Classification tree algorithm for grouped variables

A. Poterie [*], J.-F. Dupuy [⋆], V. Monbet [†] and L. Rouvière [†]

Abstract. We consider the problem of predicting a categorical variable based on groups of inputs. Some methods have already been proposed to elaborate classification rules based on groups of variables (e.g. group lasso for logistic regression). However, to our knowledge, no tree-based approach has been proposed to tackle this issue. Here, we propose the Tree Penalized Linear Discriminant Analysis algorithm (TPLDA), a new-tree based approach which constructs a classification rule based on groups of variables. It consists in splitting a node by repeatedly selecting a group and then applying a regularized linear discriminant analysis based on this group. This process is repeated until some stopping criterion is satisfied. A pruning strategy is proposed to select an optimal tree. Compared to the existing multivariate classification tree methods, the proposed method is computationally less demanding and the resulting trees are more easily interpretable. Furthermore, TPLDA automatically provides a measure of importance for each group of variables. This score allows to rank groups of variables with respect to their ability to predict the response and can also be used to perform group variable selection. The good performances of the proposed algorithm and its interest in terms of prediction accuracy, interpretation and group variable selection are loud and compared to alternative reference methods through simulations and applications on real datasets.

Keyword. Supervised classification, groups of inputs, group variable selection, multivariate classification tree algorithms, group importance measure, regularized linear discriminant analysis.

1 Introduction

Consider the supervised classification setting where the problem consists in predicting a class variable Y taking values in $\{1, \ldots, K\}$, with $K \geq 2$, based on a vector $\mathbf{X}$ which takes values in $\mathbb{R}^d$. Suppose further that the inputs are divided into J different groups. In many supervised classification problems, inputs can have a group structure or groups of inputs can be defined to capture the underlying input associations. In these cases, the study of groups of variables can make more sense than the study of inputs taken individually. For example, in the analysis of gene expression data, datasets contain the expression levels of thousands genes in a much smaller number of observations. Then it has become frequent to use in the analysis only a small number of genes which can be clustered into several groups that represent putative biological processes (Tamayo et al., 2007; Lee & Batzoglou, 2003).

[*]Univ Rennes, INSA, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France.
[†]Univ Rennes, CNRS, IRMAR - UMR 6625, F-35000 Rennes, France.

Another example is functional data, like spectrometry data, where researchers are often more interested by identifying discriminatory parts of the curve rather than individual wave lengths (Picheny et al., 2016). Finally, to handle with categorical variables, usual tree-based methods first need to convert each of them into dummy variables. Then, the resulting decision rules are based on the dummy variables taken individually, i.e. based on the individual modalities and not on the whole categorical variable. In all these situations, elaborating a classification rule based on groups of inputs rather than on the individual variables can improve both interpretation and prediction accuracy (Gregorutti et al., 2015). Several methods have already been proposed to deal with this problem. For instance, the logistic regression regularized by the Group Lasso penalty (GL) enables to elaborate classification rules based on groups of input variables (Meier et al., 2008). As far as we know, this problem has not been studied for classification trees.

Tree-based methods are popular in statistical data classification (Genuer & Poggi, 2017; Loh, 2014). Classification tree algorithms elaborate classification rules by means of recursive partitioning of the data space. Starting with all the data, these algorithms partition the data space into two or more regions, also called nodes, and repeat the splitting procedure on the resulting nodes. The splitting process is applied on each resulting node until some stopping criteria are achieved or as long as the node is not pure (i.e. all observations in the node do not have the same label). Each split is defined according to the values of one or more inputs. The choice of the optimal split is generally based on the maximization of the change in an impurity function: at each step, the algorithm splits the data space into more and more pure nodes. The terminal nodes, which are not split, are called leaves. At the end of the splitting process, the leaves define a partition of the data space which can be represented as a tree. A classification rule is associated to each leaf. In a leaf, observations are assigned to the most-represented class label in the leaf. Generally, the tree resulting from the splitting process is often not optimal with respect to a given criterion. So, a pruning method is often used to select an optimal tree (Breiman et al., 1984).

The first comprehensive study about classification tree algorithms was presented by Breiman et al. (1984), who introduced the popular CART algorithm. Since then, other classification tree algorithms have been developed, such as ID3 (Quinlan, 1986) and C4.5 (Quinlan, 1993). All these algorithms are univariate classification tree algorithms, that is, each node is determined according to the value of one single input. Multivariate classification trees algorithms that split each node according to the value of a subset of input variables, have also been studied. For most of the multivariate classification algorithms, splits are defined according to the value of a linear combination of a subset of input variables (Breiman et al. 1984, Wickramarachchi et al. 2016, Murthy et al. 1993, Wei-Yin Loh 1988, Li et al. 2003). Multivariate classification tree algorithms generally have higher accuracy and lead to smaller trees than univariate classification tree algorithms (Brodley & Utgoff, 1995; Lim et al., 2000). However, they suffer from two major drawbacks. First of all, they are generally time-consuming (Breiman et al., 1984; Li et al., 2003). Secondly, the subset of

input variables used to define a split is automatically selected by the algorithm with respect to an impurity criterion and without regarding if the combination of this subset of selected variables make sense. Consequently, some splits may not make sense. Thus, multivariate classification trees are often difficult to interpret.

As mentioned previously, in many supervised classification problems, input variables can have a known group structure. In this context, as far as we know, no multivariate classification tree algorithm enables to take account of this group structure. This led us to develop the Tree Penalized Linear Discriminant Analysis algorithm (TPLDA), a new multivariate classification tree algorithm involving linear splits and well adapted to grouped inputs. In this new tree-based approach, to split a node, the algorithm first estimates a split for each group of variables by performing the regularized linear discriminant analysis proposed by Witten & Tibshirani (2011). Next, the algorithm selects the optimal split with respect to an impurity criterion. This splitting procedure is then repeated until predetermined stopping criteria are satisfied. This results in a fully-grown tree which can be prone to overfitting. Thus, a pruning strategy is proposed to select an optimal tree. When the group structure is known, this new multivariate classification tree algorithm overcomes the two major drawbacks of the other multivariate classification tree algorithms. First, as the proposed algorithm takes into account the known group structure, it is less time-consuming than classical multivariate classification tree algorithms because the algorithm does not need to perform a greedy search to determine the input groups. Secondly, interpretation is easy because the algorithm uses the group structure which makes sense. Furthermore, as identification of relevant groups of inputs is also an important issue in many classification problems involving groups of variables, we introduce a measure of group importance. This score is based on a TPLDA tree and allows to rank all the groups of inputs according to their discriminatory power.

Our approach is quite similar to Mola & Siciliano (2002) who introduce a tree-based method to deal with regression problems involving groups of correlated inputs. Indeed, the method proposed by Mola & Siciliano (2002) uses a recursive two-step splitting process. First, linear discriminant analyses are performed to define a split for each group and next the algorithm uses an impurity criterion to select the best split. Nonetheless, in this work, Mola & Siciliano (2002) do not discuss the overfitting problem and do not provide any measure of importance of the groups of inputs. Moreover, unlike our proposed approach, their method uses non-regularized linear discriminant analysis to split a node.

To simplify matters, in this paper, we restrict our attention to binary classification problems, which already captures many of the main features of more general problems. Nonetheless, our algorithm can also be applied on classification problems involving more than two classes. Indeed, the splitting process allows to split a node into as many nodes as there are classes.

The paper is organized as follows. Section 2 describes the TPLDA algorithm and the group importance measure. In Section 3, performances of the proposed algorithm

are analyzed through a detailed simulation study. TPLDA is compared to CART and GL, which is one of the reference methods to elaborate classification rules with groups of inputs. In Section 4, TPLDA is applied on three publicly available real microarray datasets. The proposed method is then compared to CART, GL and the shrunken centroid regularized discriminant analysis (SCRDA) (Guo et al., 2006) which is one of the standard methods used to analyze microarray data. The time complexity of TPLDA and additional information about the simulation study and the application on the three microarray datasets are provided in Appendix. The method has been implemented in R language. The functions are available at `https://github.com/apoterie/TPLDA`.

2 The Penalized Tree Group algorithm

Let $(\mathbf{X}, Y)$ be a random vector taking values in $\mathcal{X} \times \{0, 1\}$, where $\mathbf{X} = (X_1, \ldots, X_d)$ is a vector of input variables with $\mathcal{X} = \mathbb{R}^d$ and Y is the class label. Let $\{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{n+m}, Y_{n+m})\}$ be independent copies of $(\mathbf{X}, Y)$, which are randomly split into a training set $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$ of size n and a validation set $\mathcal{T}_m = \{(\mathbf{X}_{n+1}, Y_{n+1}), \ldots, (\mathbf{X}_{n+m}, Y_{n+m})\}$ of size m. A discrimination rule is a measurable function $\hat{g} : \mathbb{R}^d \times (\mathbb{R}^d \times \{0, 1\})^{n+m} \to \{0, 1\}$ which classifies a new observation $\mathbf{x} \in \mathbb{R}^d$ into the class $\hat{g}(\mathbf{x}, (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{n+m}, Y_{n+m}))$. In what follows, we will write $\hat{g}(\mathbf{x})$ for the sake of convenience.

In this work, we consider the situation where $\mathbf{X}$ is structured into J known groups. For any $j = 1, \ldots, J$, let $\mathbf{X}^j$ denote the j-th group of size d_j, such that:

$$\mathbf{X}^j = (X_{j_1}, X_{j_2}, \ldots, X_{j_{d_j}}).$$

To simplify matters, the J groups are ordered such that

$$\mathbf{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^J).$$

Note that the groups are not necessarily disjoint, some input variables can belong to several groups. This assumption seems realistic (see the application on gene expression data introduced in Section 4) and has already been assumed by other authors, such as for instance Jacob et al. (2009).

Then, the objective is to construct a classification rule $\hat{g}$ which takes into account the group structure. To do this we propose a new tree-based approach named the Tree Penalized Discriminant Analysis (TPLDA). This method elaborates a classification rules based on two steps. First, the algorithm builds a maximal classification tree which is next pruned. These two steps are described below. We need to introduce some notations before describing the TPLDA algorithm.

2.1 Some notations

If T is a tree, t is the general notation for a node of T and n_t is the total number of observations in t. Let k be the class label, $k = \{0, 1\}$. We denote by $R_{t,k}$ the set of observations with the label k in the node t and $|R_{t,k}| = n_{k,t}$, such that $n_{0,t}+n_{1,t} = n_t$. The class probability in the node t is estimated by its standard empirical estimate $\pi_{k,t} = \frac{n_{k,t}}{n_t}$.

For any $j = 1, \ldots, J$, let consider the group $\mathbf{X}^j$ of inputs. In t, the standard estimate of the between-class covariance matrix B_t^j of group $\mathbf{X}^j$ is given by

$$\widehat{B}_t^j = \frac{1}{n_t - 2} \sum_{k=0}^{1} n_{k,t} (\hat{\mu}_t^j - \hat{\mu}_{k,t}^j)(\hat{\mu}_t^j - \hat{\mu}_{k,t}^j)^\top, \tag{1}$$

where $\top$ stands for the transpose vector and $\hat{\mu}_{k,t}^j$ is the empirical estimate of the class mean vector of $\mathbf{X}^j$ in the node t. Furthermore, the within-class covariance matrix Σ_t^j of $\mathbf{X}^j$ is estimated by its diagonal positive estimate $\widehat{\Sigma}_t^j$ defined as

$$\widehat{\Sigma}_t^j = \text{diag}\left((\hat{\sigma}_{t,1}^j)^2, \ldots, (\hat{\sigma}_{t,d_j}^j)^2\right), \tag{2}$$

where $\hat{\sigma}_{t,\ell}^j$, with $\ell = 1, \ldots, d_j$, denotes the within-class standard deviation estimate of the ℓ-th input of $\mathbf{X}^j$. This approximated estimate of the within-covariance matrix is used to improve the robustness when estimating the splitting rules. It will be discussed in the next subsection.

2.2 Construction of a maximal tree

As for existing tree-based methods, TPLDA elaborates a maximal tree by recursively partitioning the data space. At each step, the data space is divided into smaller and smaller nodes. This splitting process, that is applied on nodes, is made of two steps. Consider the split of the node t. First, for any $j = 1, \ldots, J$, we split the input space according to a linear combination of the inputs belonging to group $\mathbf{X}^j$. Then, we select the best split with respect to an impurity criterion (which is equivalent to selecting the splitting group). These steps are now described in greater details.

- Step 1: within group PLDA.
  In the first step, the algorithm performs a penalized linear discriminant analysis (PLDA, Witten & Tibshirani, 2011) on each group $\mathbf{X}^j = (X_1^j, \ldots, X_{d_j}^j)$, with $j = 1, \ldots, J$. That is, PLDA seeks a one-dimensional projection $(\beta^j)^\top \mathbf{x}^j$, $(\beta^j = (\beta_1^j, \ldots, \beta_{d_j}^j) \in \mathbb{R}^{d_j})$, of the observations in t, that maximizes the ratio of the between-class covariance to the within-class covariance. PLDA's criterion can be defined as:

$$\max_{\beta^j \in \mathbb{R}^{d_j}} \left\{ (\beta^j)^\top \widehat{B}_t^j \, \beta^j - \lambda_j \sum_{\ell=1}^{d_j} |\widehat{\sigma}_{t,\ell}^j \beta_\ell^j| \right\} \quad \text{subject to} \quad (\beta^j)^\top \widehat{\Sigma}_t^j \, \beta^j \leq 1, \tag{3}$$

where $\widehat{B}_t^j$ and $\widehat{\Sigma}_t^j$ are respectively given by (1) and (2). As for Fisher's linear discriminant analysis (Friedman et al., 2001, FDA), the solution of (3) is denoted by $\hat{\beta}^j$ and is called the penalized discriminant vector. In (3), the use of the regularization term $\lambda_j \sum_{\ell=1}^{d_j} |\widehat{\sigma}_{t,\ell}^j \beta_\ell^j|$ and the diagonal positive within-class covariance matrix Σ_t^j enables to solve the singularity problem occurring when the number of observations in the node t is small compared to the number of variables in the group $\mathbf{X}^j$ (for more details see Witten & Tibshirani, 2011). In practice, the parameter $\lambda_j \in \mathbb{R}^+$ enables to shrink to zero some components of β^j. The value of λ_j is chosen by K-fold cross-validation. The algorithm selects among L guided values the value for λ_j which maximizes the cross-validated estimate of the decrease in impurity defined below in (5).

PLDA divides the node t into two child nodes according to the linear decision boundary described by the linear equation $\beta^{j\top}(\mathbf{x}^j - \frac{(\hat{\mu}_{1,t}^j - \hat{\mu}_{0,t}^j)}{2}) = 0$. The two child nodes of t are defined as:

$$t_0(j) = \left\{ \mathbf{x} \in t \mid \hat{\beta}^{j\top} \left(\mathbf{x}^j - \frac{(\hat{\mu}_{1,t}^j - \hat{\mu}_{0,t}^j)}{2} \right) < 0 \right\}$$
$$\text{and} \tag{4}$$
$$t_1(j) = \left\{ \mathbf{x} \in t \mid \hat{\beta}^{j\top} \left(\mathbf{x}^j - \frac{(\hat{\mu}_{1,t}^j - \hat{\mu}_{0,t}^j)}{2} \right) \geq 0 \right\}.$$

In (3), if λ_j is equal to zero and if either the inputs in group $\mathbf{X}^j$ are mutually independent or the size d_j of the j-th group is 1, then the matrix $\widehat{\Sigma}_t^j$ is reduced to the standard estimate of the within-class covariance matrix . In this case, the PLDA problem (3) is equivalent to the FDA problem. Note that FDA cannot be used here since it is not adapted to the recursive splitting of nodes that become smaller and smaller (Shao et al., 2011; Friedman, 1989; Xu et al., 2009; Bouveyron et al., 2007). This point is discussed in Apprendix C.1.

- Step 2: choosing the splitting group.
  Selection of the splitting group is based on an impurity function $\mathcal{Q}$ which measures the homogeneity of a node. Here, we use Gini impurity function, which is computed based on the training set by

$$\mathcal{Q}(t) = \pi_{1,t}(1 - \pi_{1,t}).$$

  Note that other impurity criteria, such as the information criterion (also named Entropy), could be used. The algorithm selects the splitting group $j_t^\star \in \{1, \ldots, J\}$, which maximizes the impurity decrease defined for each group $\mathbf{X}^j$, $j = 1, \ldots, J$, by

$$\Delta\mathcal{Q}(j,t) = \left[n_t \mathcal{Q}(t) - n_{t_0(j)} \mathcal{Q}(t_0(j)) - n_{t_1(j)} \mathcal{Q}(t_1(j)) \right]. \tag{5}$$

  In practice, criterion (5) may not be satisfying since it tends to foster larger groups. Indeed, the largest groups have more possible splits than the smallest

groups. As a consequence, the largest groups would be more likely optimal with respect to the impurity decrease (Strobl et al., 2007). Thus, to control this selection bias, we propose to penalize the criterion (5) by a decreasing function $\mathsf{pen}(d_j)$ of the group size d_j:

$$\Delta_p \mathcal{Q}(j,t) = \mathsf{pen}(d_j)\Delta\mathcal{Q}(j,t). \tag{6}$$

Several penalty functions can be used. We propose:

$$\begin{aligned} \mathsf{pen}(d_j) &= 1/d_j, \\ \mathsf{pen}(d_j) &= 1/\sqrt{d_j}, \\ \mathsf{pen}(d_j) &= 1/\max(\log d_j, 1). \end{aligned} \tag{7}$$

We name these functions *size penalty*, *root-size penalty* and *log-size penalty* respectively. These penalty functions are usually used to select groups of variables, see for example Huang et al. (2012), Gregorutti et al. (2015) and Grimonprez et al. (2018). The use of the corrected impurity criterion (6) and the choice of the penalty function are discussed in Section 3.

Remark 2.1.

- As for FDA and many other statistical methods, it is generally better to standardize the inputs before computing PLDA. Note that, to perform PLDAs, we use the function PenalizedLDA (included in the R package penalizedLDA) which standardizes automatically the inputs before performing the penalized linear discriminant analysis.

- The time complexity of TPLDA at a node t of size n_t is in the worst case $\mathcal{O}\left(JLKn_t d_{\max}^2\right)$ with J referring to the number of groups, K being the number of folds in the cross-validation used to tune λ_j, L denoting the number of guided values for λ^j in the cross-validation and $d_{\max} = \max_j(d_j)$ with d_j being the size of $\mathbf{X}^j$. The computation is detailed in Appendix A. As the inequality $K \leq n_t$ is always satisfied, TPLDA is less time consuming than lots of multivariate classification tree algorithms such as for instance HHCART introduced by Wickramarachchi et al. (2016) (time complexity of HHCART $= \mathcal{O}\left(n_t^2 d^3\right)$ with $d = \sum_{j=1}^{J} d_j$ is the total number of input) and OC1 introduced by Murthy et al. (1993) (time complexity of OC1 $= \mathcal{O}\left(n_t^2 \log(n_t) d\right)$), excepted in very small nodes (i.e. $Ld_{\max} > \log(n_t)$). A detailed calculation of the time complexity of HHCART and OC1 is provided by Wickramarachchi et al. (2016).

At the very beginning of the whole procedure, steps 1 and 2 are applied to partition the entire data space into two nodes. Then, these steps are repeated recursively on each node t until each one satisfies at least one of the following stopping criteria:

- t is homogeneous (or near so) with respect to a particular class, i.e.

$$\pi_{1,t} < \epsilon \quad \text{or} \quad \pi_{1,t} > 1 - \epsilon,$$

for a small given value ϵ,

- no further partition can reduce the impurity of t, that is:

$$\Delta_p \mathcal{Q}(j, t) = 0, \text{ for any } j = 1, \ldots, J.$$

By iterating the splitting process described above, we obtain a fully-grown tree denoted by $T_{\max}$. It is well known that maximal classification trees are generally not optimal with respect to any performance criterion (such as the misclassification error). Indeed, an excessively large number of nodes is prone to overfitting (Breiman et al., 1984). Thus, we propose a pruning strategy that allows to select an optimal tree. This strategy is described below.

2.3 Pruning strategy

Let T be a subtree of $T_{\max}$ and $\widetilde{T}$ be the set of $|\widetilde{T}|$ terminal nodes of T. We define the depth of node t, which is denoted by $D(t)$, as the number of conditions that an observation $\mathbf{x} \in \mathbb{R}^d$ has to satisfy from the root to the node t. The depth $D(T)$ of the tree T is then defined as:

$$D(T) = \max_{t \in \widetilde{T}} D(t).$$

Figure 1 illustrates the notions of nodes, terminal nodes and depth.

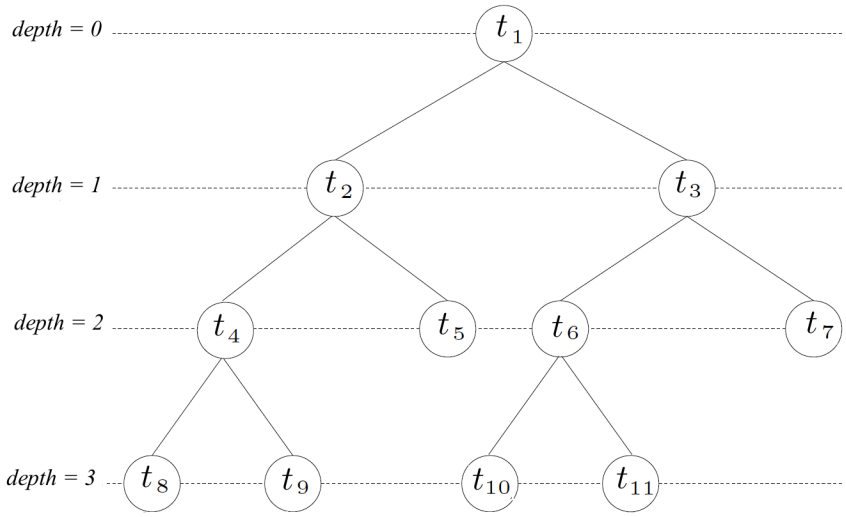


Figure 1: Example of a classification tree. Circles indicate the nodes. *depth* refers to the depth of the nodes. Here $D(T) = 3$. The terminal nodes are $\widetilde{T} = \{t_5, t_7, t_8, t_9, t_{10}, t_{11}\}$. The node t_1 denotes the tree root.

Define the sequence

$$t_1 = T_0 \subset T_1 \subset \ldots \subset T_{D(T_{\max})} = T_{\max} \tag{8}$$

of nested trees such that T_h, for any $h = 1, \ldots, D(T_{\max})$, is the subtree of $T_{\max}$ which maximizes over all subtrees $T \subset T_{\max}$ the quantity

$$\sum_{t \in \widetilde{T}} D(t) \quad \text{subject to} \quad D(t) \leq h.$$

In other words, T_h is the deeper subtree of $T_{\max}$ whose terminal nodes have a depth less than or equal to h. For example, Table 1 gives the terminal nodes for the sequence of subtrees of the tree displayed in Figure 1.

Tree	**Terminal Nodes**
T_0	t_0
T_1	t_2, t_3
T_2	t_4, t_5, t_6, t_7
T_3	$t_8, t_9, t_5, t_{10}, t_{11}, t_7$

Table 1: Terminal nodes for the subtrees in Figure 1.

Each tree T_h, $h = 1, \ldots, D(T_{\max})$, defines a classification rule $\hat{g}_h$:

$$\hat{g}_h(\mathbf{x}) = \sum_{t \in \widetilde{T}_h} \hat{y}_t \mathbb{1}_t(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \tag{9}$$

where $\mathbb{1}_t(\mathbf{x})$ is the indicator function which equals 1 if $\mathbf{x}$ falls into the leaf t and 0 otherwise, and $\hat{y}_t = \mathbb{1}_{n_{1,t} \geq n_{0,t}}$ is the most represented class in the node t. Note that the classification rules $\hat{g}_h$, $h = 1, \ldots, D(T_{\max})$, depend only on the training set $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$. The proposed pruning strategy selects the rule $\hat{g}_h$ which minimizes the misclassification error $\mathbf{P}(\hat{g}_h(\mathbf{X}) \neq Y)$. In practice, this error is estimated on the validation set $(\mathbf{X}_{n+1}, Y_{n+1}), \ldots, (\mathbf{X}_{n+m}, Y_{n+m})$. Precisely, we choose

$$\widehat{h} \;=\; \underset{h=1,\ldots,D_{T_{\max}}}{\operatorname{argmin}} \; \frac{1}{m} \sum_{i=n+1}^{n+m} \mathbb{1}_{\hat{g}_h(\mathbf{X}_i) \neq Y_i}.$$

The final tree retained by our procedure is the subtree $T_{\widehat{h}}$. The following section illustrates the TPLDA algorithm.

2.4 A toy example

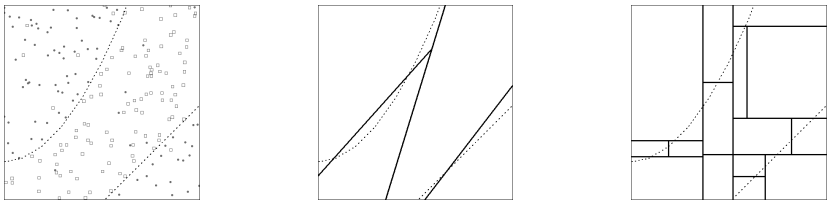
Consider the random vector $(\mathbf{X}, Y)$ with values in $\mathbb{R}^2 \times \{0, 1\}$. X_1 and X_2 are two independent random variables with distribution $\mathcal{N}(0, 1)$. The conditional distribution of Y is defined as

$$\mathcal{L}(Y \mid \mathbf{X} = \mathbf{x}) = \begin{cases} \mathcal{B}(0.9) & \text{if} \quad x_2 > 2x_1^2 + 0.20 \quad \text{or} \quad x_2 < 0.5 + x_1 \\ \mathcal{B}(0.1) & \text{otherwise.} \end{cases} \tag{10}$$

where $\mathcal{B}(\pi)$ denotes a Bernoulli distribution of parameter π. The aim is to predict the class label Y according to the unique and single group $\mathbf{X}^1 = \mathbf{X} = (X_1, X_2)$. In this scenario, the Bayes classification rule $g^*(\mathbf{x})$ is defined by:

$$g^*(\mathbf{x}) = \begin{cases} 1 & \text{if} \quad x_2 > 2x_1^2 + 0.20 \quad \text{or} \quad x_2 < 0.5 + x_1 \\ 0 & \text{otherwise.} \end{cases}$$

For TPLDA and CART, a maximal tree is first built on a training sample of 50 observations and is next pruned by using a validation set of 50 observations. TPLDA uses the proposed pruning strategy described above while CART uses the classical minimal cost-complexity pruning method (Breiman et al., 1984). Finally, the predictive performances of the two final trees have been measured by the area under the ROC curve (AUC) estimated on an independent test sample of 1000 observations. Here, TPLDA allows to elaborate a less complex partition of the input space without lost of accuracy (Figure 2). The associated trees are displayed in Appendix B (see Figure 13).



(a) (b) (c)

Figure 2: Illustration of the TPLDA method - a simple binary classification problem in $\mathbb{R}^2$. (a) 200 observations defined by model (10), (b) a TPLDA partition (AUC=0.90), (c) a CART partition (AUC=0.89). On each graph, Bayes decision boundaries are represented by the two dotted lines.

2.5 Group importance measure

In supervised classification problems involving grouped inputs, groups are seldom equally relevant. Often only a few of them are important with respect to the prediction of the response variable. The quantification of the group importance is then useful for both interpretation and performing group variable selection. TPLDA provides a measure of importance of each group. This score, which is related to a TPLDA tree, is based on the penalized splitting criterion (6). Formally, the importance of the group $\mathbf{X}^j$, $j = 1, \ldots, J$, related to a TPLDA tree T, is the sum over all non-terminal nodes of T of the *corrected* penalized impurity decrease from splitting on group j,

$$\mathcal{I}(j, T) \;=\; \sum_{t \,\in\, T \setminus \widetilde{T}} \Delta_p \mathcal{Q}(j, t) p(j, j_t^\star), \tag{11}$$

where $\Delta_p \mathcal{Q}(j, t)$ is the penalized decrease in node impurity (6) from splitting on group j, $j_t^\star$ is the index of the group selected to split the node t (see Step 2 in Section 2.2) and $p(j_t^\star, j)$ is a *correction*. The parameter $p(j, j_t^\star)$ is the empirical probability of agreement between the split of the node t based on j and the one based on $j_t^\star$. It is defined by

$$p(j, j_t^\star) \;=\; \max \left\{ p_{00}(j, j_t^\star) + p_{11}(j, j_t^\star) \,,\, p_{01}(j, j_t^\star) + p_{10}(j, j_t^\star) \right\},$$

where $p_{kk'}(j, j_t^\star)$, with $(k, k') \in \{0, 1\}^2$, is the empirical probability that the split of node t based on group j and the one based on group $j_t^\star$ send an observation in node

t both into $t_k(j)$ and $t_{k'}(j_t^\star)$. $p(j, j_t^\star)$ lies between 0 and 1 and takes the value 1 if the two splits send all observations in node t into the same child nodes. This quantity is used to prevent overestimating the importance of groups which are weakly correlated with both the relevant groups and the response variable (see chap. 5, Breiman et al., 1984).
As only the relative magnitude of this score matters, the group importance measure is normalized to a scale between 0 and 100,

$$\widetilde{\mathcal{I}}(j,T) = 100 \times \frac{\mathcal{I}(j,T)}{\max\limits_{j'=1,\ldots,J} \mathcal{I}(j',T)}. \tag{12}$$

This score induces an order of importance. Groups with the highest group importance measure are considered as important. The group importance measure is assessed in the simulation studies introduced in Section 3.

3 Evaluation of the methods by simulation studies

Several numerical experiments inspired by Friedman et al. (2001) are used to assess the performances of TPLDA. In this simulation study, the proposed method is compared to CART since the two methods are very similar when inputs are not grouped. TPLDA is also compared to GL, which is one of the reference methods to elaborate classification rules with groups of inputs. The general simulation design is described below.

3.1 Simulation design

The outcome variable Y is simulated from a Bernoulli distribution $Y \sim \mathcal{B}(0.5)$. The vector $\mathbf{X}$ of inputs is structured into J groups: $\mathbf{X} = (\mathbf{X}^1, \ldots, \mathbf{X}^J)$. Each group $\mathbf{X}^j$, $j = 1, \ldots, J$, includes d_j variables. When $Y = 0$, for any $j = 1, \ldots, J$ and any $\ell = 1, \ldots, d_j$ the component X_ℓ^j follows a standard Gaussian distribution:

$$\mathcal{L}(X_\ell^j \mid Y = 0) \;=\; \mathcal{N}(0,1).$$

When $Y = 1$, for any $j = 1, \ldots, J$ and any $\ell = 1, \ldots, d_j$ the component X_ℓ^j is defined conditionally to the value of the standard uniform random variable U:

$$\mathcal{L}(X_\ell^j \mid Y = 1, U = u) = \begin{cases} \mathcal{N}(-\mu_j, 1) & \text{if } u < u_1; \\ \mathcal{N}(\mu_j, 1) & \text{if } u_1 \leq u < u_2; \\ \mathcal{N}(0,1) & \text{otherwise.} \end{cases} \tag{13}$$

where u_1, u_2 are two fixed real numbers satisfying $0 \leq u_1 < u_2 \leq 1$. For any $j = 1, \ldots, J$, the component $\mu_j \geq 0$ can be interpreted as the discriminatory power of the group j: the higher the value of μ_j is, the more the class-conditional distributions of $\mathbf{X}^j$ differ. If $\mu_j = 0$, all inputs in group $\mathbf{X}^j$ are distributed according to

a standard Gaussian distribution, whatever the values of Y and U. In this case, the group $\mathbf{X}^j$ is not relevant to predict Y. We note $\mu = (\mu_1, \ldots, \mu_J)$.

The covariance between two inputs X_ℓ^j and $X_{\ell'}^{j'}$ ($j, j' = 1, \ldots, J$ and $\ell = 1, \ldots, d_j$ and $\ell' = 1, \ldots, p_{j'}$) is defined as:

$$\text{Cov}\left(X_\ell^j, X_{\ell'}^{j'}\right) = \begin{cases} c_w^{|\ell-\ell'|} & \text{if } j = j', \\ 0 & \text{otherwise.} \end{cases}$$

where $0 \leq c_w < 1$ and $|\ell - \ell'|$ measures the distance between two inputs belonging to a same group. Thus, in this simulation design, the group structure of the inputs comes from both the discriminatory power of the inputs defined by the vector μ and the block structure of the covariance matrix of $\mathbf{X}$. The covariance structure mimics the one of gene expression data: genes included in a same putative biological pathway are correlated and the correlation is a decreasing function of the "distance" between any two genes.

Finally, $n + m + q$ observations are generated according to this simulation model and randomly divided into three independent subsamples: a training sample of size n, a validation sample of size m and a test sample of size q.

To assess the performances of TPLDA, five experiments are considered by varying the parameters n, m and d_j, $j = 1, \ldots, J$.
In every experiment, the size of the test set is $q = 1000$ and $J = 10$ groups are simulated. The vector μ is set to $\mu = (1.25, 0, 1, 0, 0.75, 0, 0.5, 0, 0.25, 0)$. In this way, only groups with an odd index are relevant and the discriminatory power of each even group (i.e. in each relevant group) is a linear decreasing function of the group index. We choose $(u_1, u_2) = (0.25, 0.90)$ and $c_w = 0.85$.

The five considered scenarios are described below:

- **Experiment 1: ungrouped data.** Each group includes $d_j = 1$ variable and the training and the validation samples both include $n = m = 500$ observations.

- **Experiment 2: groups of equal size.** Each group includes $d_j = 10$ variable and the training and the validation samples both include $n = m = 500$ observations.

- **Experiment 3: large groups of equal size.** Each group includes $d_j = 50$ variables and the training and the validation samples both include $n = m = 100$ observations.

- **Experiment 4: inclusion of a large noisy group.** This experiment is similar to experiment 2 with the addition of a large noisy group including realizations of 50 independent standard Gaussian variables.

- **Experiment 5: inclusion of a large noisy group and of some noisy variables in the most relevant group.** This experiment is similar to experiment 4 with the addition of 10 independent standard Gaussian variables in the first group of variables (i.e. in the most relevant group of variables).

First, experiments 1, 2 and 3 are used to assess the performances of the TPLDA method in comparison with CART and GL and to evaluate the group importance measure. In these first three experiments, the Gini criterion used by the TPLDA method is not penalized, that is $\mathsf{pen}(d_j) = 1$ in (6). Next, experiments 4 and 5 are used to study the use of the penalized Gini criterion (6) when choosing the splitting group in step 2 (see Section 2.2). This aims to determine if the introduction of a penalty function in the Gini impurity criterion (6) enables to control the sensitivity of the TPLDA method to the group size. The three penalty functions defined in equation (7) are assessed.

For TPLDA, the maximal tree is built on the training set and is next pruned by applying the pruning strategy described in Section 2.3 on the validation sample. In CART, the training set is used to elaborate the maximal tree that is next pruned by using the minimal cost-complexity pruning method and the validation set. For GL, the model is elaborated on the training set and the shrinkage parameter is selected on the validation set.
A variant of TPLDA is also applied on the five experiments. In this variant, PLDA is replaced by FDA. Results, that are given in Appendix C.1, illustrate the fact that FDA is not adapted to recursively split nodes that become smaller and smaller. Moreover, in order to assess the sensitivity to the pruning method, the pruning strategy proposed in Section 2.3 is also used to prune the CART maximal tree. Results are given in Appendix C.2 and show no significant difference between methods. Moreover, CART and GL results in experiments 4 and 5 are displayed in Appendix C.3. All the results are based on the 200 samples.

3.2 Performances of TPLDA, CART and GL

In each experiment, the predictive performances of TPLDA, CART and GL are assessed and compared by the AUC on the test set. Furthermore, the complexity of the classification rule is also studied. For TPLDA and CART, this criterion is measured by using the tree depth: interpretation of a large tree is harder than the one of a small tree. For GL, the complexity of the classification rule is measured by the number of groups included in the model: the complexity increases with the number of groups included in the model.

Table 2 displays the simulation results for each assessed method. For each criterion, the median value is given followed by the values of the first and the third quartiles in brackets. The model size gives the number of groups of variables included in the final GL model. Figures 3 and 5 display group selection frequencies for TPLDA. The selection frequency of a given group is defined as the number of times that a group is included at least once in the final model. Distribution of the AUC for each method in the three experiments are displayed in Appendix C.3. Globally, TPLDA

performs well in the three scenarios. Compared to CART and GL, it elaborates more accurate and easily understandable classification rules.

	TPLDA	CART	GL
Experiment 1			
AUC	0.66 (0.65,0.68)	0.67 (0.65,0.68)	0.64 (0.63,0.66)
Tree depth	4 (3,5)	5 (3,7)	.
Model size	.	.	4 (3,7)
Experiment 2			
AUC	0.76 (0.74,0.77)	0.68 (0.66,0.7)	0.66 (0.65,0.68)
Tree depth	3 (3,4)	6 (4,8)	.
Model size	.	.	4 (3,5)
Experiment 3			
AUC	0.83 (0.7,0.85)	0.64 (0.62,0.66)	0.67 (0.64,0.69)
Tree depth	2 (2,3)	4 (2,5)	.
Model size	.	.	2 (1,4)

Table 2: Performances of the assessed methods.

In experiment 1, we highlight the similarity between TPLDA and CART when inputs are not grouped. Indeed, TPLDA selects the same input variables and has similar predictive performances as CART (Figure 3). Nonetheless, CART elaborates larger trees and tends to select less frequently the noisy groups. The two methods do not exactly give the same results since they do not use the same splitting process. To split a node, CART tries to find the splitting input and the value for this inputs that maximizes the decrease in impurity in the node (see Breiman et al., 1984). On the contrary, TPLDA first estimates a split for every group based on a maximization of the ratio of the between-class covariance matrix and the within-class covariance matrix and next selects the split that maximizes the decrease in impurity in the node (see Section 2.2).

In the second and the third experiments, input variables are grouped. In these scenarios, TPLDA outperforms the other methods. In particular, it has higher predictive performances. Besides, the final TPLDA trees are smaller than the final CART trees. This last point can be explained by the use of multivariate splits which are more informative. This leads to a quicker decreasing of the misclassification error in both the training set and the validation set and then to smaller final trees (Figures 4). Furthermore, since TPLDA splits are defined according to the groups which makes more sense that inputs taken individually, TPLDA trees are more easily interpretable than CART trees. Also, TPLDA well identifies the most relevant groups and the selection frequency of a given group behaves as an increasing

function of the discriminatory power of the group (Figures 3 and 5).

The predictive performances of TPLDA and GL seem to improve with the group size. This may be due to the fact that as in every group all inputs share the same discriminatory power, the discriminatory power of a predictive group increases when the group size increases.

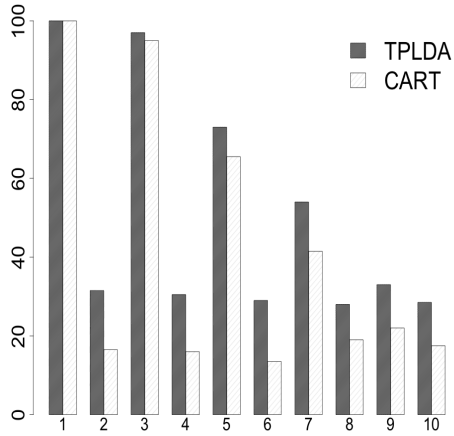


Figure 3: Group selection frequency in experiment 1 (in %).

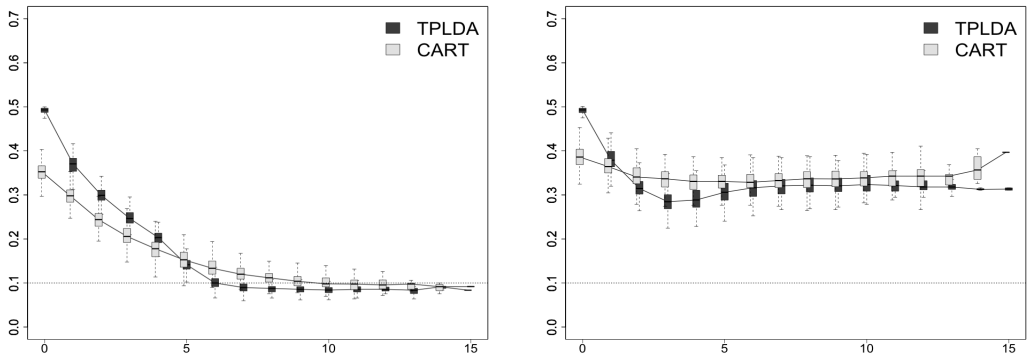


Figure 4: Misclassification error estimate according to the tree depth on the training set (left) and on the validation set (right) in experiment 2. The dotted lines denote the value of the Bayes error (Bayes error=10%).

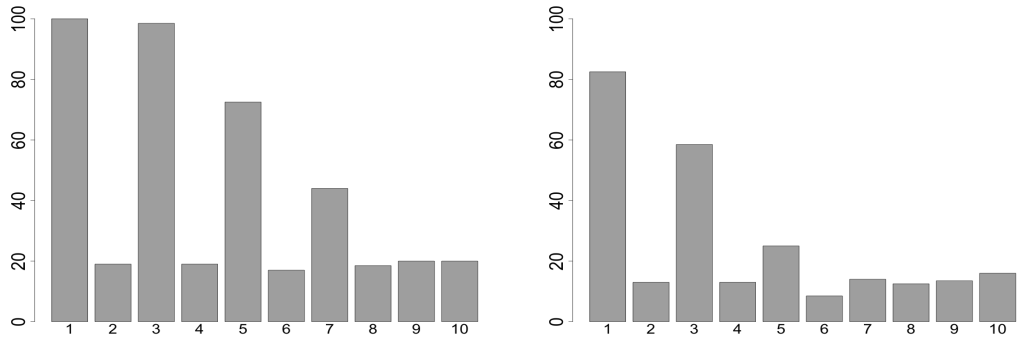


Figure 5: Group selection frequency (in %) for TPLDA in experiment 2 (left) and experiment 3 (right).

3.3 Assessment of the group importance measure

In this section, we study the performances of the group importance measure. Table 3 displays the percentage of time that the relevant groups are part of the 5 groups of inputs with the highest score of importance. The average selection frequency of the relevant groups with GL is also added, for comparison purpose. In all experiments, the TPLDA group importance measure seems to well identify the three most relevant groups. The fourth and fifth most relevant groups are less frequently identified. This may be due to the relative low discriminatory power of these groups compared to the three other relevant groups. The distribution of the group importance measure for each group in every experiment is displayed in Appendix C.3.

	TPLDA importance score	GL model
Experiment 1		
Selection rate of the 5 relevant groups	30	20.5
Selection rate of at least 3 relevant groups	100	78
Selection rate of the 3 most relevant groups	98.5	65.5
Experiment 2		
Selection rate of the 5 relevant groups	33.5	11.5
Selection rate of at least 3 relevant groups	100	79.5
Selection rate of the 3 most relevant groups	100	66.5
Experiment 3		
Selection rate of the 5 relevant groups	14	7.5
Selection rate of at least 3 relevant groups	90.5	35
Selection rate of the 3 most relevant groups	80.5	19

Table 3: Assessment of the group importance measure: top 5 groups with the highest score of importance.

3.4 Choice of the penalty function

This section investigates the use of a penalized Gini criterion (5) for controlling the sensitivity of the TPLDA method to the group size. The three penalty functions defined in (7) are evaluated. The performances of the TPLDA method when using these penalty functions are compared to the TPLDA method when using no penalty (i.e. $\mathsf{pen}(d_j) = 1$) through experiments 4 and 5. In the first one, a large irrelevant group is added. In experiment 5, there is still the large noisy group and some extra noisy input variables are also added in the most relevant group (i.e. the first group of inputs). This last experiment allows to illustrate a situation where some groups could not be well defined. More precisely, it aims to determine if the TPLDA method is still able to identify that the first group contains relevant information, even if a penalty function (7) is added to the Gini criterion (5).

Table 4 displays the performances of each assessed method in experiments 4 and 5. The median value is given followed by the values of the first and the third quartiles in brackets for each criterion. Table 5 displays the percentage of time that the relevant groups are part of the 5 input groups with the highest score of importance. Figures 6 and 7 show the distribution of the group selection for each penalty function in experiments 4 and 5. Moreover, Figures 8 and 9 display boxplots of the depth of the first node that is split by using the first group (i.e. the most relevant group) and the eleventh group (i.e. the large noisy group). Furthermore, boxplots of the group importance measure are displayed in Figures 10 and 11.

According to these two experiments, the use of a penalty function enables to control the sensitivity to the group size. Indeed, when no penalty function is used, the large noisy group is often chosen to build the tree (Figures 6 and 7). On the contrary, when using a penalty function, the large noisy group is significantly less frequently selected, and when it is, this group mainly appears at the bottom of the tree that means that the group is used to defined the last nodes, i.e. the least informative nodes of the tree (Figures 8 and 9). Furthermore, based on the results of the experiment 5, the TPLDA method is still able to identify the relevant information contained in the first group, even when adding a penalty function to the Gini impurity criterion (Figures 6 and 8). Consequently, in these scenarios, using a penalized Gini criterion allows to improve significantly the predictive performances of the classification rule (Tables 4).

As far as concerning the importance score, the use of a penalized Gini criterion seems to improve the ability of the importance score to identify the true relevant group (Table 5 and Figures 10 and 11). Nonetheless, it seems that the large noisy group still appears in the five most important groups, even when adding a penalty function to the Gini impurity criterion. This points out that the group importance score might be not reliable for estimating the importance of large noisy groups when some noisy groups are very large compared to the true relevant groups. This lack of robustness is also highlighted by the high variability of the estimate of the importance of the large noisy group (Figures 10 and 11). So, when group sizes vary

greatly, the estimated values of the importance score for the largest groups should be considered with caution.

Moreover, in these scenarios, the three penalty functions give similar results in terms of predictive performances and group selection frequency. Neither penalty function is preferable, they all seem adapted. Generally, the choice of the penalty function is highly dependent on the data. Indeed, if it is expected that the noise is mostly included in the largest groups which are much larger than the supposed relevant groups, then the penalty $\mathsf{pen}(d_j) = 1/d_j$ would be preferable. Otherwise, this penalty function may appear too strong and other penalties such as $\mathsf{pen}(d_j) = 1/\sqrt{d_j}$ or $\mathsf{pen}(d_j) = 1/\max(\log d_j, 1)$ may perform better. Note that if all groups have equal size, as for instance in the first three experiments, there is no need to use a penalty function.

Penalty	1	$1/d_j$	$1/\sqrt{d_j}$	$1/\max(\log d_j, 1)$
Exp. 4				
AUC	0.65 (0.6,0.7)	0.75 (0.74,0.77)	0.73 (0.71,0.75)	0.72 (0.68,0.74)
Tree depth	2 (2,4)	3 (3,4)	3 (2,3)	2 (2,3)
Exp. 5				
AUC	0.66 (0.6,0.71)	0.73 (0.7,0.75)	0.72 (0.68,0.74)	0.71 (0.65,0.73)
Tree depth	2 (2,3)	3 (3,4)	3 (2,3)	2 (2,3)

Table 4: Sensitivity to the choice of the penalty function pen: performances of TPLDA according to the penalty function.

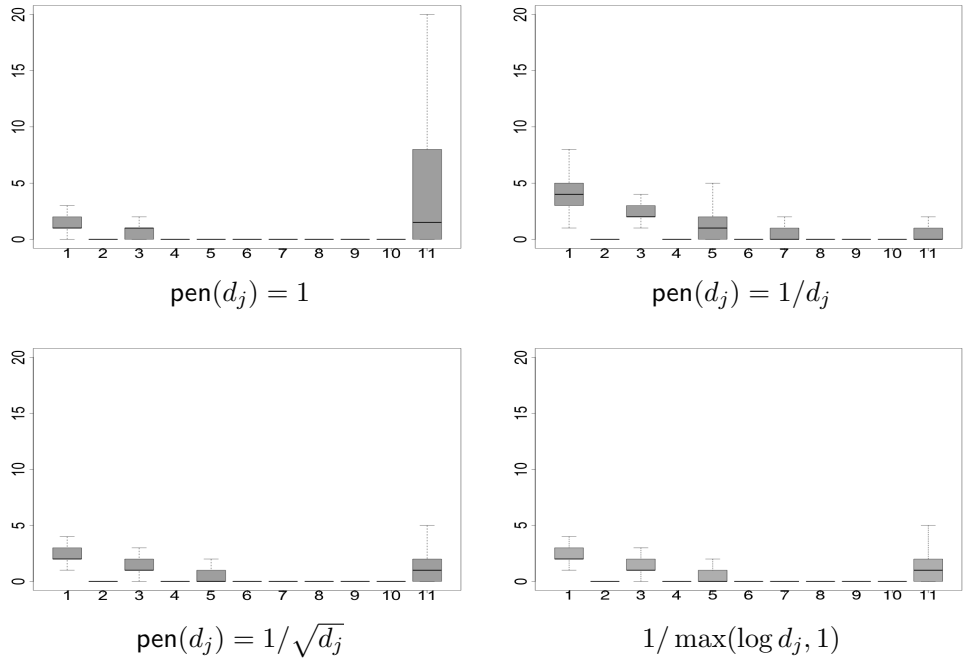


Figure 6: Sensitivity to the choice of the penalty function pen: group selection for TPLDA according to the penalty function $\mathsf{pen}(d_j)$ in experiment 4.

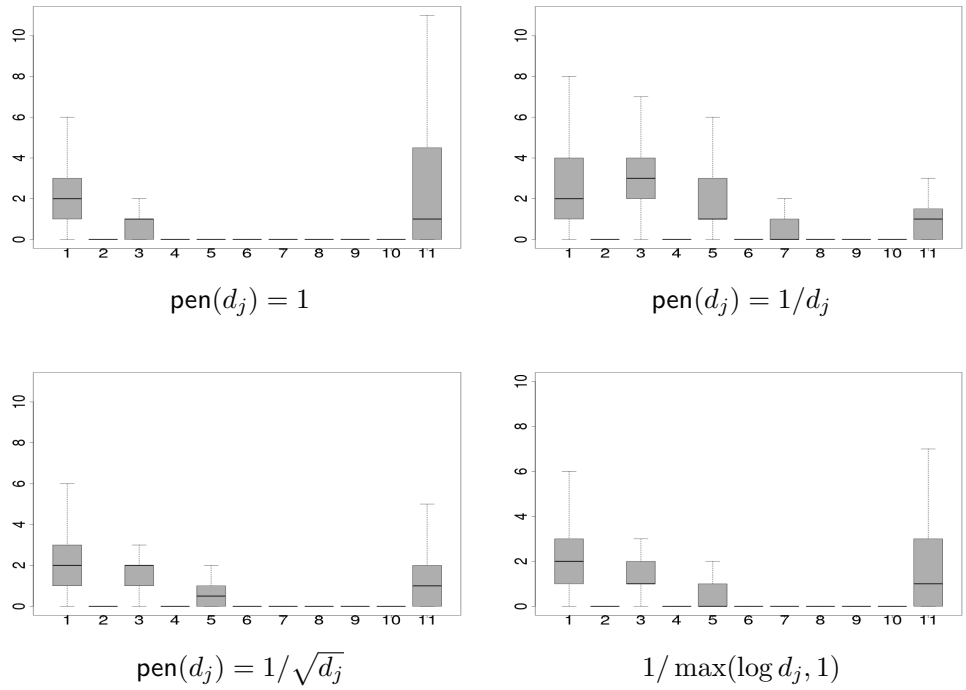


Figure 7: Group selection for TPLDA according to the penalty function $\mathsf{pen}(d_j)$ in experiment 5.

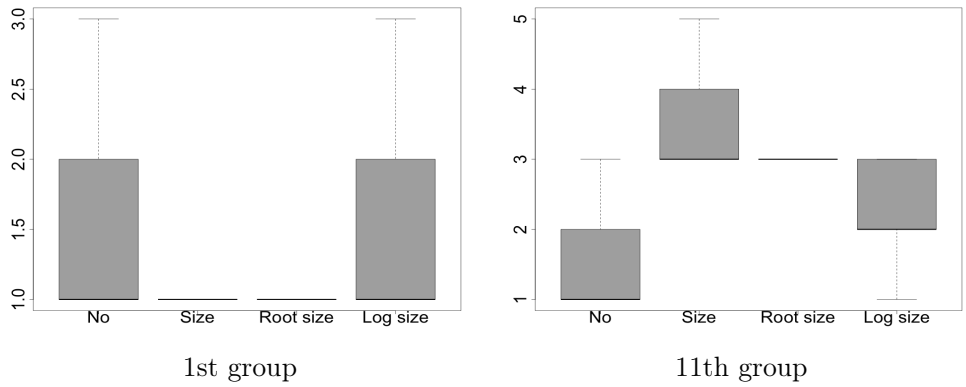
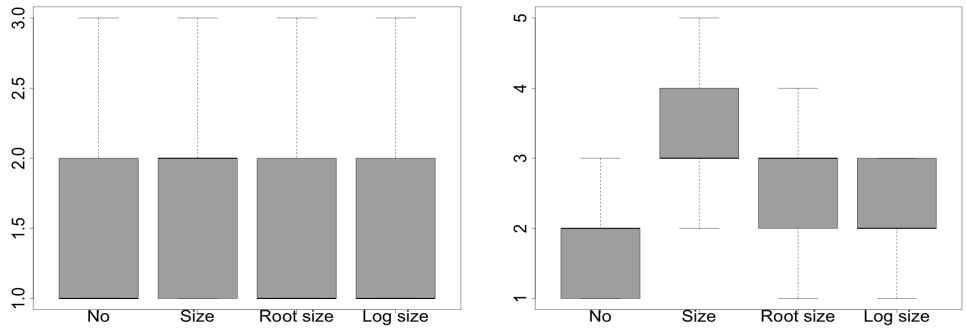
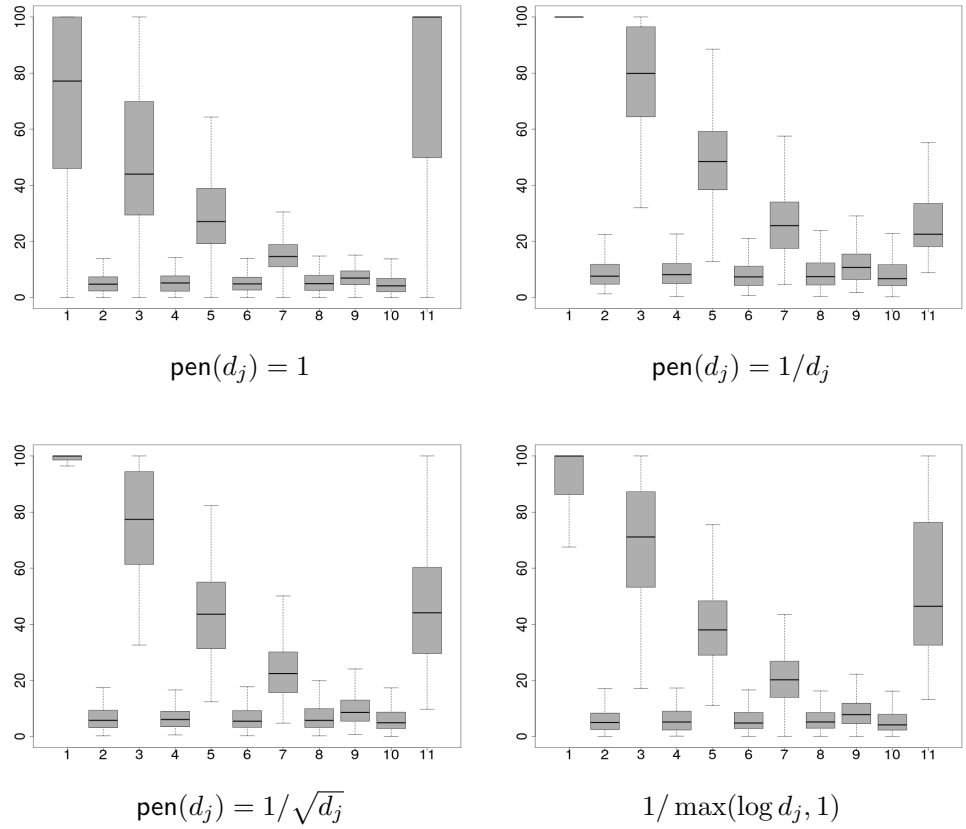


Figure 8: Sensitivity to the choice of the penalty function pen in TPLDA: depth of the first node split by using the 1st group or 11th group in experiment 4.



1st group 11th group

Figure 9: Sensitivity to the choice of the penalty function pen in TPLDA: depth of the first node split by using the 1st group or 11th group in experiment 5.



$\mathsf{pen}(d_j) = 1$ $\mathsf{pen}(d_j) = 1/d_j$

$\mathsf{pen}(d_j) = 1/\sqrt{d_j}$ $1/\max(\log d_j, 1)$

Figure 10: Distribution of the importance score for each group according to the penalty function in experiment 4.

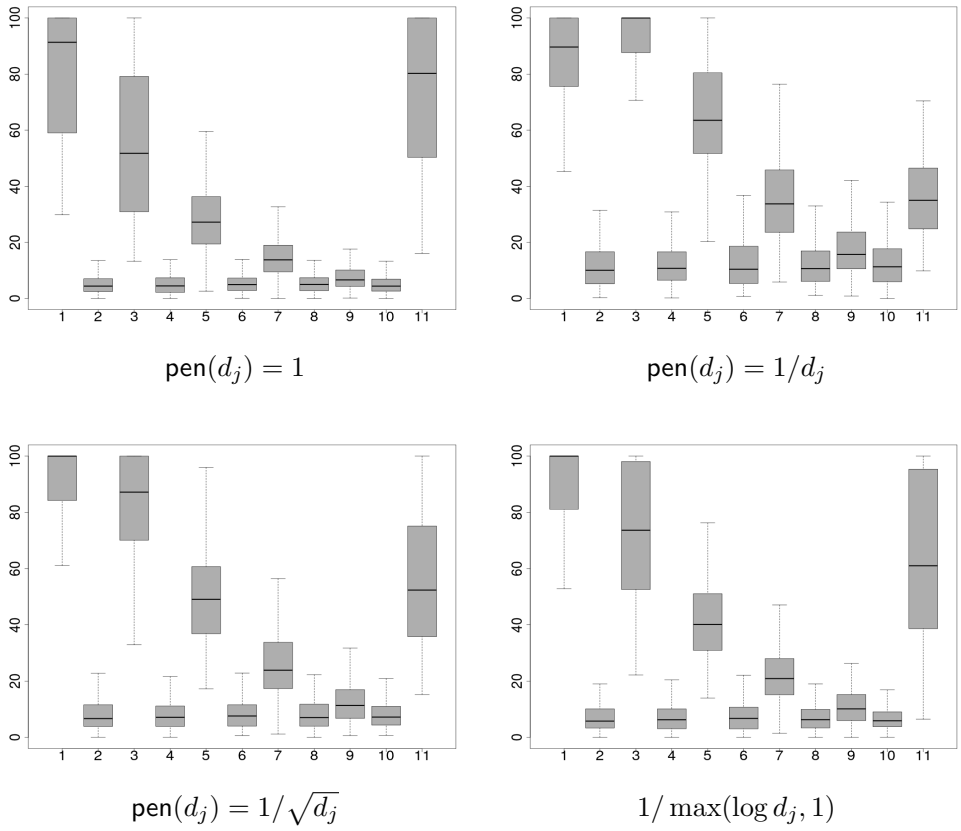


Figure 11: Distribution of the importance score for each group according to the penalty function in experiment 5.

Penalty	1	$1/d_j$	$1/\sqrt{d_j}$	$1/\max(\log d_j, 1)$
Experiment 4				
Selection rate of the 5 relevant groups	0	2	0.5	0
Selection rate of at least 3 relevant groups	99.5	100	100	100
Selection rate of the 3 most relevant groups	95.5	100	99.5	100
Median ranking of the 1st group	2 (1,2)	1 (1,1)	1 (1,2)	1 (1,2)
Median ranking of the 11th group	1 (1,3)	5 (4,5)	3 (3,4)	3 (2,4)
Experiment 5				
Selection rate of the 5 relevant groups	0	3.5	0	0
Selection rate of at least 3 relevant groups	99.5	100	100	100
Selection rate of the 3 most relevant groups	98	99.5	100	98.5
Median ranking of the 1st group	2 (1,2)	2 (1,2)	1 (1,2)	1 (1,2)
Median ranking of the 11th group	2 (1,3)	5 (4,5)	3 (3,4)	2 (3,4)

Table 5: Sensitivity to the choice of the penalty function pen: assessment of the score of group importance.

4 Application to tumor classification using gene expression data

Nowadays, the ability to classify tumors subtypes using gene expression data is still challenging. Indeed, the nature of both high dimensionality and small size associated with gene expression data, that is a large number of variables relative to a much smaller number of observations, implies the use of features selection, clustering and/or regularized methods. Moreover, the resulted model must be easily understandable to enable identification of "marker" genes and characterization of the tumor subtypes.

In this paper, TPLDA, CART, GL are applied to datasets from three published cancer gene expression studies. For comparison purpose, the shrunken centroid regularized discriminant analysis (SCRDA) method (Guo et al., 2006), which is one of the standard methods used to classify tumors with gene expression data, is also applied to the three datasets. Furthermore, in each dataset, we assume that genes have a group structure (Kaminski & Friedman, 2002; Lee & Batzoglou, 2003; Yin et al., 2006). Clusters can be defined based on existing sources of biological knowledge for gene functions or pathways (Ashburner et al., 2000; Kanehisa & Goto, 2000) or by using clustering methods (Lee & Batzoglou, 2003; Yin et al., 2006). Here, groups of genes are determined by applying independent component analysis (ICA), an unsupervised clustering approach used for instance by Lee & Batzoglou (2003) and Engreitz et al. (2010) to identify groups of genes

By applying TPLDA, CART, GL and SCRDA on each dataset, the objective is to elaborate a classification rule with a good prediction accuracy and which enables to highlight some relevant (groups of) genes. The three public microarray gene expression datasets used are briefly described below and in Table 6.

(1) The leukemia dataset (Golub et al., 1999) consists of an original training set, that gives the expression level of 7129 genes from 38 samples, and an original test set giving the expression level of 2185 genes from 34 patients. Based on pathological and histological criteria, in the training set, 27 tumor samples are classified as acute lymphobastic leukemias (called ALL) and the remaining 11 samples are classified as acute myeloid leukemias (called AML). In the test sample, there 20 ALL tumors and 14 AML tumors. The two datasets have been merged into a larger dataset that consists of the expression levels of 2135 genes from 72 samples (47 ALL and 25 AML).The data can be freely downloaded from `http://www.broadinstitute.org/cgi-bin/cancer/publications/pub_paper.cgi?paper_id=43`.

(2) The lymphoma dataset (Shipp et al., 2002) consists of 7129 gene expression levels from 77 lymphomas. The 77 samples are divided into 58 diffuse large B-cell lymphomas (DLBCL) and 19 follicular lymphomas (FL). The data can be found at `https://github.com/ramhiser/datamicroarray/blob/master/data/shipp.RData`.

(3) The colon dataset is from the microarray experiment of colon tissues samples of Alon et al. (1999). It contains the expression level of 2000 genes for 40 tumors and 22 normal colon tissues. The data can be freely downloaded from `http://microarray.princeton.edu/oncology/affydata/index.html`.

Dataset	**Reference**	**Number of samples**	**Classes**	**Number of genes**
Leukemia	Golub et al. (1999)	72	ALL(47), AML(25)	2185
Lymphoma	Shipp et al. (2002)	77	Cured(32), Disease(26)	7129
Colon	Alon et al. (1999)	62	Tumor(40), Normal(22)	2000

Table 6: The three datasets used in our application.

4.1 Data preprocessing and elaboration of the groups of genes

4.1.1 Data preprocessing

Following Dudoit et al. (2002), Shipp et al. (2002) and Sewak et al. (2009), a data preprocessing is applied to each dataset. First, values are thresholded between 16000 and 20. The upper limit of 16000 corresponds to the saturation level of the fluorescence technique used to extract the values of the gene intensity while the floor of 20 is used because measures of gene intensities below 20 are not considered as meaningful with the fluorescence technique used to extract the data (for more details see Shipp et al., 2002). Next, in each dataset, only the first quartile of genes with the greatest variation across the sample is considered, the other genes are excluded.

4.1.2 Gene clustering

After that, independent component analysis (ICA) is applied on each dataset (Lee & Batzoglou, 2003) to cluster the remaining genes. In this approach, each independent component is considered as a putative biological pathway which can be characterized by the genes that contribute the most to the related independent component. Genes are then clustered into non-mutually exclusive groups based on their load on each ICA component. Finally, in each dataset, the distribution of the intensity of each selected gene is standardized to have zero mean and unit variance. The resulted datasets are next used to assess the methods. The used process is described in the following subsection.

4.2 Assessment of the methods

4.2.1 Generation of the bootstrap samples

The assessment of the methods is based on bootstrap simulations applied on the datasets obtained after gene clustering and data preprocessing. First, each dataset is 500 times randomly divided into a training sample, a validation sample and a test

sample with the respective proportions $(0.8, 0.1, 0.1)$. In the training set, classes are balanced by using the Synthetic Minority Over-sampling Technique proposed by Chawla et al. (2002).

4.2.2 Application of the methods

For TPLDA and CART, a maximal tree is built on the training sample. To prune the maximal tree, TPLDA uses the pruning procedure described in Section 2.3 and the validation sample. CART uses cross-validation on the training sample and the minimal cost-complexity pruning method to select the final tree. For GL, the model is elaborated on the training set and the tuning parameter is selected by using 5-fold cross-validation. For SCRDA, the model is elaborated on the training set and the tuning parameter is selected by using the 10-fold cross-validation and the *Min-Min* rule proposed in the original paper (Guo et al., 2006). We use the function `rda` in the R package rda to compute SCRDA. TPLDA and GL are applied on the groups of genes created during the clustering step while CART and SCRDA are applied on the individual genes selected during the data-preprocessing phase since these two methods do not allow to take into account the groups of genes.

Following previous studies with microarray data (Guo et al., 2006; Huang et al., 2009; Tai & Pan, 2007; Sewak et al., 2009; Dudoit et al., 2002), the predictive performances of all the assessed methods are measured by using the average error rate estimated on the test sample.

Finally, the measure of group importance provided by TPLDA is investigated using the leukemia data. TPLDA is applied on the original training set to elaborate a maximal tree that is next pruned by using the original test sample and the pruning method described in Section 2.3. For the purpose of comparison, GL is also applied on the original training set to elaborate a model and the shrinkage parameter is selected by using cross-validation on the original test sample. The colon data and the lymphoma data are not used to study the interest of the measure of group importance since no test sample is available.

4.3 Results

Table 7 describes the datasets after performing data preprocessing and gene clustering. For each dataset, 15 non-mutually exclusive groups of genes are created. In Table 7, the number of selected genes refers to the number of distinct genes which belong to the 15 groups. The elaboration of the groups of genes is explained in Appendix D.

Dataset	**Size of the groups**	**Number of selected genes**
Leukemia	28	99
Lymphoma	18	100
Colon	26	106

Table 7: The datasets after applying data preprocessing and clustering genes into 15 groups.

Table 8 shows the results. For each method and each dataset, the average error rate over the 500 samples is displayed. For CART and TPLDA, the average tree depth is given. The table also shows the number of selected group (respectively genes) for GL (respectively SCRDA). Since the choice of both the number of groups and the group size may influence the results, several analyses have been performed by varying the values of these two parameters. Table D in Appendix D gives the results when genes are clustered into 50 groups. Results show no significant difference, that is consistent with previous sensitivity studies (Lee & Batzoglou, 2003).

	Classifier	Average error rate (in %)	Average tree depth	Number of groups/genes
Leukemia	TPLDA	**9 (11)**	1	
	CART	14 (14)	2	
	GL	**7 (10)**		4
	SCRDA	**9 (11)**		55
Lymphoma	TPLDA	**16** (15)	2	
	CART	20 (17)	3	
	GL	**13 (14)**		5
	SCRDA	21 (18)		25
Colon	TPLDA	**20 (17)**	2	
	CART	26 (18)	2	
	GL	**16 (17)**		4
	SCRDA	31 (9)		6

Table 8: Average error rate for 500 samples when genes are clustered into 15 groups. The standard error is given in brackets. The number of groups/genes is the average number of groups (respectively genes) included in the model for GL (respectively SCRDA).

The methods using the group structure (i.e. TPLDA and GL) generally outperform the others which emphasizes the interest of taking into account the group structure when data are grouped. Moreover, TPLDA performs consistently well for all

datasets. GL tends to select more groups that TPLDA[1]. This may be explained by the fact that groups can overlaps, i.e. the situation in which some groups can have genes in common. In this context, it is known that GL tends to select too much groups. Moreover, if some relevant groups are very correlated, GL can select just a subset of these groups and so GL might not be reliable to perform group selection (see for example Jacob et al., 2009).

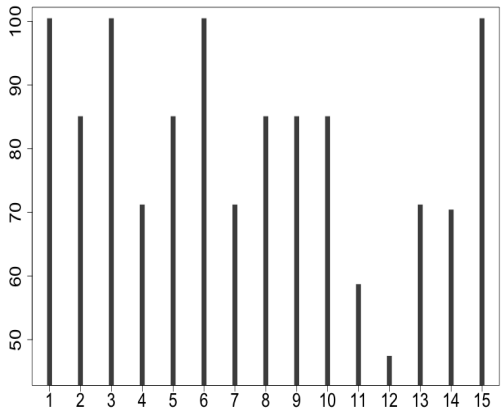


Figure 12: Importance of the 15 groups of genes in the leukemia study.

Compared to GL, TPLDA provides automatically a measure of importance for each group of genes, even for groups which are not included in the tree. This score of importance gives information about the relative importance of each group and also allows to perform selection of groups of genes. Figure 12 displays the measure of importance of each group, on the leukemia data. The measure of importance enables to highlight four groups: the first, the third, the sixth and the fifteenth groups. These groups build the same classification rule that explains why there have the same measure of importance. By comparison, GL includes only the fourth and the fifth groups in the model. Note that these two groups are not considered as the most relevant groups with respect to the measure of importance computed with TPLDA, even though they share some common genes with the groups considered as the most relevant for TPLDA (groups can be available at `https://github.com/apoterie/TPLDA`). Contrary to TPLDA, GL provides no information about the prediction strength of the groups that are not included in the model. So, GL does not able to identify relevant groups of genes that would not be included in the model because there are highly correlated with the groups included in the model. Moreover, as previously mentioned, in case of overlap groups, GL may not be reliable to identify the truth relevant groups. Since we have no expert knowledge for gene functions and pathways, we are not attempted to provide biological interpretation of the groups of genes selected by GL and TPLDA. However, these results seem quite consistent

[1]The number of groups included in the classification rule built by using TPLDA is lower than or equal to the number of splits in the final TPLDA tree. Note that it equals number of splits in the final TPLDA tree when each selected group is used only once.

with those presented in the original paper (Golub et al., 1999) because these groups include some genes that have been reported as informative genes in the original paper.

Thus, TPLDA can also be used to perform group variable selection. Compared to GL, TPLDA achieves a better trade-off between prediction accuracy and group variable selection.

Remark 4.1. *In these three examples, SCRDA seems not be the best method although it was especially developed to classify tumors with gene expression data. The observed predictive performances of SCRDA here may be explained by the exclusion of many genes during the data preprocessing and the clustering process. Indeed, previous studies showed that SCRDA gives good performances when the method is applied on datasets involving a large number of variables relative to the much small number of observations (see Huang et al., 2009; Guo et al., 2006, for instance).*

5 Conclusion

In this work we have presented a new way to classify data with grouped inputs. Our approach consists in using recursive penalized linear discriminant analysis to build a classification tree based on the groups of variables. To our knowledge, it is the first classification trees algorithms dealing with grouped inputs.

The TPLDA method can be considered as a multivariate classification tree algorithm which uses linear combinations of inputs to build the partition, as already do several multivariate classification tree algorithms. However, contrary to most of the multivariate classification tree algorithms (Breiman et al. 1984; Murthy et al. 1993; Loh & Shih 1997; Li et al. 2003; Wickramarachchi et al. 2016, etc.), TPLDA is not computationally expensive. Moreover, classification trees obtained by using TPLDA are more easily understandable since the classification rule is based on the group structure which makes sense.

Through applications on simulated datasets and real datasets, we have shown that TPLDA is well adapted to classify data with groups of inputs. Furthermore, the group importance measure computed within the TPLDA method allows to quantify the relevance of each group of variables, even if some groups are not included in the resulted final classification tree. Consequentially the TPLDA method also enables to answer the second most important issue in supervised classification: the identification of relevant groups of variables and/or the group variable selection. Thus, this algorithm shows promising results in terms of predictive performances, interpretation and variable selection.

Bibliography

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*, 6745–6750.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T. et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, *25*, 25.

Bouveyron, C., Girard, S., & Schmid, C. (2007). High-dimensional discriminant analysis. *Communications in Statistics Theory and Methods*, *36*, 2607–2623.

Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC Press.

Brodley, C. E., & Utgoff, P. E. (1995). Multivariate decision trees. *Machine learning*, *19*, 45–77.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321–357.

Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, *97*, 77–87.

Engreitz, J. M., Daigle Jr, B. J., Marshall, J. J., & Altman, R. B. (2010). Independent component analysis: mining microarray data for fundamental human gene expression modules. *Journal of biomedical informatics*, *43*, 932–944.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* volume 1. Springer Series in Statistics New York.

Friedman, J. H. (1989). Regularized discriminant analysis. *Journal of the American Atatistical Association*, *84*, 165–175.

Genuer, R., & Poggi, J.-M. (2017). Arbres CART et Forêts aléatoires,Importance et sélection de variables. Preprint.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A. et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, *286*, 531–537.

Gregorutti, B., Michel, B., & Saint-Pierre, P. (2015). Grouped variable importance with random forests and application to multiple functional data analysis. *Computational Statistics & Data Analysis*, *90*, 15–35.

Grimonprez, Q., Blanck, S., Celisse, A., & Marot, G. (2018). Mlgl: An r package implementing correlated variable selection by hierarchical clustering and group-lasso. Preprint.

Guo, Y., Hastie, T., & Tibshirani, R. (2006). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, *8*, 86–100.

Huang, D., Quan, Y., He, M., & Zhou, B. (2009). Comparison of linear discriminant analysis methods for the classification of cancer based on gene expression data. *Journal of Experimental & Clinical Cancer Research*, *28*, 149.

Huang, J., Breheny, P., & Ma, S. (2012). A selective review of group selection in high-dimensional models. *Statistical science: a review journal of the Institute of Mathematical Statistics*, *27*.

Jacob, L., Obozinski, G., & Vert, J.-P. (2009). Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning* (pp. 433–440). ACM.

Kaminski, N., & Friedman, N. (2002). Practical approaches to analyzing results of microarray experiments. *American journal of respiratory cell and molecular biology*, *27*, 125–132.

Kanehisa, M., & Goto, S. (2000). Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, *28*, 27–30.

Lange, K., Hunter, D. R., & Yang, I. (2000). Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical statistics*, *9*, 1–20.

Lee, S.-I., & Batzoglou, S. (2003). Application of independent component analysis to microarrays. *Genome Biology*, *4*, R76.

Li, X.-B., Sweigart, J. R., Teng, J. T., Donohue, J. M., Thombs, L. A., & Wang, S. M. (2003). Multivariate decision trees using linear discriminants and tabu search. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *33*, 194–205.

Lim, T.-S., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, *40*, 203–228.

Loh, W. (2014). Fifty years of classification and regression trees. *International Statistical Review*, *82*, 329–348.

Loh, W.-Y., & Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, *7*, 815–840.

Meier, L., Geer, S. V. D., & Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *70*, 53–71.

Mola, F., & Siciliano, R. (2002). Discriminant analysis and factorial multiple splits in recursive partitioning for data mining. In *International Workshop on Multiple Classifier Systems* (pp. 118–126). Springer, Berlin, Heidelberg.

Murthy, S. K., Kasif, S., Salzberg, S., & Beigel, R. (1993). OC1: A randomized algorithm for building oblique decision trees. In *Proceedings of AAAI* (pp. 322–327). AAAI volume 93.

Picheny, V., Servien, R., & Villa-Vialaneix, N. (2016). Interpretable sparse sir for functional data. Preprint.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

Sewak, M. S., Reddy, N. P., & Duan, Z.-H. (2009). Gene expression based leukemia sub-classification using committee neural networks. *Bioinformatics and Biology Insights*, *3*, 89.

Shao, J., Wang, Y., Deng, X., Wang, S. et al. (2011). Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of Statistics*, *39*, 1241–1265.

Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. S. et al. (2002). Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, *8*, 68.

Strobl, C., Boulesteix, A.-L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinformatics*, *8*, 25.

Tai, F., & Pan, W. (2007). Incorporating prior knowledge of gene functional groups into regularized discriminant analysis of microarray data. *Bioinformatics*, *23*, 3170–3177.

Tamayo, P., Scanfeld, D., Ebert, B. L., Gillette, M. A., Roberts, C. W., & Mesirov, J. P. (2007). Metagene projection for cross-platform, cross-species characterization of global transcriptional states. *Proceedings of the National Academy of Sciences*, *104*, 5959–5964.

Wei-Yin Loh, N. V. (1988). Tree-structured classification via generalized discriminant analysis. *Journal of the American Statistical Association*, *83*, 715–725.

Wickramarachchi, D., Robertson, B., Reale, M., Price, C., & Brown, J. (2016). HHCART: an oblique decision tree. *Computational Statistics & Data Analysis*, *96*, 12–23.

Witten, D. M., & Tibshirani, R. (2011). Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*, 753–772.

Xu, P., Brock, G. N., & Parrish, R. S. (2009). Modified linear discriminant analysis approaches for classification of high-dimensional microarray data. *Computational Statistics & Data Analysis*, *53*, 1674–1687.

Yin, L., Huang, C.-H., & Ni, J. (2006). Clustering of gene expression data: performance and similarity analysis. *BMC Bioinformatics*, *7*, 19–30.

A Time complexity of TPLDA

In the following section, the maximal time complexity at a node t of the TPLDA method is detailed. We assume that there are:

- n_t observations in the node t,
- J groups of variables denoted X^j, for $j = 1, \ldots, J$ and
- the group $\mathbf{X}^j$, with $j = 1, \ldots, J$, includes d_j variables such as $\mathbf{X}^j = (X_{j_1}, X_{j_2}, \ldots, X_{j_{d_j}})$.

To split a node t including n_t observations, TPLDA uses the following two steps:

- Step 1 within group PLDA.
  For any group $\mathbf{X}^j$ of variables, with $j = 1, \ldots, J$, a PLDA is applied on the node t and the shrinkage parameter λ_j is selected by cross-validation.
- Step 2 choosing the splitting group.
  For any group $\mathbf{X}^j$ of variables, with $j = 1, \ldots, J$, TPLDA computes the penalized decrease in node impurity resulting from splitting on group j and selects the group that maximizes it.

The time complexity of these steps are detailed below. Consider the group j, with $j = 1, \ldots, J$.

Complexity when performing PLDA on group j:
PLDA computation steps are described in the original paper (Witten & Tibshirani, 2011). We detailed here its maximal time complexity:

- Complexity for constructing the estimated between covariance matrix $\widehat{B}_t^j$ is $\mathcal{O}\left(n_t d_j^2\right)$.
- Complexity for constructing the diagonal positive estimate of the within covariance matrix $\widehat{\Sigma}_t^j$ is $\mathcal{O}\left(n_t d_j\right)$.
- Complexity of the eigen analysis of $(\widehat{\Sigma}_t^j)^{-1}\widehat{B}_t^j$ is $\mathcal{O}\left(d_j^2\right)$.

- Complexity of the eigen analysis of $(\widehat{B}_t^j)^{-1}\widehat{B}_t^j$ and the research for the dominant eigenvector is $\mathcal{O}\left(d_j^2\right)$.

- Complexity for estimating the penalized discriminant vector $\hat{\beta}^j$ by performing M iterations of the minimization-maximization algorithm (Lange et al., 2000) is $\mathcal{O}\left(Md_j^2\right)$.

$\Rightarrow$ So the maximal time complexity of performing PLDA on the group j in the node t is $\mathcal{O}\left(n_t d_j^2\right) + \mathcal{O}\left(n_t d_j\right) + \mathcal{O}\left(d_j^2\right) + \mathcal{O}\left(Md_j^2\right) = \mathcal{O}\left(n_t d_j^2\right)$ by supposing that $M < n_t$.

Complexity when selecting of the shrinkage parameter λ_j:
The value of shrinkage parameter λ_j is determined by using a K-fold cross-validation and a grid $\{v_1, \ldots, v_L\}$ containing L values for λ_j. The maximal time complexity of this step is detailed below:

- Complexity for dividing the n_t observations in the node t into K disjoint samples $\{S_1, \ldots, S_K\}$ is $\mathcal{O}\left(n_t\right)$.

- For each fold k, $k = 1, \ldots, K$ and each value v_ℓ, $\ell = 1, \ldots, L$:
  - Complexity for performing a PLDA on $t \setminus S_k$ (i.e. all the disjoint sets $\{S_1, \ldots, S_K\}$ excepted S_k) with $\lambda_j = v_\ell$ is $\mathcal{O}\left(\frac{K-1}{K} n_t d_j^2\right)$.
  - Complexity for predicting the class of each observation in S_k using the resulted PLDA model computed on $t \setminus S_k$ is $\mathcal{O}\left(\frac{n_t}{K} d_j\right)$.
  - Complexity for computing the penalized decrease $\Delta_j(t, v_\ell)$ in node impurity is $\mathcal{O}\left(n_t\right)$.

- Complexity for choosing the value in the grid $\{v_1, \ldots, v_L\}$ which maximizes the penalized decrease in node impurity is $\mathcal{O}\left(1\right)$.

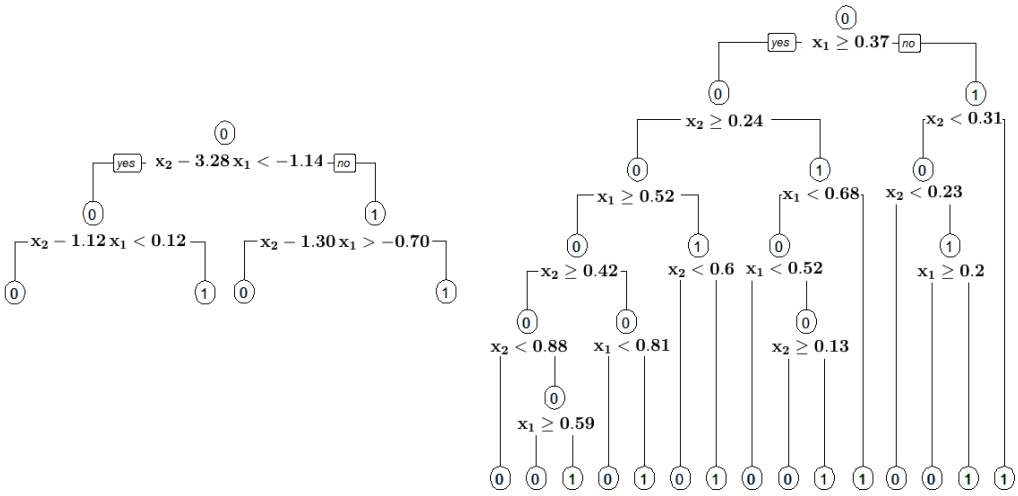
$\Rightarrow$ So the complexity for selecting the value of λ_j is $\mathcal{O}\left(n_t\right) + \mathcal{O}\left(L(K-1)n_t d_j^2\right) + \mathcal{O}\left(Ln_t d_j\right) + \mathcal{O}\left(Ln_t\right) + \mathcal{O}\left(1\right) = \mathcal{O}\left(L(K-1)n_t d_j^2\right)$.

Complexity when choosing the splitting group:
TPLDA selects among the J estimated splits the one which maximizes the impurity decrease. The complexity of this step is $\mathcal{O}\left(1\right)$.

Consequently, the maximal time complexity of TPLDA at a node t is in the worst case $\mathcal{O}\left(Jn_t d_{\max}^2\right) + \mathcal{O}\left(JL(K-1)n_t d_{\max}^2\right) + \mathcal{O}\left(1\right) = \mathcal{O}\left(JLKn_t d_{\max}^2\right)$ with $d_{\max} = \max_j(d_j)$.

B Additional figures about the illustration of the TPLDA method on a simple example

Figure 13 displays the two trees built by CART and TPLDA in the simple example used to illustrate TPLDA in Section 2.4. As mentioned previously, in this example, the TPLDA tree is much easier than the CART tree. Moreover, the simple TPLDA tree is as accurate as the complex CART tree (TPLDA AUC = 0.90, CART AUC = 0.89).



The TPLDA tree The CART tree

Figure 13: The two trees associated to the TPLDA and CART partitions displayed in Figure 2. Circles define the nodes and the figure in each node indicates the node label. The splitting rule is denoted below each node.

C Additional information about the numerical experiments

This section provides additional results and figures about the simulation studies.

C.1 Justification for using PLDA instead of FDA in the splitting process

First of all, here we discuss the choice of using PLDA instead of FDA in the splitting process. In TPLDA, PLDA is replaced by FDA. This modified TPLDA is named TLDA and is applied on each sample of the first three experiments.

Table 9 displays the simulation results for TLDA in comparison with TPLDA. First, when data are not grouped, TPLDA and TLDA give almost the same results and

can be then used interchangeably. Indeed, when the group size equals 1 and if the regularized parameter in the PLDA problem (3) is set to zero, the FDA problem and the PLDA problem are identical.

In the second and the third experiment, TLDA underperforms TPLDA. This can be explained by the fact that FDA performs badly in small nodes i.e. in the nodes where the number of observations is small relative to the size of some groups of variables (Shao et al., 2011; Friedman, 1989; Xu et al., 2009; Bouveyron et al., 2007). Yet, tree elaboration is based on a recursive splitting procedure which creates nodes that becomes smaller and smaller whereas the sizes of input groups remain unchanged. Then FDA may not be appropriate for estimating recursively the hyperplane splits. This is well illustrated by the performances of TLDA in the third experiment where the groups of input variables are large compared to the number of observations in the training sample. Indeed, in the first split, the FDA used to split the entire data space overfits the training set. This can be seen in Figure 14: the training misclassification error decreases much faster for TLDA and becomes smaller than the Bayes error from the first split while the test misclassification error for TLDA remains stable. Consequently, after applying the pruning procedure which removes the less informative nodes, the final TLDA tree is trivial in at least 25 % of the simulations (Table 9). Conversely, TPLDA does not seem to be affected by the high-dimension. Then, PLDA overcomes the weakness of FDA in high-dimensional situations.

	TPLDA	TLDA
Exp. 1		
AUC	0.66 (0.65,0.68)	0.66 (0.65,0.67)
Tree depth	4 (3,5)	4 (3,5)
Exp. 2		
AUC	0.76 (0.74,0.77)	0.67 (0.64,0.69)
Tree depth	3 (3,4)	3 (2,3)
Exp. 3		
AUC	0.83 (0.7,0.85)	0.5 (0.5,0.52)
Tree depth	2 (2,3)	1 (0,2)

Table 9: Comparison between TLDA and TPLDA: simulation results. For each criterion, the median value is given following by the values in brackets of the first and the third quartiles.

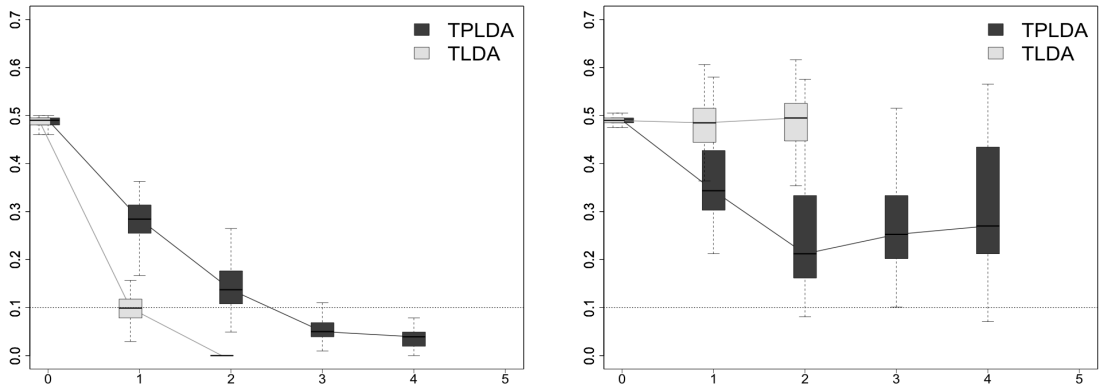


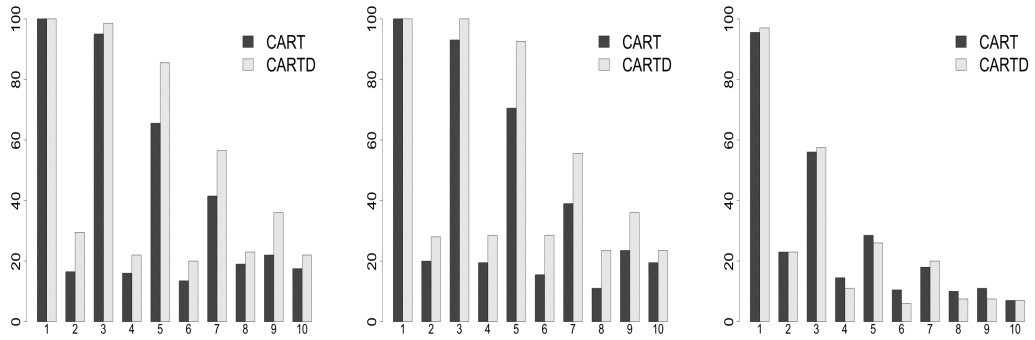
Figure 14: Misclassification error estimate according to the tree depth on the training set (left) and on the validation set (right) in experiment 3. The dotted lines denote the values of the Bayes error (Bayes error=10%).

C.2 Sensitivity to the pruning strategy for CART

Here, the performances of CART when using the cost-complexity pruning strategy are compared to those obtained by using the proposed pruning strategy based on the tree depth, in the first three experiments. The approach using the proposed pruning strategy is named CARTD (while the approach using the cost-complexity pruning is named CART). The results are given in Table 10. Figure 15 displays the group selection frequencies of CART and CARTD. Overall, the two pruning methods lead to similar CART trees and so similar classification rules. Indeed, the predictive performances and the tree depth are very close. CARTD trees may be slightly smaller. Moreover, the group selection frequencies do not really differ: they are lightly higher when using the proposed pruning strategy based on the depth. Thus, CART performances do not seem to be sensitive to the choice of one of the two pruning methods.

	CART	CARTD
Experiment 1		
AUC	0.67 (0.65,0.68)	0.67 (0.66,0.69)
Tree depth	5 (3,7)	5 (4,6)
Experiment 2		
AUC	0.68 (0.66,0.70)	0.68 (0.67,0.70)
Tree depth	6 (4,8)	5 (4,7)
Experiment 3		
AUC	0.64 (0.62,0.66)	0.65 (0.62,0.67)
Tree depth	4 (2,5)	3 (2,4)

Table 10: Performances of CART and CARTD. For each criterion, the median value is given following by the values in brackets of the first and the third quartiles.



Experiment 1 Experiment 2 Experiment 3

Figure 15: Group selection frequency (in %) for CART according to the pruning strategy in the first three experiments.

C.3 Additional results

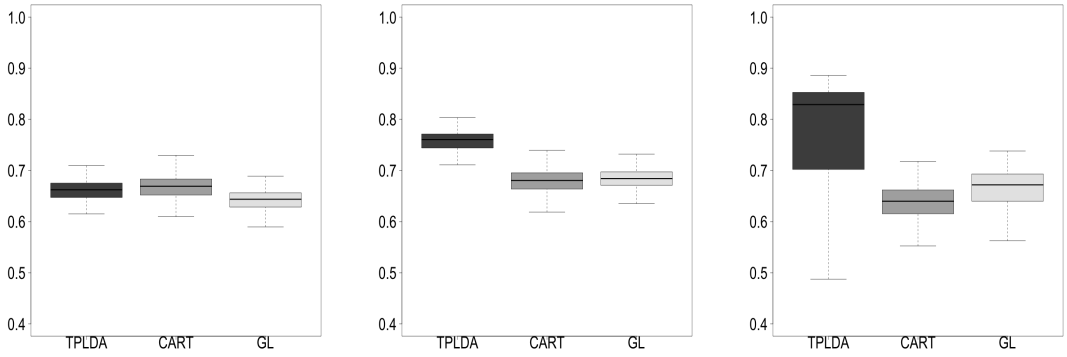
Table 11 displays the simulation results for TPLDA, TLDA, CART and GL for the fourth and fifth scenarios. As previously, TLDA underperforms TPLDA. GL and CART overperform slightly TPLDA when no penalty function is used.

	TPLDA	TLDA	CART	GL
Exp. 4				
AUC	0.65 (0.60,0.70)	0.59 (0.54,0.65)	0.68 (0.66,0.69)	0.66 (0.65,0.68)
Tree depth	2 (2,4)	2 (2,3)	5 (4,7)	.
Model size	.	.	.	4 (3,5)
Exp. 5				
AUC	0.66 (0.60,0.71)	0.58 (0.53,0.63)	0.68 (0.66,0.69)	0.66 (0.64,0.68)
Tree depth	2 (2,3)	2 (2,3)	5 (4,7)	.
Model size	.	.	.	5 (4,6)

Table 11: Additional simulation results. For each criterion, the median value is given following by the values in brackets of the first and the third quartiles. The model size gives the number of groups of variables included in the GL model.

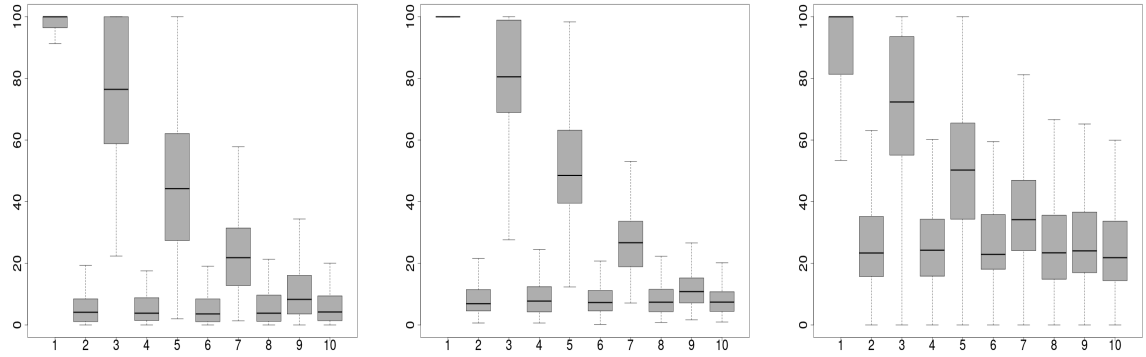
C.4 Additional figures

Additional figures about the simulation studies are displayed in this subsection. Figure 16 displays the predictive performances of TPLDA, CART and GL in the first three experiments. Figure 17 shows the distribution of the importance score for each group in the first three experiments. Figure 10 and Figure 11 display the distribution of the importance score for each group in the fourth and the fifth experiments according to the penalty function.



Experiment 1 Experiment 2 Experiment 3

Figure 16: Predictive performances of the assessed methods: boxplots of the AUC for the first three experiments.



Experiment 1 Experiment 2 Experiment 3

Figure 17: Distribution of the importance score for each group in the first three experiments.

D Additional information about the application to gene expression data

Following Lee & Batzoglou (2003), we assume that each independent component refers to a putative biological process and that a group of genes is then created for each independent component. For a given independent component, the most important genes are the genes with the largest loads in absolute terms. Then, the group of genes associated to the given independent component includes the $C\%$ of genes with the largest loads in absolute terms. The number of groups J (or equivalently the number of independent components) and the threshold parameter C are tuning parameters.

For each dataset, several values for the clustering parameters (J,C) are chosen in order to assess the sensitivity of the predictive performances of the methods TPLDA, CART, GL and SCRDA to the values of these parameters. For each dataset we show the results for two couples (J,C) (Table D). In each dataset, genes are clustered into 15 or 50 non-mutually exclusive groups (Tables 7 and 13). Table D shows the results when genes are clustered into 50 groups of equal size. These results are not significantly different from those obtained when using 15 groups (Table 8).

	Total number of genes	Number of groups (J)	Threshold parameter (C)
Leukemia	2186	15	5
		50	2.5
Lymphoma	7129	15	1
		50	0.5
Colon	2000	15	5
		50	2.5

Table 12: Choice of the clustering parameters for each dataset.

Dataset	**Size of the groups**	**Number of selected genes**
Leukemia	14	101
Lymphoma	10	108
Colon	14	112

Table 13: The datasets after applying data preprocessing and clustering genes into 50 groups.

	Classifier	Average error rate (%)	Average tree depth	Average number of groups/genes
Leukemia	TPLDA	9 (12)	1	
	CART	17 (15)	2	
	GL	**7 (10)**		5
	SCRDA	**8 (10)**		60
Lymphoma	TPLDA	**16 (15)**	2	
	CART	22 (17)	2	
	GL	**12 (13)**		7
	SCRDA	17 (16)		28
Colon	TPLDA	**20 (15)**	1	
	CART	25 (17)	2	
	GL	**16 (17)**		5
	SCRDA	29 (11)		16

Table 14: Average error rate for 500 samples when genes are clustered into 50 groups. The standard error is given in brackets. The number of groups/genes is the average number of groups (respectively genes) included in the model for GL (respectively SCRDA).