected data grid contains one single cell, meaning that the output distribution is independent from the input variables. The output variable is equidistributed on two values. The experiment is performed on a set of sample sizes ranging from 2 to 1000 instances, for 1, 2 and 10 numerical input variables uniformly distributed on the [0, 1] numerical domain. The evaluated criterion is the number of cells in the data grid. In order to obtain reliable results, the experiment is performed one million times on randomly generated train datasets for each sample size and number of input variables. In order to study the impact of variable selection in the prior distribution of the models (terms $\log(K+1) + \log\binom{K+K_s-1}{K_s}$ in Formula 3), we repeat the experiment with and without the variable selection terms. Figure 2 presents the mean cell number for each sample size and number of input variable, with and without the prior for variable selection.

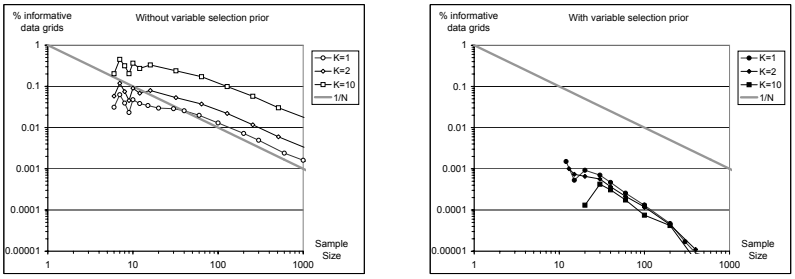


Fig. 2. Percentage of informative data grids having more than one cell, for 1, 2 and 10 numerical input variable independent from the target variable, with and without prior for variable selection.

The results demonstrate the robustness of the approach: very few data grids are wrongly detected as informative, and the percentage of false detection rapidly decreases with the sample size. However, without prior for variable selection, the percentage of false detection grows almost linearly with the number of input variables. This makes sense since a set of K variables can be detected as an informative multivariate data grid if at most one of the K variables is detected as an informative univariate discretization.

When the prior for variable selection is accounted for, the percentage of wrongly informative models falls down by a one hundred factor, and the rates of false detection are rapidly consistent for the different numbers of input variables. The selection prior significantly strengthens the robustness of the method and makes it almost independent from the number of variables in the representation space.

5.2 The Multivariate XOR Pattern

The purpose of the XOR pattern experiment is to evaluate the capacity of the method to detect complex correlations between the input variables. The pattern consists of an output variable which depends upon the input variables according to a XOR schema, as illustrated in Figure 3. All the input variables are uniformly distributed on the [0, 1] numerical domain. For each input variable, we compute a Boolean index according to whether the input value is below or beyond 0.5, and the output value is assigned a Boolean value related to the parity of the sum of the input indexes, which corresponds to a XOR pattern.

We first present a theoretical threshold of detection for the XOR pattern, then illustrate the behavior of the algorithms for this pattern, and finally report experimental results on this complex pattern detection problem.

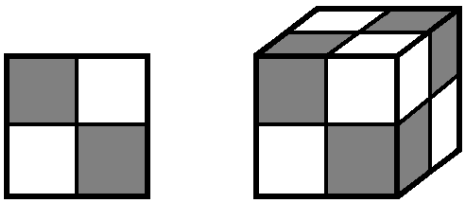


Fig. 3. Multivariate XOR pattern in dimension 2 and 3.

Theoretical Detection Threshold Let us consider K input variables, K_s of which represent a multivariate XOR pattern related to the output variable. The expected multivariate discretization for this pattern consists of a data grid model M_G with K_s selected input variables, each of which is discretized into two intervals. The data grid model M_G contains $G = 2^{K_s}$ cells. In order to obtain a closed formula, let us assume that these cells contain the same number $N_G = N/G$ of instances. Let us evaluate the null model $M_\emptyset$, reduced to one single cell, and the expected XOR data grid model M_G. According to Formula 3, we get

$$c(M_\emptyset) = \log(K+1) + \log(N+1) + \log \frac{N!}{N_1!N_2!}, \tag{4}$$

$$c(M_G) = \log(K+1) + \log \binom{K+K_s-1}{K_s-1} + K_s \log N + K_s \log(N+1) + G \log(N_G+1). \tag{5}$$

For $N_G = 1$, the null model is always preferred: one instance per cell is not enough to detect the multivariate pattern.

For small values of K and for $K_s = K$, we perform a numerical simulation to compute the minimum cell frequency N_G such that the cost $c(M_G)$ of the multivariate XOR model is lower than that of the null model. The results, reported in Figure 4, indicate that at least ten instances per cell, representing overall forty instances, are necessary to detect the bi-dimensional XOR pattern. This cell frequency threshold decreases with the number of input variables, and falls down to two instances per cell when the number of input variables is beyond ten. Let us notice that in spite of a very small cell frequency threshold, the whole dataset frequency threshold still grows exponentially with the the number of variables.

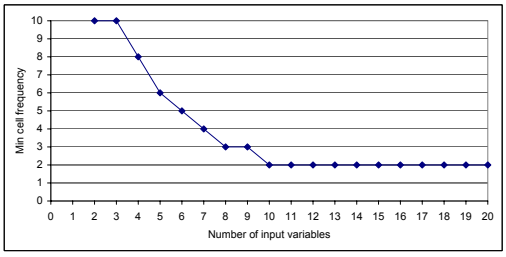


Fig. 4. Min cell frequency necessary to detect a multivariate XOR pattern owing to a data grid model. For example, for a 5-dimensional XOR, 6 instances per cell, or $192 = 2^5 * 6$ instances in the sample, allow to detect the pattern using a data grid of 32 cells.

We now extend these simulation results in the asymptotic case, assuming that each cell contains exactly $N_G = N/G$ instances. From Equations 4 and 5, we get

$$c(M_G) = c(M_\emptyset) + \log \binom{K + K_s - 1}{K_s - 1} + (2K_s - 1) \log N - N(\log 2 - \frac{1}{N_G} \log(N_G + 1)) + O(\log N).$$

This implies that for $N_G \geq 2$, the multivariate XOR model has an asymptotically lower cost than that of the null model, even when the total number K of input variables exceeds the number K_s of informative input variables.

Overall, about 2^{K+1} instances are sufficient to detect K-dimensional informative patterns, which correspond to 2 instances per cell. Since this is close from the theoretical detection threshold, this means that for a dataset consisting of N instances, it might be difficult to detect patterns exploiting more than $\log_2 N$ informative dimensions.

Empirical Analysis of the Algorithms Let us first analyze the behavior of the greedy bottom-up heuristic presented in Section 4.1. This heuristic starts with the maximum data grid, which contains $\mathrm{O}(N^K)$ cells for at most N non-empty cells. During the whole merge process, $\mathrm{O}(KN)$ merges are necessary to

transform the maximum data grid with N^K elementary cells into the null data grid with one single cell. During the first $(K-1)N$ merges, most of the merges between adjacent intervals concern merges between two empty adjacent cells or merges between one non-empty cell and one empty cell. When the data grid is too sparse, most interval merges do not involve "collisions" between non-empty cells. According to Formula 3, the only cell merges that have an impact on the likelihood of the data grid model are the "colliding" cell merges. This means that at the beginning of the greedy bottom-heuristic, the earlier part merges are guided only by the prior distribution of the models, not by their likelihood. These "blind" merges are thus likely to destroy potentially interesting patterns.

To illustrate this behavior, we perform an experiment with the basic greedy heuristic described in Algorithm 1 on a bi-dimensional XOR pattern. According to Formulas 4 and 5, about 40 instances are sufficient to detect the pattern. However, the greedy bottom-heuristic fails to discover the XOR pattern when the number of instance is below 1000.

The algorithms presented in Section 4 enhance the basic greedy heuristic using a random initialization, a pre-processing step, the greedy bottom-up merge heuristic and a post-processing step, as illustrated in Figure 5. The random initialization produces a dense enough data grid with at least one instance per cell on average. This is achieved by selecting at most $K_s = \log_2 N$ input variables and N^{1/K_s} parts per variable. The purpose of the pre-processing step is to "purify" the initial solution, since a random solution is likely to be blind to informative patterns. This pre-processing consists in moving the boundaries of the initial data grid, in order to get "cleaner" initial cells, as illustrated in Figure 5. The greedy merge heuristic is then applied on this dense cleaned data grid, and the merges are guided by the data, since the data grid is dense enough. The role of the post-processing step is to improve the final solution, by exploring a local neighborhood of the solution consisting of interval splits, merges and moves of interval boundaries.

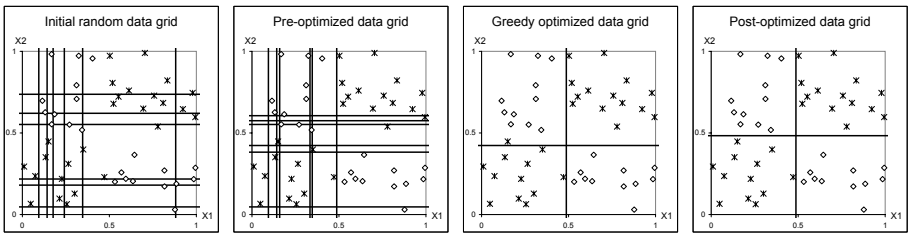


Fig. 5. Main steps in the optimization algorithm: a random initial solution is first generated to start with a dense enough data grid, then cleaned during a pre-processing step, optimized with the greedy bottom-up merge heuristic and improved during the post-processing step.

All these steps are repeated several times in the VNS meta-heuristic described in Section 4.4, which generates several random initial data grids of varying size. The only optimization parameter relates to the number of iterations in the meta-heuristic, which controls the intensity of the search.

All the algorithmic components are useful to achieve an effective search of the space of data grids and efficiently detect informative patterns. Using these algorithms, the empirical threshold for the detection of simple XOR patterns reaches the theoretical threshold, even with one single iteration in the meta-heuristic. For example, bi-dimensional randomly generated patterns require only 40 instances to be detected, and 5-dimensional XOR pattern only 200 instances. In the next sections, we study the detection of more complex XOR patterns, which require more intensive search.

Detection of a Complex Patterns with Few Instances In this experiment, we study the detection of a 10-dimensional XOR pattern in a 10-dimensional input space. The experiment is performed on a set of sample sizes ranging from 1000 to 10000 instances, and repeated 100 times for each sample size. We evaluate the empirical detection threshold for the VNS meta-heuristic, with optimization parameters T, where VNS(T) performs around 2^T iterations of the algorithm from a variety of random initial data grids. Figure 6 reports the average computation time for each sample size and for parameters of the VNS meta-heuristic ranging from $T = 1$ to $T = 12$. We also report the threshold related to the sample size and computation time, among which the XOR pattern is detected in 50% of the cases.

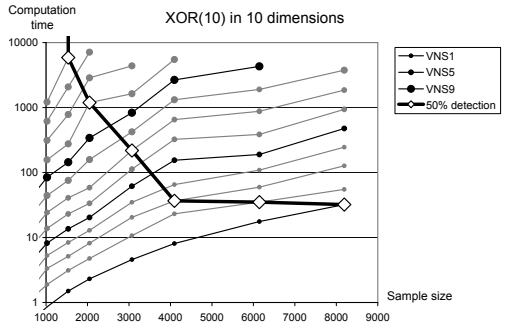


Fig. 6. Study of the algorithm for the detection of the 10-dimensional XOR pattern.

The results show that the empirical detection threshold is close to the theoretical threshold: the pattern is never detected with 1000 instances but frequently detected with only 1500 instances, which is less than 2 instances per cell of the 10-dimensional XOR pattern. However, when the instance number is close from

the theoretical threshold, the problem of finding the correct 10 variable splits among N^{10} possible XOR patterns and $(2^N)^{10}$ potential multivariate discretizations is very hard. In this case, detecting the pattern requires much more time for detection than when the instance number is large enough or when the pattern is simpler. For example, detecting the pattern with only 1500 instances require about one hundred times more computation time than with 4000 instances

Finding a Needle in a haystack In this experiment, we study the detection of a 5-dimensional XOR pattern in a 10-dimensional input space. We use the same protocol as in the previous case, and report the results in Figure 7.

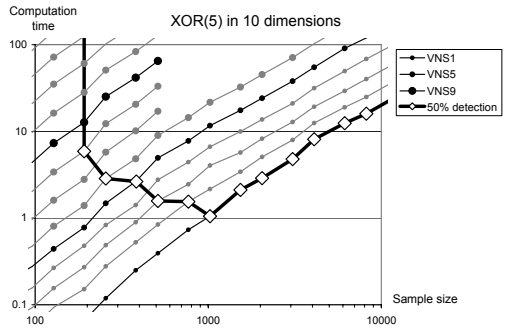


Fig. 7. Study of the algorithm for the detection of the 5-dimensional XOR pattern, hidden in a 10-dimensional input space.

The results show that about 200 instances are sufficient to detect this pattern, which is consistent with the theoretical threshold. However, whereas the 5-dimensional XOR pattern is easily detected even within one or two iterations in the VNS meta-heuristic, the search in that 10-dimensional input space requires much more intensive search.

Apart from of the problem of finding the correct XOR boundaries, which is a difficult task, the problem of variable selection hardens the detection of the pattern. The optimization algorithm is restricted to the exploration of dense data grids, which consist of $K_x \leq \max(\log_2 N, K)$ dimensions. Finding the XOR pattern requires to select a subset of K_x input variables among K, which is a superset of the K_s informative variables. The probability that such a subset contains the informative variable is $\binom{K_x}{Ks}/\binom{K}{Ks}$. For example, for the detection of a 5-dimensional XOR ($K_s = 5$) with 256 instances ($K_x = \log_2 256 = 8$), the probability of finding a potentially good subset is 100% for $K = 5$, 22% for $K = 10$, 0.36% for $K = 20$ and 0.04% for $K = 30$.

We performed an experiment to detect the 5-dimensional XOR in 20 dimensions with samples of size 256. The result confirms that there is enough instances

for a reliable detection of the pattern, but the computational time necessary to detect the pattern in 50% of the cases amounts to about one hundred times that in 10 dimensions. This result, consistent with the ratio 22/0.36, illustrates the problem of finding a needle in a haystack.

Overall, the evaluation criterion given in Formula 3 is able to reliably differentiate informative patterns from noise with very small numbers of instance. The detection of complex patterns is a combinatorial problem, hard to solve when the number of instance is close to the detection threshold, or when the informative patterns are hidden in large dimensional input spaces. Our optimization algorithm manages to reliably and efficiently detect information, with performance close from the theoretical detection threshold.

6 Evaluation on the Agnostic Learning vs. Prior Knowledge Challenge

In this section, we first summarize the evaluation protocol of the challenge, then describe how classifiers are built from data grid models, and finally report the results from a performance and understandability point of view.

6.1 The Agnostic Learning vs. Prior Knowledge Challenge

The purpose of the challenge [Guy07,GSDC07] is to assess the real added value of prior domain knowledge in supervised learning tasks. Five datasets coming from different domains are selected to evaluate the performance of agnostic classifiers vs. prior knowledge classifiers. These datasets come into two formats, as shown in Table 1. In the agnostic format, all the input variables are numerical. In the prior knowledge format, the input variables are both categorical and numerical for three datasets and have a special format in the two other datasets: chemical structure or text.

Name	Domain	Num. ex. train/valid/test	Prior features	Agnostic features
Ada	Marketing	4147/415/41471	14	48
Gina	Handwritting reco.	3153/315/31532	784	970
Hiva	Drug discovery	3845/384/38449	Chem. struct.	1617
Nova	Text classification	1754/175/17537	Text	16969
Sylva	Ecology	13086/1309/130857	108	216

Table 1. Challenge datasets with their prior and agnostic format.

We use all the datasets in their agnostic format and only three of them in their prior format (we have neither domain knowledge nor time within the

challenge schedule to exploit the chemical structure in the Hiva dataset or the native text format in the Nova dataset).

In the case of the Sylva dataset in its prior format, we replace each subset (per record) of 40 binary SoilType variables by one single categorical variable with 40 values. The resulting dataset has only 30 variables instead of 108.

6.2 Building Classifiers from Data Grid Models

In this section, we describe two ways of building classifiers from supervised data grid models.

Data Grid In this evaluation of data grid models, we consider one individual data grid, the MAP one. We build a classifier from a data grid model by first retrieving the cell related to a test instance, and predicting the output conditional probabilities of the retrieved cell. For empty cells, the conditional probability used for the prediction is that of the entire grid.

Data grid models can be considered as a feature selection method, since the input variables whose partition reduces to a single part can be ignored. The purpose of this experiment is to focus on understandable models and evaluate the balance between the number of selected variables and the predictive performance.

Data Grid Ensemble In this evaluation, we focus on the predictive performance rather than on understandability, by the means of averaging the prediction of a large number of classifiers. This principle was successfully exploited in Bagging [Bre96] using multiple classifiers trained from re-sampled datasets. This was generalized in Random Forests [Bre01], where the subsets of variables are randomized as well. In these approaches, the averaged classifier uses a voting rule to classify new instances. Unlike this approach where each classifier has the same weight, the Bayesian Model Averaging (BMA) approach [HMRV99] weights the classifiers according to their posterior probability. The BMA approach has stronger theoretical foundations, but it requires both to be able to evaluate the posterior probability of classifiers and to sample their posterior distribution.

In the case of data grid models, the posterior probability of each model is given by an analytic criterion. Concerning the problem of sampling the posterior distribution of data grid models, we have to strike a balance between the quality of the sampling and the computation time. We adopt a pragmatic choice by just collecting all the data grids evaluated during training, using the optimization algorithms introduced in Section 4. We keep all the local optima encountered in the VNS meta-heuristic and eliminate the duplicates. An inspection of the collected data grids reveals that their posterior distribution is so sharply peaked that averaging them according to the BMA approach almost reduces to the MAP model. In this situation, averaging is useless. The same problem has been noticed by [Bou07a] in the case of averaging Selective Naive Bayes models. To find a trade-off between equal weights as in bagging and extremely unbalanced weights as in the BMA approach, we exploit a logarithmic smoothing of the

posterior distribution called compression-based model averaging (CMA), like that introduced in [Bou07a].

To summarize, we collect the data grid models encountered during the data grid optimization algorithm and weight them according to a logarithmic smoothing of their posterior probability to build a data grid ensemble classifier.

Post-processing The data grid techniques are able to predict the output conditional probabilities for each test instance. When the evaluation criterion is the classification accuracy, predicting the class with the highest conditional probability is optimal. This is not the case for the BER criterion used in the challenge. We post-process each trained classifier by optimizing the probability threshold in order to maximize the BER. This optimization is performed directly on the train dataset.

6.3 Evaluation Results

Our four submissions related to supervised data grid models are named *Data Grid (MAP)* and *Data Grid (CMA)* in the prior or agnostic track and dated from February 27, 2007 for the challenge March 1st, 2007 milestone. The classifiers are trained with the any time optimization algorithm described in Section 4 using VNS(12) parameter. About 4000 data grids are evaluated, needing around one hour optimization time per dataset. Table 2 and Table 3 report our results in the agnostic and prior track.

Dataset	Winner	Best BER	Data Grid (CMA)	Data Grid (MAP)
Ada	Roman Lutz	0.166	0.1761	0.2068
Gina	Roman Lutz	0.0339	0.1436	0.1719
Hiva	Vojtech Franc	0.2827	0.3242	0.3661
Nova	Mehreen Saeed	0.0456	0.1229	0.2397
Sylva	Roman Lutz	0.0062	0.0158	0.0211

Table 2. Best challenge results vs. our data grid methods results for the datasets in the agnostic track.

Dataset	Winner	Best BER	Data Grid (CMA)	Data Grid (MAP)
Ada	Marc Boullé	0.1756	0.1756	0.2058
Gina	Vladimir Nikulin	0.0226	0.1254	0.1721
Sylva	Roman Lutz	0.0043	0.0228	0.0099

Table 3. Best challenge results vs. our data grid methods results for the Gina, Hiva and Nova datasets in the prior track.

The data grid classifiers obtain good results on the Ada and Sylva datasets, especially on the prior track, with a winning submission for the Ada dataset. The other datasets contain very large numbers of variables, which explains the poor performance of the data grid models. Since individual data grid models are essentially restricted to about $\log_2 N$ selected variable, they cannot exploit much of the information contained in the representation space. This is analyzed in Section 6.4.

The data grid ensemble classifiers confirm the benefits of compression-based model averaging. They obtain a very significant improvement of the BER criterion compared to the individual data grid classifiers. This focus on predictive performance is realized at the expense of understandability, since each trained data grid ensemble averages several hundreds of elementary data grid models.

However, even data grid ensembles fail to achieve competitive performance for datasets with large numbers of variables. A close inspection reveals that although about 4000 data grids are evaluated for each dataset, only a few hundreds ($\approx$ 500) of different solutions are retrieved. Removing the duplicates significantly improves the performances, but there is still too much redundancy between the data grids to produce an efficient ensemble classifier. Furthermore, a few hundred of redundant classifiers, each with only $\approx \log_2 N$ variables, is not enough to exploit all the variables (think of Nova with 17000 variables for example). In future work, we plan to improve our meta-heuristic in order to better explore the search space and to collect a set of data grid solutions with better diversity.

6.4 Benefit for Understandability

Let us now focus on understandability and inspect the number of selected variables in each trained data grid model. In the agnostic track, the MAP data grid exploits only 5 variables for Ada, 5 for Gina, 4 for Hiva, 8 for Nova and 8 for Sylva. In the prior track, the MAP data grid exploits 6 variables for Ada, 7 for Gina and 4 for Sylva. These numbers of variables are remarkably small w.r.t the BER performance of the models.

In Table 4, we summarize the MAP data grid trained using the 4562 train+valid instances of the Ada dataset in the prior track. This data grid selects six variables among 14 and obtains a 0.2068 test BER. The selected variables are relationship, occupation, education number, age, capital gain and capital loss, which are partitioned into 2, 2, 2, 2, 3 and 3 groups or intervals. The relationship variable is grouped into Married = {Husband, Wife} vs. Not Married = {Not-in-family, Own-child, Unmarried, Other-relative}, and the occupation into Low = {Craft-repair, Other-service, Machine-op-inspct, Transport-moving, Handlers-cleaners, Farming-fishing, Priv-house-serv} vs. High = {Prof-specialty, Exec-managerial, Sales, Adm-clerical, Tech-support, Protective-serv, Armed-Forces}. Overall, the data grid contains $144 = 2*2*2*2*3*3$ cells, but 57 of them are non empty and the twelve most frequent cells reported in Table 4 contains 90% of the instances.

Each cell of the data grid can directly be interpreted as a decision rule. For example, the most frequent cell is described by Rule 1, with a support of 736 instances.

ID	relationship	occupation	education number	age	capital gain	capital loss	frequency	% class 1
1	Married	Low	≤ 12	> 27	≤ 4668	≤ 1805	736	22.1%
2	Not married	Low	≤ 12	> 27	≤ 4668	≤ 1805	577	3.1%
3	Not married	High	≤ 12	> 27	≤ 4668	≤ 1805	531	5.8%
4	Married	High	≤ 12	> 27	≤ 4668	≤ 1805	489	41.3%
5	Married	High	> 12	> 27	≤ 4668	≤ 1805	445	68.5%
6	Not married	Low	≤ 12	≤ 27	≤ 4668	≤ 1805	425	0.2%
7	Not married	High	≤ 12	≤ 27	≤ 4668	≤ 1805	316	0.6%
8	Not married	High	> 12	> 27	≤ 4668	≤ 1805	268	20.5%
9	Not married	High	> 12	≤ 27	≤ 4668	≤ 1805	112	0.9%
10	Married	Low	≤ 12	≤ 27	≤ 4668	≤ 1805	96	5.2%
11	Married	High	> 12	> 27	> 5095	≤ 1805	93	100.0%
12	Married	Low	> 12	> 27	≤ 4668	≤ 1805	50	24.0%

Table 4. Most frequent cells in the best individual data grid model for the Ada dataset in the prior track.

Rule 1: IF relationship $\in$ Married = {Husband, Wife}
occupation $\in$ Low = {Craft-repair, Other-service, Machine-op-inspct,...}
education number ≤ 12
age > 27
capital gain ≤ 4668
capital loss ≤ 1805
THEN P(class=1) = 22.1%

The whole data grid forms a set of rules [Mit97] which forms a partition (not a coverage) of the training set. Since all the rules exploit the same variables with the same univariate partitions, interpretation is much easier. For example, rule 5 (ID cell=5 in Table 4) has a large support of 445 instances with 68.5% of class 1. Rule 4 with 41.3% of class 1 only differs in the education number variable (≤ 12 vs. > 12), and rule 8 with 20.5% of class 1 in the relationship variable (Not married vs. Married).

7 Conclusion

The data grid models introduced in this paper are based on a partitioning model of each input variable, into intervals for numerical variables and into groups of values for categorical variables. The cross-product of the univariate partitions, called a data grid, allows to quantify the conditional information relative to the output variable. We have detailed this technique in the multivariate case, with a Bayesian approach for model selection and sophisticated combinatorial algorithms to efficiently search the model space.

In extensive artificial experiments, we have shown that our technique is able to reliably detect complex patterns. Our experiments allow to quantify the limits

of the approach, with data grid models limited to about $\log_2$ variables, and provides insights on the relation between the complexity of the patterns and the required computation time necessary to detect them.

We have introduced two ways of building classifiers from data grids and experimented them on the Agnostic Learning vs. Prior Knowledge challenge. This preliminary evaluation looks promising since our method was first on one of the datasets. The analysis of the results demonstrates that the data grid models are of considerable interest for data understandability and data preparation.

In future research, we plan to investigate on how to better exploit the potential of these models to build more powerful classifiers. Apart from improving the optimization algorithms and building ensemble classifiers based on a wider diversity of data grid models, we intend to further explore the problem of conditional density estimation. Whereas the naive Bayes strategy [LIT92] is to factorize the multivariate density estimation on univariate estimations, our strategy with the data grid models is to directly estimate the multivariate joint density, which encounters a limit in the number of considered variables. Between these two opposite strategies, other approaches have been considered, based on a relaxation of the naive Bayes assumption. This is the case for example in semi-naive Bayesian classifiers [Kon91] or in Bayesian networks classifiers [FGG97]. In this context, we expect data grid models to be promising building bricks of future better multivariate density estimators.

References

[AS70] M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publications Inc., New York, 1970.

[BFOS84] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. California: Wadsworth International, 1984.

[BM96] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1996. http://www.ics.uci.edu/mlearn/MLRepository.html.

[Bou05] M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

[Bou06] M. Boullé. MODL: a Bayes optimal discretization method for continuous attributes. *Machine Learning*, 65(1):131–165, 2006.

[Bou07a] M. Boullé. Compression-based averaging of selective naive Bayes classifiers. *Journal of Machine Learning Research*, 8:1659–1685, 2007.

[Bou07b] M. Boullé. Optimal bivariate evaluation for supervised learning using data grid models. *Advances in Data Analysis and Classification*, 2007. submitted.

[Bou08] M. Boullé. Bivariate data grid models for supervised learning. Technical Report NSM/R&D/TECH/EASY/TSI/4/MB, France Telecom R&D, 2008. http://perso.rd.francetelecom.fr/boulle/publications/-BoulleNTTSI4MB08.pdf.

[Bre96] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[FGG97] N. Friedman, D. Geiger, and M. Goldsmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[GSDC07] I. Guyon, A.R. Saffari, G. Dror, and G. Cawley. Agnostic learning vs. prior knowledge challenge. In *International Joint Conference on Neural Networks*, 2007.

[Guy07] I. Guyon. Agnostic learning vs. prior knowledge challenge, 2007. http://clopinet.com/isabelle/Projects/agnostic/.

[HM01] P. Hansen and N. Mladenovic. Variable neighborhood search: principles and applications. *European Journal of Operational Research*, 130:449–467, 2001.

[HMRV99] J.A. Hoeting, D. Madigan, A.E. Raftery, and C.T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999.

[Kas80] G.V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29(2):119–127, 1980.

[Kon91] I. Kononenko. Semi-naive Bayesian classifier. In Y. Kodrato, editor, *Sixth European Working Session on Learning (EWSL91)*, volume 482 of *LNAI*, pages 206–219. Springer, 1991.

[LIT92] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *10th national conference on Artificial Intelligence*, pages 223–228. AAAI Press, 1992.

[Mit97] T.M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[Qui93] J.R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[ZR00] D.A. Zighed and R. Rakotomalala. *Graphes d'induction*. Hermes, France, 2000.

Chapter 2

Data Grid Models for Unsupervised Learning and Coclustering

Multivariate Data Grid Models for Unsupervised Learning and Coclustering

Marc Boullé

France Télécom R&D Lannion,
marc.boulle@orange-ftgroup.com

Abstract. Exploratory analysis is a key step in any data mining project, which consists in inspecting the available data prior to modeling. In this paper, we extend supervised data grid models to the unsupervised case and present their benefit for data exploration. Unsupervised data grid models are based on a partitioning of each variable, in intervals in the numerical case and in groups of values in the categorical case. The cross-product of the univariate partitions forms a multivariate partition of the representation space into a set of cells. This multivariate partition, called data grid, allows to evaluate the correlation between the variables. The best data grid is searched owing to a Bayesian model selection approach and to combinatorial algorithms.
We show that unsupervised data grid models offer a variety of techniques for exploratory analysis, such as non parametric correlation or joint density estimation, visualization, association rule mining, variable selection or coclustering of instances and variables. We also report results in the Agnostic Learning vs. Prior Knowledge Challenge, where we achieved very competitive results with our coclustering method exploited in a semi-supervised learning context.

1 Introduction

The data mining process [CCK+00] consists in six steps: business understanding, data understanding, data preparation, modeling, evaluation and deployment. Whereas most the emphasis in the literature is on the modeling step, the data preparation step, which represents 80% of the problem [Pyl99,Mam06], is both time consuming and critical for the quality of the modeling. The issue is to reduce as much as possible the need of hand-crafted solutions for any particular task.

In this paper, we introduce a new method to automatically, rapidly and reliably evaluate the joint probability distribution of any subset of variables, numerical or categorical. We extend the supervised data grid models introduced in Chapter 1 to the supervised case. Each variable is partitioned into a set of intervals (or groups of values), and the cross-product of the univariate partitions forms a data grid of cells. The instances are distributed in the cells of the grid according to a multinomial distribution. Such models describe the joint distribution between the variables. Applying the MODL approach presented in

Chapter 1, a prior is defined on the model parameters, and the maximum a posterior (MAP) data grid is optimized using the same search algorithms as in the supervised case.

In the supervised case, we have a set of input variables and the data grid model consists in partitioning the input space in cells, with a local description of the output distribution in each cell. In the unsupervised case, we consider all the variables as output variables and our task is to describe jointly all of them. The principle of our approach is that unsupervised data grid models are able to describe the redundancy between the variables, so that the description of each variable given the model of redundancy is more compact.

The rest of the paper is organized as follows. Section 2 introduces unsupervised data grid models in the bivariate case for two numerical variables, and Section 3 for two categorical variables. Section 4 generalizes the approach to any subset of variables of any type, numerical or categorical, and Section 5 summarizes the optimization algorithms exploited to search the MAP data grid. Section 6 shows how unsupervised data grid models can be applied to the problem of coclustering of the instances and variables of a dataset, and how to exploit this for supervised learning owing to a semi-supervised approach. Section 7 reports experiments performed on the agnostic learning vs. prior knowledge challenge datasets [GSDC07] and analyzes their interest for explanatory analysis tasks, such as correlation study, density estimation, visualization or rule set mining. In the case of text classification, we show that our coclustering technique is of high interest, with both understandable insights on the text corpus and remarkable classification performance. Finally, Section 8 gives a summary and discusses future work.

2 Bivariate Discretization of Numerical Variables

In this section, we focus on the case of two numerical variables and introduce unsupervised bivariate data grid models and their evaluation criterion. We then show how such models can be interpreted as nonparametric models of the correlation between the ranks of each variable.

2.1 Presentation

The purpose of unsupervised learning is to identify dense clusters of instances. It is closely related to density estimation, which aims at modeling the true density of the data. In order to illustrate this problem with two numerical variables, we present in Figure 1 the scatter-plot of the iris dataset [Fis36] considered for the estimation of the joint density of the petal length and sepal length variables. The figure shows a dense cluster on the bottom-left and a dense diagonal region on the right. We propose to exhibit the dense regions of the dataset by discretizing both variables. For example, the grid with sixteen cells presented on the left of Figure 2 allows to summarize the dense regions of the datasets, which can be seen as clusters in an unsupervised approach. They also allow to approximate

the underlying density of the dataset, owing to the cell frequencies, presented on the right of Figure 2.

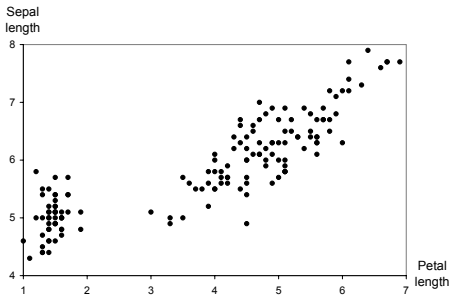
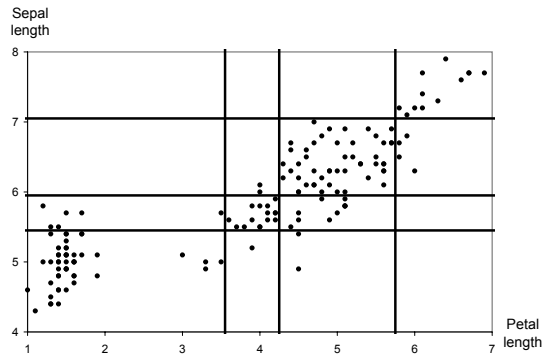


Fig. 1. Scatter-plot of the iris dataset considered for the problem of estimating the joint density of the petal length and sepal length variables.



Cell frequencies

0	0	0	12	12
0	2	49	4	55
6	15	10	0	31
49	1	2	0	52
55	18	61	16	150

Fig. 2. Bivariate data grid with sixteen cells for the petal length and sepal length variables of the iris dataset.

The issue is to find a trade-off between the precision of the density estimation and the generalization ability, on the basis of the grain level of the discretization grid. Fine grain grids allow a precise density estimation, whereas coarse grain data grids tend to be more robust.

2.2 Formalization

Let us formalize this problem using a Bayesian model selection approach. First of all, we make the two important choices below:

1. modeling the rank, not the values,
2. modeling the finite data sample.

With the first choice, we seek a model which is independent of any monotonic transformation of the variables. We thus focus on the intrinsic correlation between the variables, ignoring any potential scaling effect and being more robust to outliers. With the second choice, our purpose is to model the data with as few instances as possible.

Given these modeling choices, we introduce the family of unsupervised data grid models in Definition 1.

Definition 1. *An unsupervised bivariate discretization model is defined by:*

- *a number of intervals for each variable,*
- *the distribution of the instances among the cells of the resulting data grid.*

Notations 1 [1]

- Y_1, Y_2*: variables (both considered as output variables)*
- N*: number of training instances*
- $D = \{D_1, D_2, \ldots, D_N\}$*: training instances*
- J_1, J_2*: number of intervals for each variable*
- $G = J_1 J_2$*: number of cells in the resulting data grid*
- N_{j_1}*: number of instances in the interval* j_1 *of variable* Y_1
- N_{j_2}*: number of instances in the interval* j_2 *of variable* Y_2
- $N_{j_1 j_2}$*: number of instances in the cell* (j_1, j_2) *of the data grid*

An unsupervised data grid model is entirely characterized by the parameters $J_1,\ J_2,\ \{N_{j_1 j_2}\}_{1 \leq j_1 \leq J_1, 1 \leq j_2 \leq J_2}$. The number of instances in each interval can be deduced by adding the cell frequencies in the rows or columns of the grid, according to $N_{j_1} = \sum_{j_2=1}^{J_2} N_{j_1 j_2}$ and $N_{j_2} = \sum_{j_1=1}^{J_1} N_{j_1 j_2}$.

Our aim is to select the best model given the available data, i.e. the most likely model given the data. Adopting a Bayesian approach, it comes to maximize:

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

The data distribution $p(D)$ being constant whatever the model M, it comes to maximize $p(M)p(D|M)$ which can be written:

$$p(M)p(D|M) = p(J_1, J_2)p(\{N_{j_1 j_2}\}|J_1, J_2)p(D|M).$$

To be able to evaluate a given model, we have to choose a prior distribution for the model parameters and a likelihood function. In Definition 2, we formalize our choices by using the independence assumption and proposing a uniform distribution at each stage of the prior parameter structure and of the likelihood function.

[1] By abuse of notation, we employ N_{j_1} and N_{j_2} (instead of $N_{j_1}^{(1)}$ and $N_{j_2}^{(2)}$ for example) to denote the numbers of instances in the intervals of Y_1 and Y_2.

Definition 2. *The prior for the parameters of an unsupervised bivariate discretization model and the likelihood function of the data given a model are chosen hierarchically and uniformly at each level:*

- *the numbers of intervals J_1 and J_2 are independent from each other, and uniformly distributed between 1 and N,*
- *for a data grid of given size (J_1, J_2), every distribution of the N instances on the $G = J_1, J_2$ cells of the grid is equiprobable,*
- *for a given interval of a given variable, every distribution of the ranks of the values is equiprobable.*

Taking the negative log of the probabilities, this provides the evaluation criterion given in Theorem 1.

Theorem 1. *An unsupervised bivariate discretization model distributed according to a uniform hierarchical prior is Bayes optimal if its evaluation according to the following criteria is minimal*

$$\begin{aligned} &2\log N + \log \binom{N+G-1}{G-1} \\ &+ \log N! - \sum_{j_1=1}^{J_1}\sum_{j_2=1}^{J_2} \log N_{j_1 j_2}! + \sum_{j_1=1}^{J_1} \log N_{j_1}! + \sum_{j_2=1}^{J_2} \log N_{j_2}! \end{aligned} \tag{1}$$

Proof. The first hypothesis introduced in Definition 2 gives that $p(J_1, J_2) = p(J_1)p(J_2) = \frac{1}{N}\frac{1}{N}$.

The second hypothesis is that all the distributions of the N instances into $G = J_1 J_2$ cells are equiprobable for given J_1 and J_2. Dividing the N instances into G cells is equivalent to decompose the number N as the sum of the $N_{j_1 j_2}$ frequencies of the cells. Using combinatorics, we can prove that this number of choices for the parameters of this multinomial distribution is equal to $\binom{N+G-1}{G-1}$. Using the equiprobability assumption, we finally obtain

$$p(\{N_{j_1 j_2}\}|J_1, J_2) = \frac{1}{\binom{N+G-1}{G-1}}.$$

The prior terms being explicited, it remains to evaluate the likelihood of the data, i.e. the probability of observing the data in the data grid cells knowing the multinomial distribution model. The number of ways of observing N instances distributed according to such multinomial law is given by the multinomial term

$$\frac{N!}{\prod_{j_1=1}^{J_1}\prod_{j_2=1}^{J_2} N_{j_1 j_2}!}.$$

To finish, according to the last hypothesis, for a given interval of a given variable, every distribution of the ranks of the values are equiprobable. Using $N_{j_1} = \sum_{j_2=1}^{J_2} N_{j_1 j_2}$ and $N_{j_2} = \sum_{j_1=1}^{J_1} N_{j_1 j_2}$, we compute the frequency in each interval for each variable. The number of distribution of the ranks of N_{j_1} instances is $N_{j_1}!$, which leads to the last terms.

By taking negative logarithms, we obtain the above Formula 1.

2.3 Interpretation

Since negative log of probabilities can be interpreted as code length according to [Sha48], the evaluation criterion presented in Formula 1 can also be obtained from the minimum description length (MDL) principle of [Ris78]. Maximizing the posterior probability $p(M)p(D|M)$ is equivalent to minimizing the code length $l(M) + l(D|M)$ of the model plus that of the data given the model.

In the light of the MDL approach, the terms $2 \log N$ in Formula 1 encodes the choice of the numbers of intervals for each variable. The term $\log \binom{N+G-1}{G-1}$ represents the parameters of the multinomial distribution of the instances on the cells of the data grid. The term $\log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \log N_{j_1 j_2}!$ encodes the position of the instances in the grid given the multinomial distribution. Finally, the terms $\sum_{j_1=1}^{J_1} \log N_{j_1}! + \sum_{j_2=1}^{J_2} \log N_{j_2}!$ describe the rank of each instance locally to each interval. As the intervals are ordered within the data grid, the global rank of each instance is thus completely described.

In the case of the null model $M_\emptyset$ containing one single cell, Formula 1 reduces to

$$c(M_\emptyset) = 2 \log N + 2 \log N!,$$

which mainly corresponds to $N!$ ways of specifying the rank of N instances for each variable.

In the case of the maximum model M_{Max} containing N intervals per variable and N^2 cells, we obtain

$$\begin{aligned} c(M_{Max}) =& 2 \log N + \log \binom{N + N^2 - 1}{N^2 - 1} + \log N!, \\ =& 2 \log N + \sum_{n=0}^{N-1} \log(N^2 + n), \\ >& 2 \log N + 2N \log N, \end{aligned}$$

which shows that the null model always has a greater posterior probability than the maximum model.

Intermediate models allow to describe dense regions in cells where the ranks of the variables are correlated. In the case of two independent variables, describing jointly the rank of both variables reduces to describing independently the rank of each variable, as in the null model. The data grid models allow a non parametric description of the correlation between the ranks of the variables. The penalization of the model cost is balanced by a shorter description of each variable rank given the model. The best trade-of is searched owing to a Bayesian (or MDL) model selection approach.

Example with two identical numerical variables. Let us consider two identical numerical variables $Y_1 = Y_2$, and data grid models M_J based on J ($J = J_1 = J_2$) equidistributed intervals, as illustrated in Figure 3.

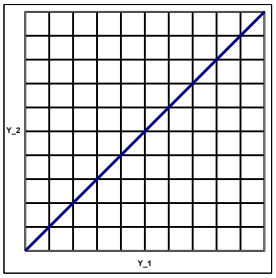


Fig. 3. Bivariate discretization data grid with ten equidistributed intervals for two identical variables $Y_1 = Y_2$.

For each variable, the frequency of each interval is N/J. Among the $G = J^2$ cells of the data grid, the J diagonal cells each contain N/J instances, and the other cells are empty. The evaluation criterion $c(M_J)$ of the grid is thus equal to

$$c(M_J) = 2\log N + \log \binom{N + J^2 - 1}{J^2 - 1} + \log N! - J\log(N/J)! + 2J\log(N/J)!.$$

For a fixed value of J and using the asymptotic approximation $\log N! = N(\log N - 1)$ based on Stirling formula, we obtain $c(M_1) \approx 2N \log N$ and

$$c(M_J) \approx c(M_1) + (J^2 - 1)\log N - N \log J. \quad (2)$$

Formula 2 shows that in the considered case of a perfect correlation between the two variables, the data grid model has an asymptotic cost which decreases with the size of the grid. In the non asymptotic case, we have showed in a numerical experiment (not reported in the paper) that the optimal data grid is obtained for $J \approx \frac{\sqrt{N}}{2}$.

3 Bivariate Value Grouping of Categorical Variables

In this section, we focus on the case of two categorical variables and introduce unsupervised bivariate data grid models and their evaluation criterion. We then show how such models can be interpreted as nonparametric models of the correlation between the values of each variable.

3.1 Presentation

Our objective is to provide a joint description of two categorical variables Y_1 et Y_2, as illustrated in Figure 4. In the case of categorical variables with many values, the contingency table between the variables is sparse and does not allow to identify reliable correlations. Standard statistical tests rely on approximations which are valid only asymptotically. For example, the chi-square test

Fig. 4. Example of joint density for two categorical variables Y_1 having 4 values a, b, c, d and Y_2 having 4 values A, B, C, D. The initial contingency table on the left contains instances only on one half on the cells (tagged as •), and the remaining cells are empty. After the bivariate value grouping, the preprocessed contingency table on the right provides a synthetic description of the correlation between Y_1 et Y_2.

requires an expected frequency of at least 5 in each cell of the contingency table [Coc54,CL03], which does not permit its application in case of sparsity. Grouping the values of each variable allows to raise the cell frequencies (at the expense of potentially mixing interesting patterns), and to be more confident in the observed correlation. However, since many grouping models might be considered, there is a risk of overfitting the data. The issue is to find a trade-off between the quality of the density estimation and the generalization ability, on the basis of the grain level of the grid.

3.2 Formalization

Like in the numerical case, we introduce a family of unsupervised data grid model to describe the joint distribution of the data. In the numerical case, describing the data reduces to describing the rank of the instances for each variable. In the categorical case, this turns into describing the value of the instances for each variable. We still consider partitioning models, in groups of values in the categorical case instead of intervals of ranks in the numerical case. This family of models is formalized in Definition 3.

Definition 3. *An unsupervised bivariate value grouping model is defined by:*

- *a number of groups for each variable,*
- *for each variable, the repartition of the values into the groups of values,*
- *the distribution of the instances of the data sample among the cells of the resulting data grid,*
- *for each variable and each group, the distribution of the instances of the group on the values of the group.*

Notations 2

- Y_1, Y_2*: variables (both considered as output variables)*
- V_1, V_2*: number of values for each variable (assumed as prior knowledge)*
- N*: number of training instances*
- $D = \{D_1, D_2, \ldots, D_n\}$*: training instances*
- J_1, J_2*: number of groups for each variable*

- $G = J_1 J_2$: *number of cells in the resulting data grid*
- $j_1(v_1), j_2(v_2)$: *index of the group containing value* v_1 *(resp.* v_2*)*
- m_{j_1}, m_{j_2}: *number of values in group* j_1 *(resp.* j_2*)*
- n_{v_1}, n_{v_2}: *number of instances for value* v_1 *(resp.* v_2*)*
- N_{j_1}: *number of instances in the group* j_1 *of variable* Y_1
- N_{j_2}: *number of instances in the group* j_2 *of variable* Y_2
- $N_{j_1 j_2}$: *number of instances in the cell* (j_1, j_2) *of the data grid*

We assume that the numbers of values V_1 and V_2 per categorical variable are known in advance and we aim at modeling the joint distribution of the finite data sample of size N on these values. The family of models introduced in Definition 3 is completely defined by the parameters describing the partition of the values into groups of values

$$J_1, J_2, \{j_1(v_1)\}_{1 \leq v_1 \leq V_1}, \{j_2(v_2)\}_{1 \leq v_2 \leq V_2},$$

by the parameters of the multinomial distribution of the instances on the data grid cells

$$\{N_{j_1 j_2}\}_{1 \leq j_1 \leq J_1, 1 \leq j_2 \leq J_2},$$

and by the parameters of the multinomial distribution of the instances of each group on the values of the group

$$\{n_{v_1}\}_{1 \leq v_1 \leq V_1}, \{n_{v_2}\}_{1 \leq v_2 \leq V_2}.$$

The numbers of values per groups m_{j_1} and m_{j_2} are derived from the specification of the partitions of the values into groups: they do not belong to the model parameters. Similarly, the number of instances in each group can be deduced by adding the cell frequencies in the rows or columns of the grid, according to $N_{j_1} = \sum_{j_2=1}^{J_2} N_{j_1 j_2}$ and $N_{j_2} = \sum_{j_1=1}^{J_1} N_{j_1 j_2}$.

In order to select the best model, we apply a Bayesian approach, using the prior distribution on the model parameters described in Definition 4.

Definition 4. *The prior for the parameters of an unsupervised bivariate value grouping model are chosen hierarchically and uniformly at each level:*

- *the numbers of groups* J_1 *and* J_2 *are independent from each other, and uniformly distributed between* 1 *and* V_1 *for* Y_1*, between* 1 *and* V_2 *for* Y_2*,*
- *for a given number of groups* J_1 *of* Y_1*, every partition of the* V_1 *values into* J_1 *groups is equiprobable,*
- *for a given number of groups* J_2 *of* Y_2*, every partition of the* V_2 *values into* J_2 *groups is equiprobable,*
- *for a data grid of given size* (J_1, J_2)*, every distribution of the* N *instances on the* $G = J_1, J_2$ *cells of the grid is equiprobable,*
- *for a given group of a given variable, every distribution of the instances of the group on the values of the group is equiprobable.*

Taking the negative log of the probabilities, this provides the evaluation criterion given in Theorem 2.

Theorem 2. *An unsupervised bivariate value grouping model distributed according to a uniform hierarchical prior is Bayes optimal if its evaluation according to the following criteria is minimal*

$$\begin{aligned}
&\log V_1 + \log V_2 + \log B(V_1, J_1) + \log B(V_2, J_2) \\
&+ \log \binom{N+G-1}{G-1} + \sum_{j_1=1}^{J_1} \log \binom{N_{j_1}+m_{j_1}-1}{m_{j_1}-1} + \sum_{j_2=1}^{J_2} \log \binom{N_{j_2}+m_{j_2}-1}{m_{j_2}-1} \\
&+ \log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \log N_{j_1 j_2}! \\
&+ \sum_{j_1=1}^{J_1} \log N_{j_1}! + \sum_{j_2=1}^{J_2} \log N_{j_2}! - \sum_{v_1=1}^{V_1} \log n_{v_1}! - \sum_{v_2=1}^{V_2} \log n_{v_2}!
\end{aligned} \tag{3}$$

The first line in Formula 3 relates to the prior distribution of the group numbers J_1 et J_2 and to the specification the partition of the values in groups for each variable. These terms are the same as in the case of the MODL supervised univariate value grouping method [Bou05]. $B(V, J)$ is the number of divisions of V values into J groups (with eventually empty groups). When $J = V$, $B(V, J)$ is the Bell number. In the general case, $B(V, J)$ can be written as $B(V, J) = \sum_{j=1}^{I} S(V, j)$, where $S(V, j)$ is the Stirling number of the second kind [AS70], which stands for the number of ways of partitioning a set of V elements into j nonempty sets.

The second line in Formula 3 represents the specification of the parameters of the multinomial distribution of the N instances on the G cells of the data grid, followed by the specification of the multinomial distribution of the instances of each group on the values of the group. The third line stands for the likelihood of the distribution of the instances on the data grid cells, by the mean of a multinomial term. The last line corresponds to the likelihood of the distribution of the values locally to each group, for each variable.

3.3 Interpretation

In the case of the null model $M_\emptyset$ containing one single cell, Formula 3 reduces to

$$\begin{aligned}
c(M_\emptyset) = &\log V_1 + \log V_2 + \log \binom{N+V_1-1}{V_1-1} + \log \binom{N+V_2-1}{V_2-1} \\
&+ \log \frac{N!}{n_{v_1}! n_{v_2}! \dots n_{V_1}!} + \log \frac{N!}{n_{v_1}! n_{v_2}! \dots n_{V_2}!}
\end{aligned} \tag{4}$$

which corresponds to the posterior probability of the multinomial model for the distribution of the instances on the values, for each variable. This means that each variable is described independently.

Like in the numerical case, the data grid models allow a non parametric description of the correlation between the variables and the best trade-of is searched owing to a Bayesian (or MDL) model selection approach.

Example with two identical categorical variables. Let us consider two identical categorical variables $Y_1 = Y_2$ and the maximum data grid model M_{Max} with as many groups as values ($J_1 = V_1$), as illustrated in Figure 5. The evaluation criterion of the grid is equal to

$$c(M_{Max}) = 2\log V_1 + 2\log B(V_1, V_1) + \log \binom{N + V_1^2 - 1}{V_1^2 - 1} + \log \frac{N!}{n_{v_1}! n_{v_2}! \ldots n_{V_1}!} \tag{5}$$

d	∅	∅	∅	•
c	∅	∅	•	∅
b	∅	•	∅	∅
a	•	∅	∅	∅
	a	b	c	d

Fig. 5. Bivariate value grouping data grid with as many groups as values for two identical categorical variables $Y_1 = Y_2$, having four values a, b, c and d.

If we compare $c(M_\emptyset)$ in Formula 4 to $c(M_{Max})$ in Formula 5 in the case of two identical categorical variables, we observe an overhead in the prior terms of the maximum model (specification of the value grouping with Bell numbers and specification of the distribution of the N instances on the ${V_1}^2$ cells of the grid). On the opposite, the likelihood term is divided by a factor two: since the correlation between the variables is perfectly detected owing to the data grid model, describing the joint distribution of the data given the model reduces to describing the distribution of one single variable.

To fix the ideas, let us compare Formulae 4 and 5 in the asymptotic case. The multinomial term for the distribution of the values of a categorical variable can be approximated with

$$\log \frac{N!}{n_{v_1}! n_{v_2}! \ldots n_{V_1}!} \approx N H(Y_1),$$

where $H(Y_1)$ is the Shannon entropy of variable Y_1 [Sha48]. In the case of the null model having one single cell, we get

$$c(M_\emptyset) \approx 2(V_1 - 1)\log N + 2NH(Y_1).$$

In the case of the maximum model with as many groups as values, we obtain

$$c(M_{Max}) \approx (V_1^2 - 1)\log N + NH(Y_1).$$

The maximum model, which detects the correlation between the variables, will thus be preferred as soon as there are enough instances compared to the number of values. It is noteworthy that Formulae 4 and 5 allow to select the best model in the non asymptotic case.

4 Unsupervised Data Grids for any Subset of Variables

In this section, we extend unsupervised bivariate data grids to the multivariate case.

4.1 Evaluation Criterion for Unsupervised Data Grids

The purpose of the unsupervised data grid model is to describe the joint distribution of all the variables. When a variable is partitioned into at least two parts, it can be considered as selected, and discarded in case of one single part. Like in the supervised case presented in Chapter 1, we use this variable selection explicitly by introducing in Definition 5 a new level in the model parameters.

Definition 5. *An unsupervised data grid model is defined by:*

- *a subset of selected variables,*
- *a number of parts (groups or intervals) for each selected variable,*
- *for each categorical variable, the repartition of the values into the groups of values,*
- *the distribution of the instances of the data sample among the cells of the data grid,*
- *for each categorical variable and each group, the distribution of the instances of the group on the values of the group.*

Notations 3

- K*: number of variables*
- $\mathbb{K} = \{Y_1, Y_2, \ldots, Y_K\}$*: set of variables*
- $\mathbb{K}_n$*: subset of numerical variables*
- $\mathbb{K}_c$*: subset of categorical variables*
- $V_k, k \in \mathbb{K}_c$*: number of values of categorical variable* X_k
- N*: number of training instances*
- K_s*: number of selected variables*
- $\mathbb{K}_s$*: subset of selected variables* ($|\mathbb{K}_s| = K_s$)
- J_k*: number of parts (intervals or groups) of the univariate partition of variable* Y_k
- $G = \prod_{k=1}^{K} J_k$*: number of cells in the data grid*
- $m_{j_k}, k \in \mathbb{K}_c$*: number of values in group* j_k *of categorical variable* Y_k
- $n_{v_k}, k \in \mathbb{K}_c$*: number of instances for value* v_k *of categorical variable* Y_k
- $N_{j_k}, k \in \mathbb{K}$*: number of instances in part (interval or group)* j_k *of variable* Y_k
- $N_{j_1 j_2 \ldots j_K}$*: number of instances in cell* $(j_1, i_2, \ldots, j_K)$ *of the data grid*

We extend to the multivariate case the prior introduced in Section 2 for bivariate unsupervised numerical data grids and in Section 3 for bivariate unsupervised categorical data grids. For the variable selection parameters, we reuse the prior described in Chapter 1 in the case of supervised data grid models. We apply the Bayesian model selection approach and obtain the evaluation criterion of a data grid model M in Formula 6.

$$\begin{aligned}
&\log(K+1) + \log \binom{K+K_s-1}{K_s} \\
&+ \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_n} \log N + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_c} \log V_k + \sum_{k \in \mathbb{K}_s \cap \mathbb{K}_c} \log B(V_k, J_k) \\
&+ \log \binom{N+G-1}{G-1} + \sum_{k \in \mathbb{K}_c} \sum_{j_k=1}^{J_k} \log \binom{N_{j_k}+m_{j_k}-1}{m_{j_k}-1} \\
&+ \log N! - \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_K=1}^{J_K} \log N_{j_1 j_2 \cdots j_K}! \\
&+ \sum_{k \in \mathbb{K}} \sum_{j_k=1}^{J_k} \log N_{j_k}! - \sum_{k \in \mathbb{K}_c} \sum_{v_k=1}^{V_k} \log n_{v_k}!
\end{aligned} \tag{6}$$

4.2 Interpretation

In the case where no variable is selected, the resulting data grid contains one single cell and Formula 6 reduces to

$$\log(K+1) + \sum_{k \in \mathbb{K}_c} \log \binom{N+V_k-1}{V_k-1} + \sum_{k \in \mathbb{K}_n} \log N! + \sum_{k \in \mathbb{K}_c} \log \frac{N!}{n_{v_k}! n_{v_k}! \ldots n_{V_k}!} \tag{7}$$

which corresponds to the specification of the ranks for each numerical variable and the specification of the values for each categorical variable owing to a multinomial model.

The number of cells $G = \prod_{k=1}^{K} J_k$ of a multivariate data grid grows exponentially with the number of selected variables. For example, for a number of selected variables $K_s \approx \log_2 N$, the resulting data grid contains $G \geq 2^{K_s} \approx N$ cells with an average number of instance per cell below one. The description length for the multivariate correlation model becomes a limiting factor. The variable selection determines a subset of correlated variables, which joint description is shorter owing to the unsupervised data grid model. Unselected variables are described independently like in Formula 7.

5 Optimization algorithm

The evaluation criterion can be decomposed as a sum of terms related to the grid, the variables, the parts and the cells according to

$$c(M) = c^{(G)}(\mathcal{J}) + \sum_{k=1}^{K} c^{(V)}(Y_k, J_k) + \sum_{k=1}^{K} \sum_{j_k=1}^{J_k} c^{(P)}(P_{j_k}) + \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \cdots \sum_{j_k=1}^{J_K} c^{(C)}(C_{j_1 j_2 \ldots j_K})$$

where

- the grid criterion $c^{(G)}(\mathcal{J})$ relies only on the sizes $\mathcal{J} = \{J_1, J_2, \ldots, J_K\}$ of the univariate partitions of the data grid and on global features of the data sample,
- the variable criterion $c^{(V)}(Y_k, J_k)$ relies only on features of the variable Y_k and on the number of parts J_k of its partition,
- the part criterion $c^{(P)}(P_{j_k})$ for each part P_{j_k} of the univariate partition of the variable Y_k relies only on features of the part,
- the cell criterion $c^{(V)}(C_{j_1 j_2 \ldots j_K})$ for each cell $C_{j_1 j_2 \ldots j_K}$ of the data grid relies only on features of the cell, and is null for empty cells.

We adopt the algorithm described in Chapter 1 to optimize such additive criterion. The main heuristic is a greedy bottom-up heuristic, which starts from a random data grid and iteratively merges the parts as long as the criterion improves. This heuristic is enhanced with pre-processing and post-processing optimization steps, and embedded into a meta-heuristic which repeats the optimization starting from different random solutions.

The main loop of this algorithm has a time complexity of

$$O(KN\sqrt{N}\log N \max(K, \log N)),$$

where N is the number of instances and K the number of variables. The algorithm can be used as an anytime algorithm: the more time you spend, the better the solution.

6 Coclustering of Instances and Variables

We first introduce the application of unsupervised data grids to the coclustering problem, then describe how to build a classifier on the basis of coclustering.

6.1 Coclustering

A coclustering [Har72] is the simultaneous clustering of the rows and columns of a matrix. In case of binary sparse datasets, coclustering is an appealing data

preparation technique to identify correlation between clusters of instances and clusters of variables. Let us notice that continuous variables can be transformed into binary variables according to whether their value is null or non null.

Let us consider a sparse binary dataset with N instances, K variables and V non-null values. A sparse dataset can be represented in tabular format, with two columns and V rows. This corresponds to a new *dataset* with two *variables* named "Instance ID" and "Variable ID" where each *instance* is a couple of values (Instance ID, Variable ID), like in Figure 6.

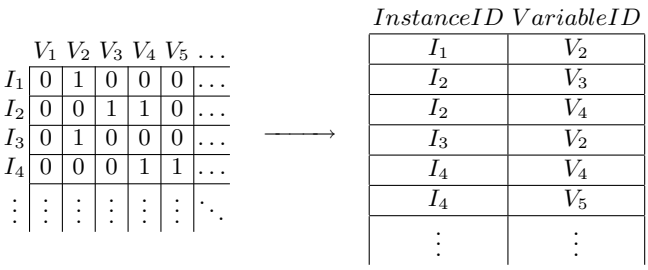


Fig. 6. Sparse binary dataset: from the sparse (instances x variables) table to the dense bivariate representation.

Bivariate unsupervised data grid models are applied to form groups of instances IDs and groups of variable IDs, so as to maximize the correlation between instances and variables. We expect to find "natural" patterns both in the space of instances and in the space of variables. It is noteworthy that the clusters retrieved by data grid models are non-overlapping, since they form a partition of the whole dataset.

6.2 Application to Supervised Learning

We apply a semi-supervised learning approach [CSZ06] to exploit all the data from the train, validation and test datasets. In a first step, all the instances are processed without any output label to identify the "natural" clusters of instances owing to the data grid coclustering technique. In a second step, the available labeled instances are used to describe the output distribution in each cluster of instances. The label of a test instance is then predicted according to the output distribution of its cluster.

Preprocessing the data with semi-supervised coclustering makes sense under the assumption that the "natural" clusters are correlated with the output values (predefined clusters). We expect that this assumption is true for some datasets, especially in the pattern recognition domain.

7 Evaluation of Data Grid Models

In the section, we evaluate the unsupervised data grid models using the datasets of the agnostic learning vs. prior knowledge challenge [Guy07,GSDC07]. We first illustrate the interest of bivariate and multivariate unsupervised data grid models for data exploration. We then report the results obtained with our coclustering method in the supervised context of the challenge.

7.1 The Agnostic Learning vs. Prior Knowledge Challenge Datasets

The purpose of the challenge is to assess the real added value of prior domain knowledge in supervised learning tasks. Five datasets coming from different domains are selected to evaluate the performance of agnostic classifiers vs. prior knowledge classifiers. These datasets come into two formats, as shown in Table 1. In the agnostic format, all the input variables are numerical. In the prior knowledge format, the input variables are both categorical and numerical for three datasets and have a special format in the two other datasets: chemical structure or text.

Name	Domain	Num. ex.	Prior	Agnostic Representation		
		train/valid/test	features	features	#cat.	#num.
Ada	Marketing	4147/415/41471	14	48	8	6
Gina	Handwritting reco.	3153/315/31532	784	970	0	784
Hiva	Drug discovery	3845/384/38449	Chem. struct.	1617	1617	0
Nova	Text classification	1754/175/17537	Text	16969	19616	0
Sylva	Ecology	13086/1309/130857	108	216	2	28

Table 1. Challenge datasets with the number of categorical and numerical features in our representation.

We use all the datasets in their prior format, except in the case of the Hiva dataset for which we have neither domain knowledge nor time to exploit the chemical structure. We preprocess the Nova text format by keeping all words of at least three characters, converting them to lowercase, truncating them to at most seven characters, and keeping the most frequent resulting words (≥ 8) so as to get a manageable bag-of-words representation (with less than 20000 words). In the case of the Sylva dataset, each instance is composed of two records of 54 variables belonging to the same class. We replace each subset (per record) of 40 binary SoilType variables by one single categorical variable with 40 values. The resulting dataset contains only 30 variables instead of 108.

In our experiments, we use the datasets with small numbers of variables (Ada and Sylva) to illustrate the interest of data grid for bivariate and multivariate correlation analysis. We consider the three other datasets (Gina, Hiva and Nova) as sparse binary datasets to conduct an instances*variables coclustering analysis.

In the case of the Gina dataset, the binary representation is obtained by replacing each non zero value by 1.

7.2 Bivariate Analysis

In this section, we show how unsupervised bivariate data grid models allow to perform correlation analysis and density estimation. Using all the unlabeled instances (train+valid+test), we compute the best bivariate data grid model for each pair of input variables. We first introduce a normalized indicator to evaluate each pair, then illustrate the bivariate analysis on the Ada and Sylva datasets.

Compression To compare the pairs of variables by decreasing correlation, we propose to use the evaluation criterion $c(M)$ given in Formula 6 (restricted to two variables for bivariate correlation analysis). This criterion is related to the probability that an unsupervised data grid model M explains the variables jointly. In order to provide a normalized indicator, we consider the following transformation of $c(M)$:

$$g\left(M\right) = 1 - \frac{c\left(M\right)}{c\left(M_{\emptyset}\right)}, \tag{8}$$

where $M_{\emptyset}$ is the null data grid model, which explains each variable independently (see Section 4.2). The indicator $g\left(M\right)$ can be interpreted as a compression gain, since negative log of probabilities are no other than coding lengths [Sha48]. The compression gain $g(M)$ holds its values between 0 and 1, since the null model is always considered in our optimization algorithm. It has value 0 for the null model and is maximal when the best possible correlation between the variables is achieved.

Ada Results For each of the $91 = 14 * 13/2$ pairs of input variables of the Ada dataset, we compute the best data bivariate data grid model using the 46033 unlabeled instances. Table 2 reports the ten most correlated pairs of variables, with the data grid size (number of parts per variable and number of non empty cells).

According to this analysis, the most correlated variables are maritalStatus and relationship, which are two categorical variables. The correlation is illustrated in Figure 7 owing to a bivariate histogram, which exhibits hight densities for some pairs of values. The second most correlated variables are education (categorical) and educationNum (numerical). It turns out that the education variable is a label related to the number of year of education. This is detected by the data grid, which is diagonal one: each on 16 non empty cells is related to a singleton group for the education variable (from preschool to doctorate) and to one elementary interval (from 1 to 16 years of education). The redundancy between the two variables is correctly detected.

Variable 1	Variable 2	Compression	# parts 1	# parts 2	# cells
maritalStatus	relationship	0.2625	5	5	20
education	educationNum	0.1693	16	16	16
relationship	sex	0.1282	5	2	10
nativeCountry	race	0.0745	8	4	31
maritalStatus	sex	0.0598	5	2	10
education	occupation	0.0505	10	9	90
occupation	sex	0.0342	7	2	14
occupation	workclass	0.0326	13	6	73
occupation	relationship	0.0201	11	5	55
age	maritalStatus	0.0198	15	3	44

Table 2. Most correlated pairs of variables in the Ada dataset.

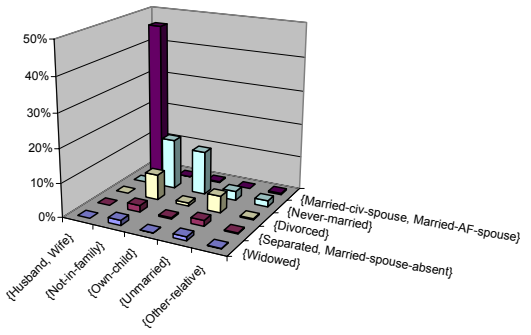


Fig. 7. Bivariate value grouping data grid for the relationship (X axis) and maritalStatus (Y axis) variables of the Ada dataset. The Z axis represents the percentage of the data sample that falls in each cell of the data grid. For example, about 45% of the instances have a husband or wife relationship and a married (group of two values) maritalStatus. The instances with the never-married marital status fall into four cells, two of which represent more than 10% of the instances (not-in-family or own-child relationship).

Sylva Results For each of the $435 = 30 * 29/2$ pairs of input variables of the Sylva dataset, we compute the best data bivariate data grid model using the 145252 unlabeled instances. Table 3 reports the most correlated pairs of variables.

As detailed in Section 7.1, each instance in the Sylva dataset is composed of two records of the same class, so that the variables come twice. It is noteworthy that our method automatically rediscovers this, by identifying the most correlated pairs in either the first or the second record. Furthermore, as can be seen in Table 3, the level of correlation (compression indicator) and the shape of the data grid is very similar for the pairs of variable belonging to each record.

Variable 1	Variable 2	Compression	# parts 1	# parts 2	# cells
Aspect_2	Hillshade9am_2	0.0559	142	35	1856
Aspect_1	Hillshade9am_1	0.0556	140	33	1754
Aspect_1	Hillshade3pm_1	0.0541	189	69	5421
Aspect_2	Hillshade3pm_2	0.0540	184	71	5722
Elevation_2	SoilType_2	0.0426	24	30	397
Elevation_1	SoilType_1	0.0422	29	22	376
Aspect_1	HillshadeNoon_1	0.0415	210	48	5574
Aspect_2	HillshadeNoon_2	0.0412	202	46	5162
Hillshade3pm_1	Hillshade9am_1	0.0404	61	58	2016
Hillshade3pm_2	Hillshade9am_2	0.0404	62	58	2109
$\text{RawahWildernessArea}_1$	SoilType_1	0.0363	2	11	19
$\text{RawahWildernessArea}_2$	SoilType_2	0.0362	2	9	16
HillshadeNoon_2	Slope_2	0.0329	57	31	1115
HillshadeNoon_1	Slope_1	0.0329	56	30	1055
Hillshade3pm_1	Slope_1	0.0302	81	29	1542
Hillshade3pm_2	Slope_2	0.0302	86	28	1608

Table 3. Most correlated pairs of variables in the Sylva dataset. It is noteworthy that each pair belong to either record_1 or record_2 of the Sylva representation.

According to Table 3, the most correlated variables are Aspect_2 and Hillshade9am_2, which are two numerical variables of the second record. The correlation is illustrated in Figure 8 owing to a bivariate diagram, which shows how the data grid models the joint density of the two variables, with intervals of varying width.

Using our non parametric approach is a clear advantage to identify and describe such complex correlation. Furthermore, since our method is fully automatic, hundreds to thousands of pairs of variables can be analyzed and sorted by decreasing correlation. This allows the data analyst to explore large datasets and focus only on the most relevant correlations.

7.3 Multivariate Analysis

In this section, we show how unsupervised multivariate data grid models allow to identify subsets of highly correlated variables. Using all the unlabeled instances (train+valid+test), we compute the best multivariate data grid model and comment the results on the Ada and Sylva datasets.

Ada Results For the Ada dataset, we get a data grid with six selected variables: education, educationNum, maritalStatus, occupation, relationship and sex. The education and educationNum variables are partitioned into 11 parts instead of 16 in the bivariate case: the numbers of years of education are grouped together for low levels (below 4 years) and higher levels (above 14 years) of education. Since the multivariate data grid model is more complex, a trade-off is found by providing a coarser grain of partition for each variable, but including more variables. The maritalStatus variable is divided into two groups of values (instead

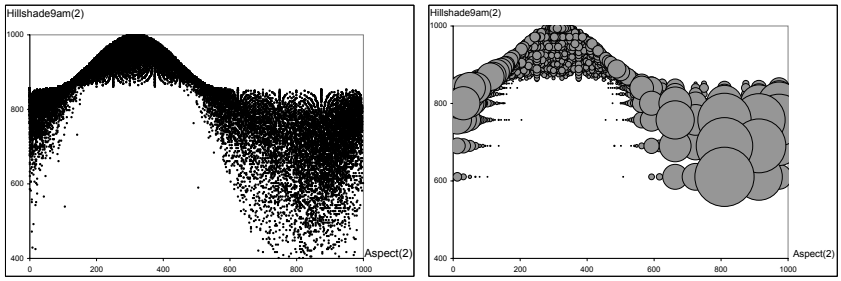


Fig. 8. Bivariate discretization data grid for the Aspect_2 (X axis) and Hillshade9am_2 (Y axis) variables of the Sylva dataset. The dispersion diagram is drawn on the left and summarized on the right by the bivariate data grid (each disk is centered on its data grid cell, with a surface proportional to the cell frequency).

education	ed.Num	maritalStatus	occupation	relationship	sex	Frequency
HS-grad	9	Married	Low	Husband	Male	4018
HS-grad	9	Not married	Medium	Other	Female	2876
HS-grad	9	Not married	Low	Other	Male	2637
Some-college	10	Not married	Medium	Other	Female	2455
Bachelors	13	Married	High	Husband	Male	1988
Some-college	10	Married	Low	Husband	Male	1653
HS-grad	9	Married	Medium	Husband	Male	1368
Some-college	10	Not married	Medium	Other	Male	1176
HS-grad	9	Not married	Medium	Other	Male	1160
Some-college	10	Not married	Low	Other	Male	1137
Bachelors	13	Not married	High	Other	Female	1067
Some-college	10	Married	Medium	Husband	Male	1056

Table 4. Most frequent cells in the best data grid summarizing the Ada dataset.

of five): married versus not married. The occupation variable is partitioned into three groups (instead of nine):

Low: Craft-repair, Machine-op-inspct, Transport-moving, Handlers-cleaners, Farming-fishing, Protective-serv, Armed-Forces,
Medium: Adm-clerical, Sales, Other-service, Tech-support, Priv-house-serv,
High: Prof-specialty, Exec-managerial.

The relationship variable is partitioned into three groups (instead of five): husband, wife, and others. It is noteworthy that the two values husband and wife were grouped together in the bivariate data grid relationship vs. maritalStatus pictured in Figure 7. In the multivariate data grid, these two values are separated so as to capture the correlation with the sex variable (male, female).

We can notice that these six variables are involved in the most correlated pairs of variables, as shown in Table 2. The multivariate data grid contains

$6534 = 11 * 11 * 2 * 3 * 3 * 2$ cells, but only 181 of them are non empty; which indicates high density cells. Table 4 describe the twelve most frequent cells of the data grid, which amount to about 50% of the instances of the dataset. Each cell can be interpreted as a cluster, which is conceptually described by an association rule [AIS93]. For example, the most frequent cell is described by Rule 1, with a support of 4018 instances.

Rule 1: education = HS-grad
educationNum = 9
maritalStatus ∈ Married = {Married-civ-spouse, Married-AF-spouse}
occupation ∈ Low = {Craft-repair, Machine-op-inspct, Transport-moving,...}
relationship = Husband
sex = Male

Sylva Results For the Sylva dataset containing 145252 instances and 30 variables, we retrieve a data grid with five numerical variables, Aspect_2, HillshadeNoon_2, Hillshade3pm_2, Hillshade9am_2 and Slope_2, discretized respectively with 14, 10, 10, 10 and 6 intervals. These five variables all belong to record_2 of the Sylva representation, which is consistent with the dataset specifications. They jointly form a subset of highly correlated variables, which confirms the bivariate analysis summarized in Table 3. The multivariate data grid contains $84000 = 14*10*10*10*6$ cells, but less than one percent (only 710) of them are non empty and the 90 most frequent cells summarize more than 50% of the dataset instances. This means that the five dimensional manifold is well approximated by the data grid, which discretizes the representation space so as to identify hight density regions.

Interestingly, the second most probable data grid identified by our search heuristic is related to the fives variables Aspect_1, HillshadeNoon_1, Hillshade3pm_1, Hillshade9am_1 and Slope_1. This is exactly the same subset of correlated variables, in record_1 of the representation instead of record_2 for the MAP data grid.

The multivariate data grids retrieved by our method can be visualized using a scatter plot matrix. Such technique presents all the pairwise scatter plots between the variables selected in the data grid, so as to analyze all the pairwise interactions. Since our method is able to automatically detect subsets of highly correlated variables, it is an efficient way of preprocessing the set of all variables in order to feed the visualization techniques with informative subsets of variables.

7.4 Coclustering Analysis

We apply the semi-supervised coclustering method introduced in Section 6 on the Gina, Hiva and Nova datasets, using the representation presented in Section 7.1.

Coclustering Results The coclustering method exploits all the available unlabeled data to represent the initial binary matrix (instances x variables) which is potentially sparse into a denser matrix with clusters of instances related to

clusters of variables. It is noteworthy that the space of coclustering models is very large. For example, in the case of the Nova dataset, the number of ways of partitioning both the text and the words, based on the Bell number, is greater than to 10^{120000}. To obtain the best possible coclusterings according to our MAP approach, we allocated several days of computation time to our anytime optimization heuristic.

Dataset	Initial representation				Coclustering representation			
	Inst.	Var.	Size	Sparseness	Inst. cl.	Var. cl.	Size	Sparseness
Gina	35000	784	$2.74\ 10^7$	19.2%	480	125	$6.00\ 10^4$	79.1%
Hiva	42673	1617	$6.90\ 10^7$	9.1%	1230	210	$2.58\ 10^5$	52.2%
Nova	17537	19616	$3.44\ 10^8$	0.6%	207	1058	$2.19\ 10^5$	84.3%

Table 5. Properties of the (instances*variables) matrix for the Gina, Hiva and Nova datasets, in their initial and coclustering representation.

In Table 5, we recall the properties of each dataset in its initial representation and present its preprocessed representation after the coclustering. The datasets are initially represented using very larges matrices, with up to hundreds of millions of cells. Their sparseness vary according to the dataset from less than 1% to about 20%. The number of their non-null elements (one variable activated for one instance) is about five millions for Gina, six millions for Hiva and two millions for Nova. Once the coclustering is performed, we get dense representations with numbers of cells reduced by a one hundred to one thousand factor.

Impact on Supervised Classification In order to evaluate the quality of the representation, and assuming that the "natural clusters" of instances in the unsupervised context are correlated with the labels in the supervised context, we train classifiers using the train and validation labeled instances to learn the distribution of the labels in each cluster of instances. In the case where a test instance belongs to a cluster with no labeled instance, we iteratively merge this unlabeled cluster so as to keep the coclustering evaluation criterion as low as possible, until at least one labeled cluster is encountered.

Dataset	Prior track		Agnostic track		Coclustering
	Winner	Best BER	Winner	Best BER	BER
Gina	Vladimir Nikulin	0.0226	Roman Lutz	0.0339	0.0516
Hiva	Chloé Azencott	0.2693	Vojtech Franc	0.2827	0.3127
Nova	Jorge Sueiras	0.0659	Mehreen Saeed	0.0456	0.0370

Table 6. Best challenge results vs. our coclustering method results for the Gina, Hiva and Nova datasets.

We recall in Table 6 the BER results of the challenge winner in the agnostic and prior track [GSDC07], and present our results obtained with the semi-supervised coclustering method (submission named "Data Grid(Coclustering)", dated 2007-02-27 for Gina and Hiva and 2007-09-19 for Nova). The results show that the supervised coclustering method obtains good predictive performance, competitive with that of most of the challenge participants and not far from that of the top results. In the case of the Nova dataset, the predictive performance significantly outperforms that of the winners, which is remarkable since our clusters were learnt without using any class label.

Impact on Interpretation The assumption that the "natural" patterns identified owing to coclustering are correlated with the classes looks true in the challenge datasets. Since we obtain many more patterns than classes, it is interesting to provide an interpretation of our coclusters.

The Gina dataset comes from the MNIST dataset [LC98]. The task, which is handwritten digit recognition, consists in predicting the value of a digit from an image representation of $28 * 28$ pixels. The coclustering method identifies about one hundred clusters of pixels (regions) and five hundred clusters of images ("natural" shapes), each of them distributed similarly on the regions.

In the case of the Hiva, further investigation with a domain specialist would be necessary to understand the meaning of the clusters of instances and variables.

The Nova dataset comes from the 20-Newsgroup dataset [Mit99]. The original task is to classify the texts into 20 classes (atheism, graphics, forsale, autos, motorcycles, baseball, hockey, crypt, electronics, med, space, religion.christian, politics.guns, politics.mideast, politics.misc, religion.misc). In the challenge, the classification task was a binary one, with two groups of classes (politics or religion vs. others). The coclustering method identifies about one thousand of clusters of words (vocabulary themes) and two hundred clusters of texts ("natural" topics), each of them distributed similarly on the themes.

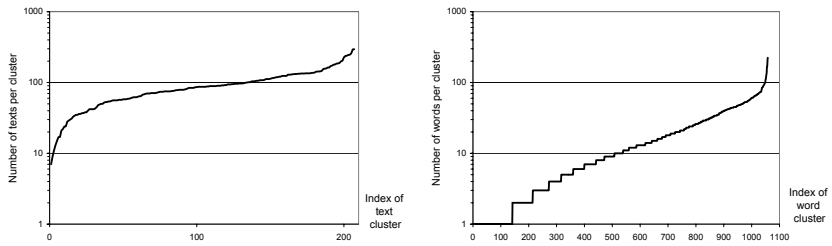


Fig. 9. Distribution of the sizes of the clusters of texts and words in the Nova dataset.

The distribution of the 17537 texts in the 207 clusters of texts (topics) is reasonably balanced. On the opposite, the repartition of the 19616 words in the

1058 clusters of words (themes) is not at all balanced, as shown in Figure 9. About 150 themes are singletons, like for example *the*, *and*, *for*, *that*, *this*, *have*, *you*. These are frequent words with low semantic, and even slightly different distribution of the topics on these singleton themes are significant and might be helpful for classification. For example, observing one of the singleton themes *say*, *why* or *who* approximately doubles the conditional probability of being in the challenge positive class (politics or religion).

A correlation study between the themes and the 20 original labels available on the train dataset reveals that the most informative themes are:

- *hockey, playoff, nhl, penguin, devils, pens, leafs, bruins, islande, goalie, mario, puck,...*
- *team, season, league, fans, teams, rangers, detroit, montrea, wins, scored, coach,...*
- *clipper, encrypt, nsa, escrow, pgp, crypto, wiretap, privacy, cryptog, denning,...*
- *dod, bike, motorcy, ride, riding, bikes, ama, rider, helmet, yamaha, harley, moto,...*
- *basebal, sox, jays, giants, mets, phillie, indians, cubs, yankees, stadium, cardina,...*
- *bible, scriptu, teachin, biblica, passage, theolog, prophet, spiritu, testame, revelat,...*
- *christi, beliefs, loving, rejecti, obedien, desires, buddhis, deity, strive, healed,...*
- *windows, dos, apps, exe, novell, ini, borland, ver, lan, desqvie, tsr, workgro, sdk,...*
- *pitcher, braves, pitch, pitchin, hitter, inning, pitched, pitches, innings, catcher,...*
- *car, cars, engine, auto, automob, mileage, autos, cactus, pickup, alarm, sunroof,...*

About one third of the theme are detected as informative with respect to the original labels. The partition of the words is very fine grained, so that many themes are hard to interpret, whereas other ones clearly capture semantics, such as:

- *book, books, learnin, deals, booksto, encyclo, titled, songs, helper*
- *cause, caused, causes, occur, occurs, causing, persist, excessi, occurin*
- *importa, extreme, careful, essenti, somewha, adequat*
- *morning, yesterd, sunday, friday, tuesday, saturda, monday, wednesd, thursda,...*
- *receive, sent, placed, returne, receivi, sends, resume*

Overall, our coclustering preprocessing method is able to produce a precise and reliable summary of the corpus of texts, which is demonstrated by the very good classification performance reported in Table 6.

8 Conclusion

The data grid models introduced in this paper are based on a partitioning model of each variables, in intervals for numerical variables and in groups of values for categorical variables. The cross-product of the univariate partitions, called a data grid, allows to quantify the joint density between the variables.

We have shown that this technique apply to a variety of tasks of data exploration, such correlation study, density estimation, visualization or association rule mining. We have also demonstrated that unsupervised data grid models are able to produce coclusterings of instances of variables, with valuable insights on the data and striking performance obtained on challenge datasets.

Like in the supervised case introduced in Chapter 1, we observed empirically that the number G of data grid cells is always below the number N of instances. This means that the number K_s of selected variables, each consisting of at least two parts, is always belows $\log_2 N$. The best subsets of correlated variables discovered by our method summarize one aspect of the dataset and estimate the joint density up to maximum number of variables. Our method automatically finds a trade-off between precision and reliability by focusing on a variable subspace, which illustrates how it manages the curse of dimensionality.

To overcome this limitation related to the number of selected variables, we plan in future work to explore two complementary approaches. In the first one, we intend to exploit the posterior distribution of data grid models to estimate the joint density on the whole variable space owing to an ensemble method. In the second one, our objective is to better approximate the whole multivariate probability distribution by exploiting the naive Bayes independence assumption [LIT92] or its relaxed extensions like in semi-naive Bayesian classifiers [Kon91] or in Bayesian network classifiers [FG96].

References

[AIS93] R. Agrawal, T. Imielinski, and A.N. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD conference on management of data*, pages 207–216, Washington, D.C., 1993.

[AS70] M. Abramowitz and I. Stegun. *Handbook of mathematical functions*. Dover Publications Inc., New York, 1970.

[Bou05] M. Boullé. A Bayes optimal approach for partitioning the values of categorical attributes. *Journal of Machine Learning Research*, 6:1431–1452, 2005.

[CCK+00] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth. *CRISP-DM 1.0 : step-by-step data mining guide*, 2000.

[CL03] J. Connor-Linton. Chi square tutorial, 2003. http://www.georgetown.edu/-faculty/ballc/webtools/web_chi_tut.html.

[Coc54] W.G. Cochran. Some methods for strengthening the common chi-squared tests. *Biometrics*, 10(4):417–451, 1954.

[CSZ06] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. (in press).

[FG96] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning bayesian networks. In *International Conference on Machine Learning*, pages 157–165, 1996.

[Fis36] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7:179–188, 1936.

[GSDC07] I. Guyon, A.R. Saffari, G. Dror, and G. Cawley. Agnostic learning vs. prior knowledge challenge. In *International Joint Conference on Neural Networks*, 2007.

[Guy07] I. Guyon. Agnostic learning vs. prior knowledge challenge, 2007. http://clopinet.com/isabelle/Projects/agnostic/.

[Har72] J.A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

[Kon91] I. Kononenko. Semi-naive Bayesian classifier. In Y. Kodrato, editor, *Sixth European Working Session on Learning (EWSL91)*, volume 482 of *LNAI*, pages 206–219. Springer, 1991.

[LC98] Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1998. http://yann.lecun.com/exdb/mnist/.

[LIT92] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In *10th national conference on Artificial Intelligence*, pages 223–228. AAAI Press, 1992.

[Mam06] R. Mamdouh. *Data Preparation for Data Mining Using SAS*. Morgan Kaufmann Publishers, 2006.

[Mit99] T.M. Mitchell. The 20 newsgroup dataset, 1999. http://kdd.ics.uci.edu/-databases/20newsgroups/20newsgroups.html.

[Pyl99] D. Pyle. *Data preparation for data mining*. Morgan Kaufmann Publishers, Inc. San Francisco, USA, 1999.

[Ris78] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[Sha48] C.E. Shannon. A mathematical theory of communication. Technical report, Bell systems technical journal, 1948.