Noname manuscript No.
(will be inserted by the editor)

Isotonic boosting classification rules

David Conde · Miguel A. Fernández · Cristina Rueda · Bonifacio Salvador

the date of receipt and acceptance should be inserted later

Corresponding Author. Miguel A. Fernández E-mail: miguelaf@eio.uva.es. Phone: +34983423945. Orcid: 0000-0002-1272-8950

Departamento de Estadística e Investigación Operativa. Facultad de Ciencias. Paseo de Belén 7. Universidad de Valladolid. 47011 Valladolid, Spain.

Isotonic boosting classification rules

Abstract In many real classification problems a monotone relation between some predictors and the classes may be assumed when higher (or lower) values of those predictors are related to higher levels of the response. In this paper, we propose new boosting algorithms, based on LogitBoost, that incorporate this isotonicity information, yielding more accurate and easily interpretable rules. These algorithms are based on theoretical developments that consider isotonic regression. We show the good performance of these procedures not only on simulations, but also on real data sets coming from two very different contexts, namely cancer diagnostic and failure of induction motors.

Keywords Classification · Boosting · LogitBoost · Additive models · Isotonic regression · TMP.

Mathematics Subject Classification (2010) 62H30

1 Introduction

The classical problem of classifying observations in one of K groups using a rule defined from a sample of observations for which the true group is known (training sample) is known as supervised classification. This problem has been receiving a lot of attention in the last decades due to its applicability in a very wide range of problems in different scientific fields such as economy, engineering, medicine or molecular biology. In fact, it could be said that the problem can appear in any scientific field. As a consequence, many methods and techniques to build classification rules have been developed, from the classic linear or quadratic rules to the more recent K Nearest Neighbors (KNN), Support Vector Machines (SVM) or decision trees. Even more recently, methods based on so-called weak classifiers (Schapire, 1990) have been developed and a family of procedures called boosting procedures has been defined.

In practice, it is quite frequent to have some a priori knowledge on monotone relations among the predictors and the response groups. For example, in a cancer trial, it may be known that higher values of a predictor are associated with more advanced stages of the illness (Conde et al., 2012). One way of representing this information is to include order restrictions among the means of the predictors in the groups. This scheme allows to define rules (Fernández et al., 2006; Conde et al., 2012) that improve the performance of linear discriminant rules. Bootstrap estimators of the performance of these

rules have been provided in Conde et al. (2013) and the rules have been implemented in the R package dawai presented in Conde et al. (2015).

There are some other procedures where isotonicity has been considered to define classification procedures, or regression models that can be adapted for classification. In Auh and Sampson (2006) a logistic rule is defined isotonizing the boundaries of the original rule. Ghosh (2007) proposes a semiparametric regression model, first smoothing and then isotonizing the components, to evaluate biomarkers, while Meyer (2013) considers a more general semiparametric additive constrained regression model with more efficient estimation and inferential procedures. Another approach is that of Hofner et al. (2016), where a unified framework to incorporate restrictions for univariate and bivariate effects estimates using P-splines is proposed. Finally, Pya and Wood (2014) considers a penalized spline method for generalized shape-restricted (under monotonicity and concavity) additive models and Chen and Samworth (2016) proposes a non-parametric estimator of the additive components under the same setting. The problem of building isotonic classification rules has also attracted the attention of the machine learning community where it is known as monotonic ordinal classification. Many of the procedures developed from that area of research require that the training data set satisfies the monotonicity relationships and others consider preprocessing algorithms to "monotonize" the data set. See Cano and García (2017); Cano et al. (2019) and references therein for a complete overview of these procedures and how they work. Here, we will define procedures that neither need any preprocessing of the training data nor are obtained modifying a previous rule to obtain monotonicity. Their performance properties will come from the direct incorporation of monotonicity in their design.

The purpose of this paper is to develop isotonic boosting algorithms yielding classification rules that satisfy the monotone relations between the predictors and the response. Boosting algorithms are widely used nowadays. They are a family of iterative procedures that combine simple rules that may perform slightly better than random, to build an ensemble where the performance of the simple members is improved ("boosted"). Among these algorithms, we have decided to select LogitBoost (Friedman et al., 2000) as starting procedure in the development of our algorithms for the following reasons. In that paper, the authors explain the performance of AdaBoost, the first boosting algorithm of practical use (Freund and Schapire, 1996, 1997), considering the algorithm as a stage-wise descent procedure in an exponential loss function. In the same paper, they also design LogitBoost replacing the loss function by a logistic loss function, so that the binomial log-likelihood is directly optimized. This leads to changes in the weights assigned to misclassified observations in such a way that the misclassified observations that are

further from the boundaries are given in LogitBoost a smaller weight. Consequently, the procedure is less sensitive to outliers and noise than AdaBoost (Dietterich, 2000; McDonald et al., 2003) and we expect it to perform better in monotonic classification problems.

In this paper we develop two isotonic boosting procedures for binary classification based on LogitBoost, yielding rules that follow the known monotone relations in the problem at hand and are therefore more easily interpretable and efficient. The first procedure, that we call Simple Isotonic LogitBoost (SILB), selects in each step the variable that best fits the appropriate weighted regression problem taking isotonicity into account where needed. In the second, Multiple Isotonic LogitBoost (MILB), the whole problem is refitted in each step, so that all predictors change their role in the rule in each step, also considering isotonicity where needed.

Moreover, multiclass rules are also developed in this paper. Instead of decomposing this problem in multiple binary problems, which makes more difficult the consideration of isotonicity, we develop theoretical results that allow us to define procedures for the multinomial log-likelihood, based on two ordinal logistic models, namely the adjacent categories model and the cumulative probabilities model (see Agresti, 2010). For these two models again we develop simple and multiple isotonic LogitBoost algorithms that, following the previous notation, we call Adjacent-categories Simple Isotonic LogitBoost (ASILB), Adjacent-categories Multiple Isotonic LogitBoost (AMILB), Cumulative probabilities Simple Isotonic LogitBoost (CSILB) and Cumulative probabilities Multiple Isotonic LogitBoost (CMILB).

The layout of the paper is as follows. In Section 2 we recall the LogitBoost algorithm and present the new boosting algorithms developed from it, both for the binary and multiclass problems. We devote Section 3 to simulation studies showing that the new rules perform better than other up-to-date procedures in different scenarios. In Section 4 the results for two real data problems are presented. Finally, the discussion and future developments are exposed in Section 5.

2 Isotonic Boosting classification rules

Let us consider $K \geq 2$ classes and a training sample $\{(\mathbf{x}_i, y_i), i = 1, \ldots, n\}$, where $\mathbf{x}_i$ is a d-dimensional vector of predictors and $y_i \in \{1, \ldots, K\}$ the variable identifying the class. The aim is to classify a new observation $\mathbf{x}$ into one of the K classes.

Moreover we assume that it is known that higher values of some of the predictors are associated to higher values of the class variable y (monotone increasing) and that higher values of other predictor variables are associated with lower values of y (monotone decreasing). We denote as I the set of

indexes of the former set of predictors and as D the set of indexes of the latter one. Obviously, $I \cup D \subseteq \{1, \ldots, d\}$.

2.1 Binary classification rules

Here, we develop two new boosting algorithms based on LogitBoost that incorporate the known monotonicity information between the predictors and the response. Let us define $y_i^* = y_i - 1 \in \{0, 1\}, i = 1, \ldots, n$ and $p(\mathbf{x}) = p(y^* = 1|\mathbf{x})$. For the two-class problem LogitBoost is based on the logistic model:

$$\log \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} = F(\mathbf{x}). \tag{1}$$

The general version of a LogitBoost procedure considers $F(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{x})$ where M is the number of iterations of the procedure and $f_m(\mathbf{x})$ are the functions obtained in each iteration. In the most common versions of LogitBoost, each $f_m(\mathbf{x})$ function depends on a single predictor. These univariate functions make it easier to incorporate the isotonicity restrictions. For our first isotonic procedure, SILB, in model (1) we consider $F(\mathbf{x}) = \sum_{m=1}^{M} f_m(x_{j_m})$, while for MILB in (1) we consider $F(\mathbf{x}) = \sum_{j=1}^{d} f_j(x_j)$ where d is the number of predictors. In both cases, we impose that $f_m(x_s)$ or $f_j(x_s)$ is monotone increasing if $s \in I$ or decreasing if $s \in D$. These constraints imply that higher values of variables in I are associated with higher values of $p(\mathbf{x})$, i.e. of y, and that higher values of variables in D are associated with lower values of $p(\mathbf{x})$, i.e. of y.

Let us begin recalling the LogitBoost algorithm:

LogitBoost

1. Start with weights $w_i = 1/n, i = 1, \ldots, n$, $F(\mathbf{x}) = 0$ and probability estimates $p(\mathbf{x}_i) = \frac{1}{2}$ for $i = 1, \ldots, n$.
2. Repeat for $m = 1, \ldots, M$:
   (a) Compute $w_i = p(\mathbf{x}_i)(1 - p(\mathbf{x}_i))$, $z_i = \frac{y_i^* - p(x_i)}{w_i}$, $i = 1, \ldots, n$.
   (b) Fit $f_m(\mathbf{x})$ by a weighted least-squares regression of z_i to $\mathbf{x}_i$ using weights w_i.
   (c) Update $F(\mathbf{x}) = F(\mathbf{x}) + f_m(\mathbf{x})$ and $p(\mathbf{x}) = \frac{1}{1+e^{-F(\mathbf{x})}}$.
3. Classify in class 0 if $p(\mathbf{x}) < 0.5$, in class 1 if $p(\mathbf{x}) \geq 0.5$.

Although $f_m(\mathbf{x})$ in 2(b) can be obtained using any weighted least-squares regression method, Friedman et al. (2000) uses 2 and 8 terminal node regression trees when comparing the performances of LogitBoost and other algorithms. For simplicity, we will use 2 terminal node trees (stumps) when comparing the performances. If stumps are used, only one predictor variable

is incorporated in each $f_m(\mathbf{x})$. In this way we can reformulate the algorithm to incorporate the additional ordering information.

Next, two isotonic algorithms are proposed. We first present SILB, an algorithm based on LogitBoost, with 2(b) modified to incorporate the additional information as follows. For those variables j for which an monotonicity restriction holds (i.e. $j \in I \cup D$) a weighted isotonic regression, using the well known PAVA algorithm (Barlow et al., 1972; Robertson et al., 1988), is fitted instead of the usual weighted regression stump. Then, as usual, we choose for $f_m(\mathbf{x})$ the variable yielding the best weighted least squares fit among all.

Simple Isotonic LogitBoost (SILB)

1. Start with weights $w_i = 1/n, i = 1, \ldots, n$, $F(\mathbf{x}) = 0$ and probability estimates $p(\mathbf{x}_i) = \frac{1}{2}$ for $i = 1, \ldots, n$.
2. Repeat M times:
   (a) Compute $w_i = p(\mathbf{x}_i)(1 - p(\mathbf{x}_i))$, $z_i = \frac{y_i^* - p(\mathbf{x}_i)}{w_i}$, $i = 1, \ldots, n$.
   (b) Repeat for $j = 1, \ldots, d$:
   - If $j \in I \cup D$, fit a weighted isotonic regression $f_j(x)$ of z_i to x_{ij} using weights w_i.
   - If $j \notin I \cup D$, fit a 2 terminal node regression stump $f_j(x)$ by weighted least-squares of z_i to x_{ij} using weights w_i.

   (c) Consider $h = arg\ min_{j \in \{1,\ldots,d\}} \sum_{i=1}^{n} w_i (z_i - f_j(x_{ij}))^2$, and update
   $F(\mathbf{x}) = F(\mathbf{x}) + f_h(x_h)$ and $p(\mathbf{x}) = \frac{1}{1+e^{-F(\mathbf{x})}}$.
3. Classify in class 0 if $p(\mathbf{x}) < 0.5$, in class 1 if $p(\mathbf{x}) \geq 0.5$.

As in LogitBoost, the performance of the algorithm is expected to improve with the number of iterations M, although Mease and Wyner (2008) suggests that LogitBoost can also be prone to overfitting if M is large enough, which could as well happen with SILB.

Now we present MILB. This algorithm is also based on LogitBoost but in this case in each step we fit the whole model using a backfitting algorithm (Härdle and Hall, 1993) and considering weighted isotonic regression where needed. In LogitBoost (and subsequently in SILB), each new value of $F(\mathbf{x})$ was the sum of the old value of $F(\mathbf{x})$ plus the weighted expectation of the Newton step (see Friedman et al., 2000). For this algorithm, as in Hastie and Tibshirani (2006), each new value of $F(\mathbf{x})$ is the weighted expectation of the sum of the old value of $F(\mathbf{x})$ plus the Newton step.

Multiple Isotonic LogitBoost (MILB)

1. Start with weights $w_i = 1/n, i = 1, \ldots, n$, $F(\mathbf{x}) = 0$ and probability estimates $p(\mathbf{x}_i) = \frac{1}{2}$ for $i = 1, \ldots, n$.
2. Repeat until convergence:
   (a) Compute $w_i = p(\mathbf{x}_i)(1 - p(\mathbf{x}_i))$, $z_i = F(\mathbf{x}_i) + \frac{y_i^* - p(\mathbf{x}_i)}{w_i}, i = 1, \ldots, n$.
   (b) Using the backfitting algorithm, fit an additive weighted regression $F(\mathbf{x}) = \sum_{j=1}^{d} f_j(x_j)$ of z_i to $\mathbf{x}_i$ with weights w_i with $f_j(x)$ being the isotonic regression if $j \in I \cup D$, or the linear regression if $j \notin I \cup D$, $j = 1, \ldots, d$.
   (c) Update $p(\mathbf{x}) = \frac{1}{1+e^{-F(\mathbf{x})}}$.
3. Classify in class 0 if $p(\mathbf{x}) < 0.5$, in class 1 if $p(\mathbf{x}) \geq 0.5$.

Notice that, as $F(\mathbf{x})$ is now the addition of a fixed number of d terms, overfitting is not expected to appear.

2.2 Multiclass classification

Let us denote $p_k(\mathbf{x}) = P(y = k|\mathbf{x})$, $y_k^* = I_{[y=k]}, k = 1, \ldots, K$, and I and D as in Section 2.1. The most common way of tackling multiclass classification problems is to split the problem in multiple binary problems. Among the several ways of doing this, the most common (Allwein et al., 2000; Holmes et al., 2002) are: One-against-rest, where K binary classifications for each class against all others are considered, and One-against-one, which considers every pair of classes and performs $\binom{K}{2}$ binary classifications. However, direct multiclass procedures that fit a single model can also be used. They exhibit a performance comparable to (if not better than) those of these multiple binary problems strategies, and are more appropriate for the incorporation of the information about monotone relationships since they allow considering the full monotonicity existing among all the classes and not only the pairwise order. For these reasons, in this paper we will consider multiclass procedures instead of combinations of binary ones.

Moreover, models such as the baseline category model (see Agresti, 2002) where each category is compared with a baseline are not appropriate to incorporate monotone relationships among the categories. In our setting, the response variable is ordinal and models where this characteristic is taken into account should be considered. The two most widely used models for ordinal responses are the cumulative logit and the adjacent-categories logit models (Agresti, 2010).

The adjacent-categories logit model is

$$\log \frac{p_k(\mathbf{x})}{p_{k-1}(\mathbf{x})} = F_k(\mathbf{x}), \text{for } k = 2, \ldots, K, \tag{2}$$

while the cumulative logit model is

$$\log \frac{\sum_{i=k}^{K} p_i(\mathbf{x})}{\sum_{i=1}^{k-1} p_i(\mathbf{x})} = F_k(\mathbf{x}), \text{for } k = 2, \ldots, K. \tag{3}$$

It is known (see Agresti, 2010, p.70) that the cumulative logit model (3) has structural problems as the cumulative probabilities may be out of order at some setting of the predictors. To avoid this problem we will adopt, as in the above reference, the parallelism restriction for this model and assume that, in this case, $F_k(\mathbf{x}) = \alpha_k + F(\mathbf{x})$, $k = 2, \ldots, K$.

The choice between these models is an interesting question that has been considered not only in Agresti (2010) but also in more recent books as Fullerton and Xu (2016). There are some technical reasons for preferring each model. For example, in the cumulative probabilities model the sample size used for fitting each equation does not change among equations, while the adjacent categories model belongs to the exponential family. However, according to the above-mentioned authors and from a practical point of view, the main reason for choosing among the models is the probability of interest in the problem. If the individual response categories are of substantive interest (as, for example, with Likert scales) the adjacent categories model is recommended, while the cumulative probabilities model is preferred when these cumulative probabilities are of interest. For these reasons, both models have been used in health research (Marshall, 1999; Fullerton and Anderson, 2013), cumulative probabilities models in, among many other, studies about worker attachment (Halaby, 1986) or attitudes towards science (Gauchat, 2011), and adjacent categories models in, for example, occupational mobility (Sobel et al., 1998) or credit scoring (Masters, 1982).

Now, we develop our isotonic multiclass boosting procedures. We start with the adjacent-categories model (2). For this model the constraints imply that the higher the values of variables in I the greater the $p_k(\mathbf{x})$ with respect to $p_{k-1}(\mathbf{x})$, and the higher the values of variables in D the lower the $p_k(\mathbf{x})$ with respect to $p_{k-1}(\mathbf{x})$, so that high values of variables in I are associated with high values of y and high values of variables in D are associated with low values of y. As in the binary case, we propose two procedures for this model. These proposals require the development of theoretical results that can be found in Appendix A.1.

Our first proposal for this model is the ASILB procedure, where we assume that $F_k(\mathbf{x}) = \sum_{m=1}^{M} f_{km}(x_{j_{km}})$, imposing $f_{km}(x)$ to be isotonic if $j_{km} \in I \cup D$ and we add terms fitting the new quasi-Newton steps according to the results in Appendix A.1. The reason to use quasi-Newton steps instead of full Newton steps in this case is the use of stumps when the predictor $j \notin I \cup D$.

Adjacent-categories Simple Isotonic LogitBoost (ASILB)

1. Start with $F_k(\mathbf{x}) = 0, k = 1, \ldots, K$, and $p_k(\mathbf{x}) = \frac{1}{K}, k = 1, \ldots, K$.
2. Repeat M times:
   (a) Let $\mathbf{W}_i$ be a diagonal $(K-1) \times (K-1)$ matrix, where each diagonal element is $W_{i_{kk}} = (\sum_{j=k}^{K} p_j(\mathbf{x}_i))(1 - \sum_{j=k}^{K} p_j(\mathbf{x}_i))$, $2 \leq k \leq K$. Define also vector $\mathbf{S}_i$ with $S_{ik} = \sum_{j=k}^{K}(y_{ij}^* - p_j(\mathbf{x}_i))$ for $2 \leq k \leq K$ and compute $\mathbf{z}_i = \mathbf{W}_i^{-1}\mathbf{S}_i, i = 1, \ldots, n$.
   (b) For $k = 2, \ldots, K$:
   Repeat for $j = 1, \ldots, d$:
   – If $j \in I \cup D$, fit a weighted isotonic regression $f_j(x)$ of z_{ik} to x_{ij} using weights $W_{i_{kk}}$.
   – If $j \notin I \cup D$, fit a 2 terminal node regression stump $f_j(x)$ by weighted least-squares of z_{ik} to x_{ij} using weights $W_{i_{kk}}$.
   Consider $h = arg\ min_{j \in \{1,\ldots,d\}} \sum_{i=1}^{n} W_{i_{kk}}(z_{ik} - f_j(x_{ij}))^2$, and update $F_k(\mathbf{x}) = F_k(\mathbf{x}) + f_h(x_h)$.
   (c) Update $p_k(\mathbf{x}) = \frac{\exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}{\sum_{k=1}^{K} \exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}, k = 1, \ldots, K$.
3. Classify in class $h = arg\ max_{k \in \{1,\ldots,K\}} p_k(\mathbf{x})$.

For our second proposal for model (2), AMILB, we assume that $F_k(\mathbf{x}) = \sum_{j=1}^{d} f_{kj}(x_j)$, imposing $f_{kj}(x)$ to be isotonic if $j_{km} \in I \cup D$. As in MILB, we fit the whole model in each iteration using a backfitting algorithm and the results obtained in Appendix A.1. In this case, instead of using a quasi-Newton step, as in ASILB, we use the full Newton step, since in this algorithm we consider linear or isotonic regression for which non-diagonal weights can be easily included. For this reason in this algorithm the weight matrix is not diagonal.

Adjacent-categories Multiple Isotonic LogitBoost (AMILB)

1. Start with $F_k(\mathbf{x}) = 0, k = 1, \ldots, K$, and $p_k(\mathbf{x}) = \frac{1}{K}, k = 1, \ldots, K$. Let $\mathbf{F}(\mathbf{x}) = [F_2(\mathbf{x}), \ldots, F_K(\mathbf{x})]'$.
2. Repeat until convergence:
   (a) Let $\mathbf{W}_i$ be a $(K-1) \times (K-1)$ matrix, where each element is:

$$W_{i_{km}} = \begin{cases} (\sum_{j=m}^{K} p_j(\mathbf{x}_i))(1 - (\sum_{j=k}^{K} p_j(\mathbf{x}_i)), \text{ if } m \geq k \\ (\sum_{j=k}^{K} p_j(\mathbf{x}_i))(1 - (\sum_{j=m}^{K} p_j(\mathbf{x}_i)), \text{ if } m < k \end{cases}$$

   for $2 \leq k, m \leq K$, and for $i = 1, \ldots, n$ compute $\mathbf{z}_i = \mathbf{F}(\mathbf{x}_i) + \mathbf{W}_i^{-1}\mathbf{S}_i$ with $\mathbf{S}_i$ the vector defined in step 2(a) of the ASILB algorithm .

(b) Using the backfitting algorithm, fit an additive weighted regression $\mathbf{F}(\mathbf{x}) = \sum_{j=1}^{d} f_j(x_j)$ of $\mathbf{z}_i$ to $\mathbf{x}_i$ with weights matrices $\mathbf{W}_i$, where $f_j(x)$ is the isotonic regression if $j \in I \cup D$ or the linear regression if $j \notin I \cup D$, $j = 1, \ldots, d$.

(c) Update $p_k(\mathbf{x}) = \frac{\exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}{\sum_{k=1}^{K} \exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}, k = 1, \ldots, K$.

3. Classify in class $h = arg\ max_{k \in \{1,\ldots,K\}} p_k(\mathbf{x})$.

Now, we consider the cumulative model (3) and describe the corresponding CSILB and CMILB algorithms. In this case, due to the parallelism assumption, we have a single function $F(\mathbf{x})$ but we additionally have to estimate the α_k, $k = 2, \ldots, K$ parameters. For this reason the algorithms are more involved. Their theoretical justification can be found at Appendix A.2.

Cumulative probabilities Simple Isotonic LogistBoost (CSILB)

1. Start with $F(\mathbf{x}) = 0$ and $p_k(\mathbf{x}) = \frac{1}{K}, k = 1, \ldots, K$, so that $\alpha_k = -\log \frac{k-1}{K-k+1}, k = 2, \ldots, K$. Denote as $\boldsymbol{\alpha} = (\alpha_2, \ldots, \alpha_K)'$ and let $\gamma_k(\mathbf{x}) = \sum_{j=k}^{K} p_j(\mathbf{x}), k = 1, \ldots, K$ with $\gamma_{K+1}(\mathbf{x}) = 0$.
2. Repeat M times:
   (a) Compute $w_i = \sum_{k=1}^{K} y_{k,i}^* [\gamma_k(\mathbf{x}_i)(1 - \gamma_k(\mathbf{x}_i)) + \gamma_{k+1}(\mathbf{x}_i)(1 - \gamma_{k+1}(\mathbf{x}_i))]$, $s_i = \sum_{k=1}^{K} y_{k,i}^* [1 - \gamma_k(\mathbf{x}_i) - \gamma_{k+1}(\mathbf{x}_i)], i = 1, \ldots, n$ and $z_i = w_i^{-1} s_i$.
   (b) Repeat for $j = 1, \ldots, d$:
   - If $j \in I \cup D$, fit a weighted isotonic regression $f_j(x)$ of z_i to x_{ij} using weights w_i.
   - If $j \notin I \cup D$, fit a 2 terminal node regression stump $f_j(x)$ by weighted least-squares of z_i to x_{ij} using weights w_i.

   Consider $h = arg\ min_{j \in \{1,\ldots,d\}} \sum_{i=1}^{n} w_i (z_i - f_j(x_{ij}))^2$, and update $F(\mathbf{x}) = F(\mathbf{x}) + f_h(x_h)$.
   (c) Consider the $K - 1$ dimensional score vector $\mathbf{S} = (s_k)$ defined in (4) and the $(K-1) \times (K-1)$ Hessian matrix $\mathbf{H}$ defined in (5) to (7) in Appendix A.2. Compute $\boldsymbol{\alpha} = \boldsymbol{\alpha} - \mathbf{H}^{-1}\mathbf{S}$.
   (d) Update $\gamma_k(\mathbf{x}) = \frac{\exp(\alpha_k + F(\mathbf{x}))}{1 + \exp(\alpha_k + F(\mathbf{x}))}$ for $k = 2, \ldots, K$, and $p_k(\mathbf{x}) = \gamma_k(\mathbf{x}) - \gamma_{k+1}(\mathbf{x})$ for $k = 1, \ldots, K$.
3. Classify in class $h = arg\ max_{k \in \{1,\ldots,K\}} p_k(\mathbf{x})$.

Finally, we present the corresponding CMILB algorithm also based on the results developed at Appendix A.2. In this case the algorithm is more similar to CSILB than what happened for model (2).

Cumulative probabilities Multiple Isotonic LogistBoost (CMILB)

1. As in CSILB, start with $F(\mathbf{x}) = 0$ and $p_k(\mathbf{x}) = \frac{1}{K}, k = 1, \ldots, K$, so that $\alpha_k = -\log \frac{k-1}{K-k+1}, k = 2, \ldots, K$. Let $\gamma_k(\mathbf{x}) = \sum_{j=k}^{K} p_j(\mathbf{x}), k = 1, \ldots, K$, and $\gamma_{K+1}(\mathbf{x}) = 0$.
2. Repeat until convergence:
   (a) Compute w_i and s_i as in CSILB step 2(a) and let $z_i = F(x_i) + w_i^{-1} s_i$.
   (b) Using the backfitting algorithm, fit an additive weighted regression $F(\mathbf{x}) = \sum_{j=1}^{d} f_j(x_j)$ of z_i to $\mathbf{x}_i$ with weights w_i with $f_j(x)$ being the isotonic regression if $j \in I \cup D$, or the linear regression if $j \notin I \cup D$, $j = 1, \ldots, d$.
   (c) Perform same computations as in CSILB step 2(c).
   (d) Update $\gamma_k(\mathbf{x})$ and $p_k(\mathbf{x})$ as in CSILB step 2(d).
3. Classify in class $h = arg\ max_{k \in \{1,\ldots,K\}} p_k(\mathbf{x})$.

It can be checked that the multiclass simple LogitBoost rules (ASILB and CSILB) and the multiclass multiple LogitBoost rules (AMILB and CMILB) defined in this subsection coincide, respectively, in the case $K = 2$ with the corresponding SILB and MILB two class rules defined in the subsection 2.1.

2.3 Rules implementation in R

We have developed an R (R Core Team, 2018) package, called isoboost (Conde et al., 2020) and available from the Comprehensive R Archive Network (CRAN), which provides the functions asilb, csilb, amilb and cmilb implementing the correspoding procedures developed in this paper.

These functions depend on the R packages rpart (Therneau and Atkinson, 2019) for performing weighted regression trees with functions rpart and predict.rpart, Iso (Turner, 2019) for performing isotonic regression with function pava, and isotone (De Leeuw et al., 2009) for performing weighted isotonic regression with function gpava.

3 Simulation study

In this section we present the results of the simulation studies we have performed to evaluate the behavior of the new proposed methods. The full set of methods used in the simulation study can be found in Table 1 and the R code used to perform them is contained in the Supplementary material section of the paper.

These methods include not only standard up-to-date procedures such as LDA, logistic regression random forest, SVM or Logitboost, but also methods that account for monotonicity such as restricted LDA (Conde et al., 2012) or the monotone version of XGBoost (Chen and Guestrin, 2016), which is one

of the procedures more widely used nowadays. The R packages considered for these procedures are as follows. MASS (Venables, 2002) has been used for performing LDA, nnet (Venables, 2002) for performing LOGIT, randomForest (Liaw and Wiener, 2002) for performing RF, e1071 (Meyer et al., 2019) for performing SVM, caTools (Tuszynski, 2019) for performing LGB, dawai (Conde et al., 2015) for performing RLDA, and xgboost (Chen et al., 2019) for performing MONOXGB.

Method	Acronym
Linear discriminant analysis	LDA
Logistic regression	LOGIT
Random forest	RF
Support vector machines	SVM
Logitboost	LGB
Restricted linear discriminant analysis	RLDA
Monotone extreme gradiesnt boosting	MONOXGB
Adjacent-categories Simple Isotonic LogitBoost	ASILB
Adjacent-categories Multiple Isotonic LogitBoost	AMILB
Cumulative probabilities Simple Isotonic LogistBoost	CSILB
Cumulative probabilities Multiple Isotonic LogistBoost	CMILB

Table 1: Methods used in the simulations and their acronyms.

To compare the results from the procedures considered we have used several performance criteria. First, we have considered the total misclassification probability (TMP), i.e. the percentage of misclassified observations, which is equivalent to using a 0-1 loss and is the most commonly used performance measure. We have also considered the mean absolute error (MAE). MAE is a performance measure frequently used (see Cano et al., 2019) when evaluating monotone procedures as it also takes into account the "distance" between the observed and predicted values of the response. It is computed as $MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$, where $\hat{y}_i$ is the predicted class for observation i. Obviously, MAE equals TMP in the binary case. The third performance measure we have considered is the well-known area under the ROC curve (AUC). The multiclass version of this measure is defined in Hand and Till (2001). Notice that, while lower values of TMP and MAE indicate a better performance of the rule, the opposite happens for AUC.

Two different schemes of simulations designed following the lines considered in other related papers are considered. The first one is based on the adjacent categories model (2), while, for the second, a model where the means of the predictors in the groups follow a known order is considered. Table 2 shows the characteristics of each of these two schemes. Notice that, in each scheme, we generate a total of 100 datasets for each combination of number of classes K and predictors d.

The first scheme of simulations is based on model (2) instead of on the cumulative logit model (3) since the former is more flexible, not requiring the parallelism between the $F_j(\mathbf{x})$ functions. The functions considered are

	Scheme 1	Scheme 2
# classes K	2, 3, 5	2, 3, 5
# predictors d	5, 10	5, 10
# training samples	100	100
Training sample size n	$20K$	$10K$, $20K$
Test sample size	$50K$	$50K$
Distribution of predictors	$X_i \sim U(-1,1)$	$X_i \sim U(-1,1) + 0.2h$
Response	$Y \sim Mult(1,(p_1(\mathbf{x}),\ldots,p_K(\mathbf{x})))$ with $p_k(\mathbf{x}) = \frac{\exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}{\sum_{k=1}^{K}\exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}$	$Y = h$

Table 2: Conditions of the simulations. The F_j functions are given in Table 3.

given in Table 3. Different monotone increasing additive functions in $\mathbf{x}$ (polynomial, logarithmic, exponential) have been included. Simulations schemes similar to this one can be found, for example, in Bühlmann (2012); Chen and Samworth (2016); Dettling and Bühlmann (2003); Friedman et al. (2000). For the predictors in this scheme we have generated d-dimensional vectors $\mathbf{x}$ from a $U(-1,1)^d$ distribution. For the response we consider the $F_j(\mathbf{x})$ functions and compute $p_k(\mathbf{x})$ for $k = 1,\ldots,K$ (see Table 2). Then, we generate a single observation from a multinomial distribution with probability vector $(p_1(\mathbf{x}_i),\ldots,p_K(\mathbf{x}_i))$ and take Y as the index where the observation appears.

The mean results obtained with the three performance measures considered (TMP, MAE and AUC) are qualitatively similar and, for this reason, in the main text we only detail the TMP results while the full numerical values of the three measures are given in Appendix A.3 to improve the readability of the paper. The TMP results for the first simulations scheme are shown in Figure 1. Figure 1 shows, especially for $K = 2$ and $K = 5$, that the rules that incorporate additional information (ASILB, AMILB, CSILB, CMILB and RLDA) outperform not only the rules that do not account for this information (RF, SVM, LogitBoost, logistic regression and LDA) but also MONOXGB. We can also see that rules CSILB and CMILB perform as well as ASILB and AMILB although the cumulative logit model (3) under which the former algorithms were developed is not the one used for the simulations.

For the second set of simulations, uniform distributions for the predictors, as in Fang and Meinshausen (2012) or Bühlmann (2012), have been con-

K	j	$F_j^{K,1}$	$F_j^{K,2}$
2	2	$\sum_{i=1}^{3} x_i^3 + \sum_{i=4}^{5}\left(e^{x_i^3}-1\right)$	$\sum_{i=1}^{3} x_i^5 + \sum_{i=4}^{5}\left(\frac{1}{1+e^{-x_i^3}}-0.5\right)$
3	2	$F_2^{2,1} + 0.15$	$F_2^{2,2} + 0.15$
	3	$\sum_{i=1}^{3}\log(x_i+1) + \sum_{i=4}^{5} x_i^5 + 0.25$	$\sum_{i=1}^{3}\log(x_i+1) + \sum_{i=4}^{5} x_i^3 + 0.3$
5	2	$F_2^{2,1} + 0.55$	$F_2^{2,2} + 0.35$
	3	$F_3^{3,1} + 0.85$	$F_2^{2,2} + 0.8$
	4	$\sum_{i=1}^{5} x_i^2 - 0.2$	$F_4^{5,1}$
	5	$e^{x_1^3} - 1 + \frac{1}{1+e^{-x_2^3}} - 0.5 + \log(x_3+1) + x_4^3 + x_5^5 - 0.5$	$F_5^{5,1}$

Table 3: Values of the two sets ($s = 1,2$) of F_j functions considered in the first scheme of simulations under model (2), for different values of K and $d = 5$. For $d = 10$, $F_j^{K,s}(x_1,\ldots,x_{10}) = F_j^{K,s}(x_1,\ldots,x_5) + F_j^{K,s}(x_6,\ldots,x_{10})$.

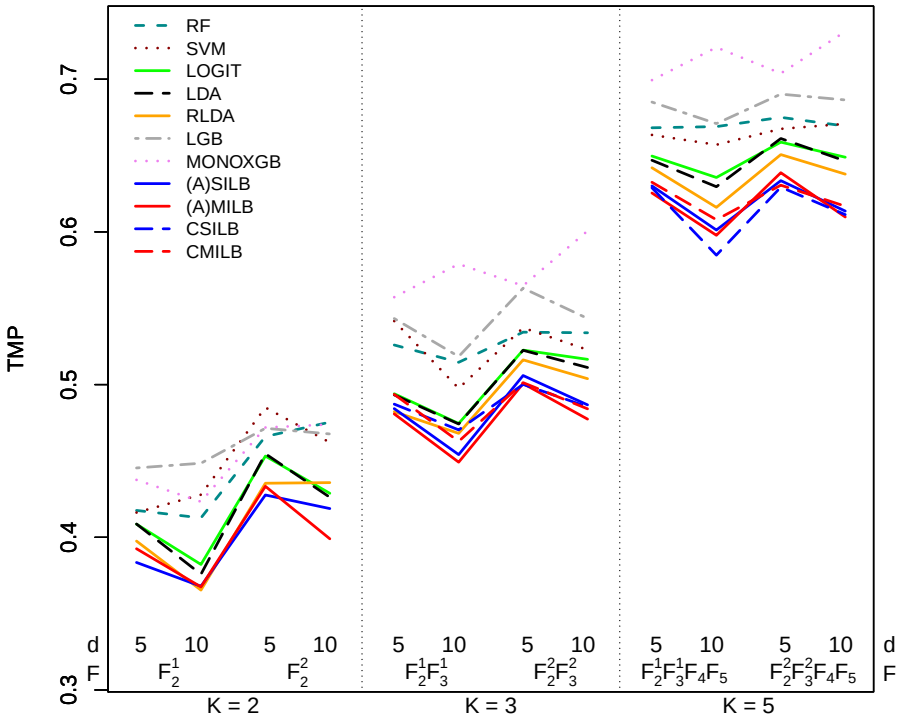

Fig. 1: TMP for the first simulation scheme for different classification rules, number of groups K, predictors d and functions F.

sidered. Independent $U(-1,1)+0.2h$ distributions are used to generate the predictors X_j, $j=1,\ldots,d$ for observations in class $Y=h$, representing a simple order among the means all d predictors with respect to the K classes. Full details are given in Table 2.

As in the first scheme, the results with the three performance measures are qualitatively similar and we only detail here the mean TMP results while we include the full numerical mean results for all measures in Appendix A.3. The mean TMP results obtained in this second set of simulations are shown in Figure 2.

We can see that the rules defined in this paper outperform clearly again the ones that do not take into account the additional information present in the data, and that they also improve over RLDA and MONOXGB, which are rules that take into account the monotonicity information available on the order of the means. In fact, under this scheme the new defined rules improve over RLDA more than they did for the first scheme where the difference with the new rules was smaller.

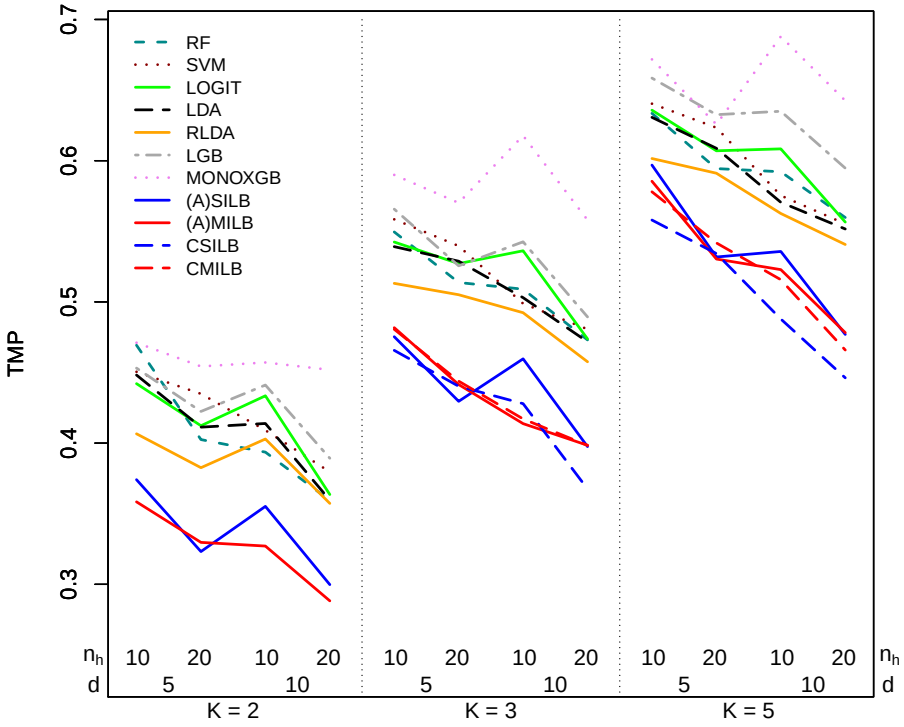

Fig. 2: TMP for different classification rules, number of groups k, predictors d and training sample sizes n, for the second simulation scheme.

4 Real data examples

In this section we evaluate the performance of the new rules in two problems we have encountered in our statistical practice. The first appears in a medical context where we were trying to find a non-invasive diagnostic kit for bladder cancer. In the second, we were considering an interesting industrial engineering problem, namely the diagnostic of electrical induction motors.

4.1 Bladder cancer diagnostic

The correct diagnostic and classification of cancer patients in the appropriate class is essential to provide them with the correct treatment. It is also very relevant that this diagnostic can be done with a procedure as non-invasive as possible. These were the main motivations of the work that we developed with Proteomika S.L. and Laboratorios SALVAT S.A. as industrial and pharmaceutical partners. In that work we tried to build diagnostic kit for several types of cancer. The bladder cancer data, already considered in Conde et al. (2012), is analyzed here. Neither the values nor the names of the predictors considered are given for confidential issues.

The patients were initially classified in 5 classes. The first level is the control level (patients that did not have the illness) and the other four levels are Ta, T1G1, T1G3 and T2, corresponding to increasingly advanced levels of illness. As a first step, a pilot study with a moderate number of patients was developed previous to a possible larger scale multicenter study. First, we received a 141 patient dataset that we consider as training set and later on a second sample of 149 different patients that we will use as test set was provided. As the sample sizes in some groups were small compared with those of the others, and according to our partners, we decided to merge the initial 5 levels in 3 groups, namely the control group, the Ta+T1G1 group and the T1G3+T2 group. Also according to information given by our partners we decided to consider 5 proteins as predictors. The mean values of each of these proteins were expected to increase with the illness level. The performance results for the test set obtained with the different methods considered throughout the paper appear in Table 4 with the best results marked in bold.

Measure	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	ASILB	AMILB	CSILB	CMILB
TMP	.617	.409	.738	.805	.785	.758	.785	.403	**.356**	.362	.362
MAE	.732	.510	.846	.799	.913	1.044	.980	.490	.443	.463	**.436**
AUC	.527	.718	.478	.582	.510	.446	.510	.706	**.721**	.717	.717

Table 4: Performance for the different procedures for the bladder cancer test dataset.

The results for RLDA, ASILB, AMILB, CSILB and CMILB, that take into account the order restrictions are much better than the rest. The main reason is that some of the predictors did not verify the restrictions in the training set. Therefore, we can see that these new methods are able to cope with a 'bad' training sample and obtain reasonable results without dropping any observations or manipulating the original data in that sample. This does not happen with MONOXGB. We can also see that the new methods also outperform RLDA.

4.2 Diagnostic of electrical induction motors

Electrical induction motors are widely used in industry. In fact, as Garcia-Escudero et al. (2017) points out, it is estimated that these motors account for 80% of energy converted in trade and industry. For this reason, and for the losses that a possible unexpected shutdown might yield, it is important to be able to detect possible failures in these machines as early as possible.

There are many techniques to diagnose a faulty motor, see Choudhary et al. (2019) and references therein. The most widespread is based on the spectral analysis of the stator current and is usually known as Motor Current Signature Analysis. The underlying principle is that motor faults cause an

asymmetry that is reflected as additional harmonics in the current spectrum. Therefore, side bands around the main frequency are considered and amplitudes of these side bands around odd harmonics are measured as predictors of damage severity of the motor. Lower values of the amplitudes are expected to be related to higher levels of damage severity (see Figure 3 for a graphic description of these variables). Four condition states were considered with state 1 corresponding to an undamaged motor, state 2 to a motor with an incipient fault, state 3 with a moderately damaged motor and state 4 with a severely damaged motor.

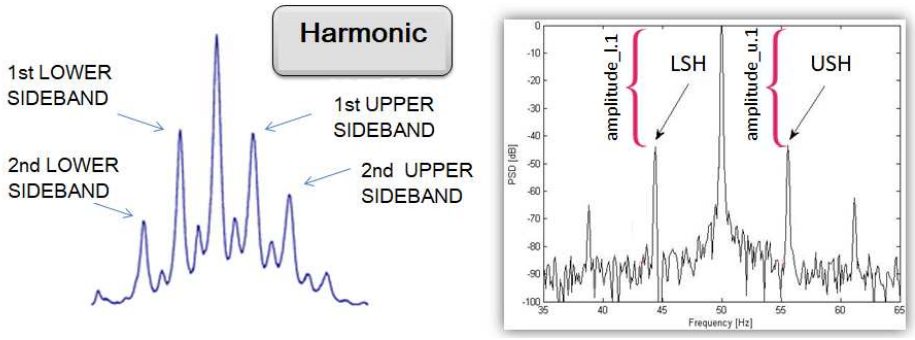

Fig. 3: Graphic representation of the predictors variables for the motor diagnostic example.

Here, we consider a sample of 280 motor observations, for which the real motor state was known, that were recorded at the Electrical Engineering Department laboratory of the Universidad de Valladolid. The distribution of these observations among the different groups appears in Table 5. Three variables, namely the amplitudes of the first lower and upper side bands around harmonic 1 and the first lower side band around harmonic 5 are the predictors. Three different classifications problems are solved. The first one considers the four different states, the second considers three states, joining states 2 and 3, and in the third the problem is to classify in group 1 vs the rest of groups. The second problem is interesting from the industrial point of view since in this case we are distinguishing healthy and incipiently or moderately damaged motors from those in a state that may cause operative problems. In the third undamaged motors are distinguished from those with any kind of damage. As no test sample is available here, the TMP estimators are obtained using 10-fold crossvalidation. The code used for performing this analysis is contained in the Supplementary material section of the article while the data can be found in the isoboost (Conde et al., 2020) package.

The results for these three classification problems appear in Table 6. We can see again that the methods proposed in this paper perform very well in this case and that the best result is obtained with one of these new methods.

Group	Observations
State 1 (Undamaged)	83
State 2 (Incipient fault)	67
State 3 (Moderate damage)	70
State 4 (Severe damage)	60

Table 5: Number of observations in each group for the motor diagnostic example.

In this case the isotonic boosting methods perform much better than RLDA which in turn yields same results than LDA. This happens when the training sample fulfills the isotonicity restrictions imposed. Therefore, we can see that the new isotonic methods proposed here improve even when the training sample verifies the restrictions. We can also see that, in this example, the unrestricted methods perform well from the AUC point of view and that some of them perform slightly better than the monotone methods.

Scheme	Measure	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	(A)SILB	(A)MILB	CSILB	CMILB
4 classes	TMP	.179	.179	.143	.143	.143	.151	.146	**.136**	.146	.164	.143
	MAE	.179	.179	**.143**	**.143**	**.143**	.152	.161	.146	.146	.164	**.143**
	AUC	.966	.966	.972	.971	**.976**	.958	.966	.970	.963	.953	.960
3 classes	TMP	.232	.232	.143	.136	.143	.133	.125	.125	.121	.139	**.114**
	MAE	.232	.232	.143	.136	.143	.133	.125	.125	.121	.139	**.114**
	AUC	.946	.946	.967	.964	**.968**	.953	.957	.958	.963	.957	.961
2 classes	TMP	.129	.129	.043	.043	.046	.043	.046	**.036**	**.036**		
	AUC	.967	.967	**.995**	.974	.987	.990	.985	.990	.993		

Table 6: Performance for the different procedures and classification problems for the induction motor data.

5 Discussion

In this paper, classification problems in scenarios where there are monotone relationships among predictors and classes are considered, and the idea of using isotonic regression, instead of standard regression, in boosting classification rules, is exploited.

From a methodological point of view, the specific contribution of this paper is the definition of novel rules developed for binary and multiclass classification problems. Theoretical results that endorse the classification rules based on maximum likelihood estimation are developed and simulations results, performed under different scenarios, validate the rules.

From a practical point of view, two real problems in different contexts have been efficiently solved using the new rules. In the first one, where cancer patients are classified in different diagnostic groups, a deficient training sample that does not verify the expected monotone relationships is available, so that standard procedures yield very high misclassification errors. In this case, the incorporation of the isotonicity information is compulsory, not only to get more efficient results but also to obtain meaningful ones. The new rules reduce the error rates between 33% and 66%. In the second case, that deals with the diagnostic of induction motors, the training sample fulfills the

expected monotone relationships and the error rates are quite low. Also in this case the new rules manage to reduce the error rates significantly.

The question of computational efficiency and scalability of statistical procedures is also interesting nowadays. We have performed a study recording the time consumed by the procedures considered in the paper in the simulations and we have found that the new procedures have a behavior similar to the other ones considered when the sample size of the dataset or the number of predictors is increased. When the number of classes increases we have found that CSILB and CMILB are also competitive when compared with previously existent procedures and more efficient than ASILB and AMILB.

Future developments that involve the procedures exposed here will include new ways of expressing the additional information, the incorporation of other type of additional information, such as concavity, and the consideration of other methodology, for example isotonic regression splines, in the design of the rules.

Acknowledgments

The authors thank the Associate Editor and two anonymous reviewers for suggestions that led to this improved version of the paper.

References

Agresti, A. (2002). *Categorical Data Analysis*. John Wiley and Sons. New Jersey.

Agresti, A. (2010). *Analysis of Ordinal Categorical Data, 2nd edition*. John Wiley and Sons. New Jersey.

Allwein, E. L., Schapire, R. E., and Singer, Y. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, Vol.1(Dec), pp. 113–141.

Auh, S., and Sampson, A. R. (2006). Isotonic logistic discrimination. *Biometrika*, Vol. 93(4), pp. 961–972.

Barlow, R. E., Bartholomew, D. J., Bremner, J. M. and Brunk, H. D. (1972). *Statistical Inference Under Order Restrictions*. John Wiley and Sons. New York.

Bühlmann, P. (2012). Bagging, Boosting and Ensemble Methods, in *Handbook of Computational Statistics, Springer*. Chapter 33, pp. 985–1022.

Cano, J. R., and García, S. (2017). Training set selection for monotonic ordinal classification. *Data & Knowledge Engineering*, Vol. 112, pp. 94–105.

Cano, J. R., Gutiérrez, P. A., Krawczyk, B., Wozniak, M., and García, S. (2019). Monotonic classification: An overview on algorithms, performance measures and data sets. Neurocomputing, 341, 168-182.

Chen, T., and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., and Li Y. (2019). xgboost: Extreme Gradient Boosting. R package version 0.82.1, URL `https://CRAN.R-project.org/package=xgboost`.

Chen, Y., and Samworth, R. J. (2016). Generalized additive and index models with shape constraints. *J. R. Statist. Soc. B*, Vol. 78, pp. 729–754.

Choudhary, A., Goyal, D., Shimi, S. L., and Akula, A. (2019). Condition monitoring and fault diagnosis of induction motors: A review. Archives of Computational Methods in Engineering. In press. doi:10.1007/s11831-018-9286-z.

Conde, D., Fernández, M. A., Rueda, C., and Salvador, B. (2012). Classification of samples into two or more ordered populations with application to a cancer trial. *Statistics in Medicine*, Vol. 31(28), pp. 3773–3786.

Conde, D., Fernández, M. A., Rueda, C., and Salvador, B. (2020). isoboost: Isotonic Boosting Classification Rules. R package version 1.0.0, URL `https://CRAN.R-project.org/package=isoboost`.

Conde, D., Salvador, B., Rueda, C., and Fernández, M. A. (2013). Performance and estimation of the true error rate of classification rules built with additional information. An application to a cancer trial. *Statistical Applications in Genetics and Molecular Biology*, Vol. 12(5), pp. 583–602.

Conde, D., Fernández, M. A., Salvador, B., and Rueda, C. (2015). dawai: An R Package for Discriminant Analysis With Additional Information. *Journal of Statistical Software*, Vol. 66(10), pp. 1–19.

De Leeuw, J., Hornik, K., and Mair, P. (2009). Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods. *Journal of Statistical Software*, Vol. 32(5), 1–24. URL `http://www.jstatsoft.org/v32/i05/`.

Dettling, M., and Bühlmann, P. (2003). Boosting for tumor classification with gene expression data. *Bioinformatics*, Vol. 19(9), pp. 1061–1069.

Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, Vol. 40(2), pp. 139–157.

Fang, Z. and Meinshausen, N. (2012). LASSO Isotone for High-Dimensional Additive Isotonic Regression. *Journal of Computational and Graphical Statistics* vol. 21:1, pp. 72–91.

Fernández, M. A., Rueda, C., and Salvador, B. (2006). Incorporating additional information to normal linear discriminant rules. *Journal of the American Statistical Association*, Vol. 101, pp. 569–577.

Freund, Y., and Schapire, R. E. (1996). Experiments with a new boosting algorithm. *ICML'96 Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, pp. 148–156.

Freund, Y., and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, Vol. 55(1), pp. 119–139.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive Logistic Regression: a statistical view of Boosting. *The Annals of Statistics*, Vol. 28(2), pp. 337–407.

Fullerton, A. S., and Anderson, K. F. (2013). The role of job insecurity in explanations of racial health inequalities. *Sociological Forum*, Vol. 28(2), pp. 308–325.

Fullerton, A. S., and Xu, J. (2016). *Ordered regression models: Parallel, partial, and non-parallel alternatives*. CRC Press. Boca Raton, FL.

Garcia-Escudero, L. A., Duque-Perez, O., Fernandez-Temprano, M., and Moriñigo-Sotelo, D. (2017). Robust detection of incipient faults in VSI-fed induction motors using quality control charts. IEEE Transactions on Industry Applications, Vol. 53(3), pp. 3076–3085.

Gauchat, G. (2011). The cultural authority of science: Public trust and acceptance of organized science. *Public Understanding of Science*, Vol. 20(6), pp. 751–770.

Ghosh, D. (2007). Incorporating monotonicity into the evaluation of a biomarker. *Biostatistics*, Vol. 8(2), pp. 402–413.

Halaby, C. N. (1986). Worker attachment and workplace authority. *American Sociological Review*, Vol. 51(5), pp. 634–649.

Hand, D. J., and Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classication problems. *Machine Learning*, Vol. 45, pp. 171–186.

Härdle, W. and Hall, P. (1993). On the backfitting algorithm for additive regression models. *Statistica Neerlandica*, Vol. 47, pp. 43–57.

Hastie, T., and Tibshirani, R. (2006). Generalized additive models. *Encyclopedia of Statistical Science*.

Hofner, B., Kneib, T., and Hothorn, T. (2016). A unified framework of constrained regression. *Statistics and Computing*, Vol. 26(1-2), pp. 1–14.

Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., and Hall, M. (2002). Multiclass alternating decision trees. In *European Conference on Machine Learning*. Springer, Berlin, Heidelberg.

Liaw, A., and Wiener, M. (2002). Classification and Regression by randomForest. *R News* Vol. 2(3), pp. 18–22. URL `https://cran.r-project.org/web/packages/randomForest/index.html`.

Marshall, R. J. (1999). Classification to ordinal categories using a search partition methodology with an application in diabetes screening. *Statistics in Medicine*, Vol.18, pp. 2723–2735.

Masters, G. N. (1982). A Rasch model for partial credit scoring. *Psychometrika*, Vol. 47, pp. 149–174.

McDonald, R., Hand, D., and Eckley, I. (2003). An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *MSC2003: Multiple Classifier Systems*, pp. 35–44.

Mease, D., and Wyner, A. (2008). Evidence Contrary to the Statistical View of Boosting. *Journal of Machine Learning Research*, Vol. 9, pp. 131–156.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2019). e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.7-1, URL `https://CRAN.R-project.org/package=e1071`.

Meyer, M. C. (2013). Semi-parametric additive constrained regression. *Journal of Nonparametric Statistics*, Vol. 25(3), pp. 715–730.

Pya, N., and Wood, S. N. (2014). Shape constrained additive models. *Statistics and Computing*, Vol. 25(3), pp. 543–559.

R Core Team (2019). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

Robertson, T., Wright, F. T. and Dykstra, R. (1988). *Order Restricted Statistical Inference*. John Wiley and Sons. New York.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, Vol. 5(2), pp. 197–227.

Sobel, M. E., Becker, M. P., and Minick, S. M. (1998). Origins, destinations, and association in occupational mobility. *American Journal of Sociology*, Vol. 104(3), pp. 687–721.

Therneau, T., and Atkinson, B. (2019). rpart: Recursive Partitioning and Regression Trees. R package version 4.1-15, URL `https://CRAN.R-project.org/package=rpart`.

Jarek Tuszynski (2019). caTools: Tools: moving window statistics, GIF, Base64, ROC, AUC, etc.. R package version 1.17.1.2, URL `https://CRAN.R-project.org/package=caTools`.

Turner, R. (2019). Iso: Functions to Perform Isotonic Regression. R package version 0.0-18, URL `https://CRAN.R-project.org/package=Iso`.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S. Fourth Edition*. Springer, New York.

A Appendix

A.1 Theoretical justification for algorithms under the adjacent categories model

Let $\mathbf{x} \in \mathbb{R}^d, y \in \{1,\dots,K\}, y_k^* = I_{[y=k]}, k = 1,\dots,K$ and assume the adjacent probabilities model (2). Denote further $F_1(\mathbf{x}) = 0$, so the a posteriori probabilities are:

$$p_k(\mathbf{x}) = \frac{\exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}{\sum_{k=1}^{K} \exp(\sum_{j=1}^{k} F_j(\mathbf{x}))}, k = 1,\dots,K.$$

Now, the expected log-likelihood is:

$$El(F_2,\dots,F_K) = E\left[\sum_{k=2}^{K} y_k^* \left(\sum_{j=2}^{k} F_j(\mathbf{x})\right) - \log\left(1 + \sum_{k=2}^{K} \exp\left(\sum_{j=2}^{k} F_j(\mathbf{x})\right)\right)\right].$$

Conditioning on $\mathbf{x}$, the score vector $\mathbf{S(x)} = (s_k(\mathbf{x}))$ for the population Newton algorithm is:

$$s_k(\mathbf{x}) = \frac{\partial El(F_2(\mathbf{x}),\dots,F_K(\mathbf{x}))}{\partial F_k(\mathbf{x})} = E\left(\sum_{j=k}^{K}(y_j^* - p_j(\mathbf{x}))\middle|\mathbf{x}\right), k = 2,\dots,K.$$

The Hessian is a $(K-1)\times(K-1)$ matrix, $\mathbf{H}(\mathbf{x}) = (H_{km}(\mathbf{x}))$, $2 \le k, m \le K$, where each element $H_{km}(\mathbf{x})$ is:

$$H_{km}(\mathbf{x}) = \frac{\partial^2 El(F_2(\mathbf{x}),\dots,F_K(\mathbf{x}))}{\partial F_k(\mathbf{x})\partial F_m(\mathbf{x})} = \begin{cases} -(\sum_{j=m}^{K} p_j(\mathbf{x}))(1 - \sum_{j=k}^{K} p_j(\mathbf{x})), & m \ge k \\ -(\sum_{j=k}^{K} p_j(\mathbf{x}))(1 - \sum_{j=m}^{K} p_j(\mathbf{x})), & m < k \end{cases}$$

If $\mathbf{W}(\mathbf{x}) = -diag(\mathbf{H}(\mathbf{x}))$, a quasi-Newton update for the ASILB algorithm is:

$$\begin{bmatrix} F_2(\mathbf{x}) \\ \vdots \\ F_K(\mathbf{x}) \end{bmatrix} \leftarrow \begin{bmatrix} F_2(\mathbf{x}) \\ \vdots \\ F_K(\mathbf{x}) \end{bmatrix} + E_W\left(\mathbf{W}^{-1}(\mathbf{x})\mathbf{s}(\mathbf{x})\middle|\mathbf{x}\right).$$

The full Newton update, which is implemented in the AMILB algorithm, is:

$$\begin{bmatrix} F_2(\mathbf{x}) \\ \vdots \\ F_K(\mathbf{x}) \end{bmatrix} \leftarrow E_H\left(\begin{bmatrix} F_2(\mathbf{x}) \\ \vdots \\ F_K(\mathbf{x}) \end{bmatrix} - \mathbf{H}^{-1}(\mathbf{x})\mathbf{s}(\mathbf{x})\middle|\mathbf{x}\right)$$

A.2 Theoretical justification for algorithms under the cumulative probabilities model

In this case we have to update the function $F(\mathbf{x})$ and the parameters $\alpha_k, k = 2,\dots,K$. We will perform a "two step" update, first on $F(\mathbf{x})$ and then on the α parameters.

Let us denote $\gamma_k(\mathbf{x}) = \sum_{j=k}^{K} p_j(\mathbf{x}), k = 1,\dots,K$, and assume the cumulative probabilities model (3). For this model

$$\gamma_k(\mathbf{x}) = \frac{\exp(\alpha_k + F(\mathbf{x}))}{1 + \exp(\alpha_k + F(\mathbf{x}))}, k = 2,\dots,K,$$

with $\gamma_1(\mathbf{x}) = 1$ and $\gamma_{K+1}(\mathbf{x}) = 0$.

First, we perform the $F(\mathbf{x})$ update. Here, as in the previous model, we consider a single observation as this step is used for updating the weights and the values to be adjusted. The expected log-likelihood is:

$$El(F) = E\left(\sum_{k=1}^{K} y_k^* \log(\gamma_k(\mathbf{x}) - \gamma_{k+1}(\mathbf{x}))\right).$$

Conditioning on $\mathbf{x}$, the first and second derivatives for the population Newton algorithm are:

$$\frac{\partial El(F(\mathbf{x}))}{\partial F(\mathbf{x})} = E\left(\sum_{k=1}^{K} y_k^*[1 - \gamma_k(\mathbf{x}) - \gamma_{k+1}(\mathbf{x})]\middle|\mathbf{x}\right),$$

and

$$w(\mathbf{x}) = -\frac{\partial^2 El(F(\mathbf{x}))}{\partial F(\mathbf{x})^2} = E\left(\sum_{k=1}^{K} y_k^*[\gamma_k(\mathbf{x})(1 - \gamma_k(\mathbf{x})) + \gamma_{k+1}(\mathbf{x})(1 - \gamma_{k+1}(\mathbf{x}))]\middle|\mathbf{x}\right),$$

so that the Newton update for $F(\mathbf{x})$ is

$$F(\mathbf{x}) \leftarrow E_H\left(F(\mathbf{x}) + \frac{1}{w(\mathbf{x})}\frac{\partial El(F(\mathbf{x}))}{\partial F(\mathbf{x})}\middle|\mathbf{x}\right).$$

As for the parameters $\alpha_k, k = 2, \ldots, K$, let us denote $\boldsymbol{\alpha} = (\alpha_2, \ldots, \alpha_K)'$. Now, we use all the $\mathbf{x}_i$ observations as in this case we are going to perform a Newton step for updating the α's which do not depend on $\mathbf{x}$. Then, the log-likelihood is:

$$l(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \sum_{k=1}^{K} y^*_{k,i} \log(\gamma_k(\mathbf{x}_i) - \gamma_{k+1}(\mathbf{x}_i)).$$

The score for the Newton algorithm $\mathbf{S} = (s_2, \ldots, s_K)'$ is:

$$s_k = \frac{\partial l(\boldsymbol{\alpha})}{\partial \alpha_k} = \sum_{i=1}^{n} \left(\frac{y^*_{k,i}}{p_k(\mathbf{x}_i)} - \frac{y^*_{k-1,i}}{p_{k-1}(\mathbf{x}_i)} \right) \gamma_k(\mathbf{x}_i)(1 - \gamma_k(\mathbf{x}_i)), k = 2, \ldots, K. \tag{4}$$

And the Hessian is a tri-diagonal symmetric $(K-1) \times (K-1)$ matrix $\mathbf{H} = (H_{km})$ with $H_{km} = \frac{\partial^2 l(\boldsymbol{\alpha})}{\partial \alpha_k \partial \alpha_m}$, for $2 \leq k, m \leq K$, such that

$$H_{kk} = -\sum_{i=1}^{n} \gamma_k(\mathbf{x}_i)(1 - \gamma_k(\mathbf{x}_i)) \left[\frac{y^*_k}{p^2_k(\mathbf{x}_i)} (p^2_k(\mathbf{x}_i) + \gamma_{k+1}(\mathbf{x}_i)(1 - \gamma_{k+1}(\mathbf{x}_i))) \right.$$
$$\left. + \frac{y^*_{k-1}}{p^2_{k-1}(\mathbf{x}_i)} (p^2_{k-1}(\mathbf{x}_i) + \gamma_{k-1}(\mathbf{x}_i)(1 - \gamma_{k-1}(\mathbf{x}_i))) \right] \tag{5}$$

$$H_{k,k-1} = H_{k-1,k} = \sum_{i=1}^{n} \frac{y^*_{k-1,i}}{p^2_{k-1}(\mathbf{x}_i))} \gamma_k(\mathbf{x}_i)(1 - \gamma_k(\mathbf{x}_i))\gamma_{k-1}(\mathbf{x}_i)(1 - \gamma_{k-1}(\mathbf{x}_i)) \tag{6}$$

$$H_{km} = H_{mk} = 0 \text{ otherwise.} \tag{7}$$

In these conditions the Newton update is $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - \mathbf{H}^{-1}\mathbf{S}$.

A.3 Full numerical results for the simulations performed

This subsection contains the Tables showing the full numerical mean results for TMP, MAE and AUC for the two sets of simulations. In Tables 7, 8 and 9 appear the results for the first set of simulations performed under model 2 for the different F functions appearing in Table 3. Tables 10, 11 and 12 contain the mean results for the simulations performed under the uniform order-restricted predictors scheme. In all cases the best results appear in bold. Notice that there are no results for CSILB and CMILB when $K = 2$ as in that case those algorithms coincide with ASILB and AMILB. For this case the TMP and MAE values also coincide. They are given in both tables for completeness.

F	K	d	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	(A)SILB	(A)MILB	CSILB	CMILB
1	2	5	.4087	.3975	.4176	.4162	.4086	.4454	.4376	**.3835**	.3925		
1	2	10	.4547	.4354	.4660	.4850	.4532	.4714	.4719	**.4277**	.4334		
2	2	5	.3758	.3654	.4127	.4279	.3822	.4485	.4233	.3679	**.3675**		
2	2	10	.4262	.4358	.4754	.4621	.4288	.4677	.4739	.4188	**.3990**		
1	3	5	.4933	.4823	.5259	.5416	.4940	.5433	.5573	.4843	**.4808**	.4872	.4936
1	3	10	.5225	.5162	.5343	.5372	.5226	.5631	.5647	.5059	.5007	**.5001**	.5013
2	3	5	.4742	.4681	.5146	.4975	.4745	.5184	.5789	.4541	**.4492**	.4703	.4628
2	3	10	.5112	.5039	.5339	.5228	.5165	.5433	.6007	.4867	**.4774**	.4855	.4841
1	5	5	.6469	.6420	.6682	.6635	.6497	.6850	.6994	.6301	**.6255**	.6289	.6325
1	5	10	.6612	.6506	.6750	.6675	.6588	.6902	.7038	.6336	.6388	**.6293**	.6306
2	5	5	.6296	.6161	.6690	.6571	.6356	.6709	.7208	.6012	.5978	**.5848**	.6080
2	5	10	.6465	.6378	.6693	.6708	.6489	.6864	.7312	.6136	**.6098**	.6113	.6168

Table 7: Mean TMP for the first simulation scheme for different classification rules, number of groups K, predictors d and functions F.

F	K	d	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	(A)SILB	(A)MILB	CSILB	CMILB
1	2	5	.4087	.3975	.4176	.4162	.4086	.4454	.4376	**.3835**	.3925		
1	2	10	.4547	.4354	.4660	.4850	.4532	.4714	.4719	**.4277**	.4334		
2	2	5	.3758	.3654	.4127	.4279	.3822	.4485	.4233	.3679	**.3675**		
2	2	10	.4262	.4358	.4754	.4621	.4288	.4677	.4739	.4188	**.3990**		
1	3	5	.5931	.5732	.6467	.6713	.5949	.6465	.7025	**.5627**	.5727	.5808	.5868
1	3	10	.6437	.6342	.6706	.6564	.6421	.7063	.7279	**.5928**	.6041	.6096	.5977
2	3	5	.5540	.5423	.6193	.5889	.5547	.6221	.7393	**.5191**	.5371	.5337	.5384
2	3	10	.6065	.5916	.6457	.6251	.6133	.6670	.7646	**.5596**	.5761	.5620	.5768
1	5	5	.9853	.9505	1.0448	1.0318	.9872	1.0930	1.1610	**.8821**	.9166	.9012	.8956
1	5	10	1.0224	.9771	1.0614	1.0449	1.0169	1.1309	1.1710	**.9075**	.9451	.9162	.9154
2	5	5	.9422	.9103	1.0525	1.0250	.9523	1.0673	1.2901	.8318	.9041	**.8119**	.9046
2	5	10	.9758	.9482	1.0653	1.0460	.9778	1.0921	1.3094	.8573	.9268	**.8559**	.9084

Table 8: Mean MAE for the first simulation scheme for different classification rules, number of groups K, predictors d and functions F.

F	K	d	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	(A)SILB	(A)MILB	CSILB	CMILB
1	2	5	.6374	.6596	.6263	.5675	.6371	.5700	.5683	**.6754**	.6628		
1	2	10	.5650	.5963	.5432	.4800	.5654	.5370	.5273	**.6034**	.6020		
2	2	5	.6939	.6861	.6254	.6041	.6660	.5925	.5705	.6961	**.7063**		
2	2	10	.5809	.5824	.5442	.4767	.5800	.5420	.5244	.6168	**.6290**		
1	3	5	.6969	**.7049**	.6602	.6504	.6969	.6433	.6189	.7016	.6993	.6876	.6802
1	3	10	.6721	.6866	.6439	.6419	.6714	.6139	.6095	**.6877**	.6824	.6753	.6691
2	3	5	.7152	.7240	.6758	.6717	.7113	.6533	.5982	.7320	**.7344**	.7206	.7234
2	3	10	.6875	.6968	.6603	.6706	.6855	.6285	.5744	**.7060**	.7044	.7055	.7025
1	5	5	.7089	.7175	.6731	.6893	.7091	.6244	.6263	**.7226**	.7198	.7028	.7014
1	5	10	.6981	.7072	.6681	.6754	.6967	.6205	.6127	**.7127**	.7078	.6929	.6939
2	5	5	.7137	.7270	.6719	.6782	.7091	.6350	.5805	**.7383**	.7305	.7378	.7115
2	5	10	.7070	.7159	.6681	.6883	.7053	.6306	.5812	**.7334**	.7197	.7180	.7047

Table 9: Mean AUC for the first simulation scheme for different classification rules, number of groups K, predictors d and functions F.

K	n	d	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	(A)SILB	(A)MILB	CSILB	CMILB
2	10K	5	.4482	.4066	.4693	.4505	.4421	.4531	.4711	.3741	**.3584**		
2	20K	5	.4139	.4029	.3936	.4091	.4335	.4411	.4571	.3552	**.3271**		
2	10K	10	.4113	.3826	.4025	.4349	.4123	.4224	.4543	**.3232**	.3297		
2	20K	10	.3596	.3573	.3594	.3796	.3636	.3893	.4519	.2998	**.2883**		
3	10K	5	.5391	.5131	.5495	.5585	.5424	.5657	.5899	.4753	.4817	**.4657**	.4806
3	20K	5	.5030	.4923	.5090	.4988	.5362	.5426	.6177	.4597	**.4138**	.4279	.4171
3	10K	10	.5288	.5051	.5137	.5396	.5271	.5255	.5701	**.4296**	.4415	.4403	.4439
3	20K	10	.4717	.4576	.4731	.4807	.4738	.4894	.5580	.3975	.3985	**.3675**	.3984
5	10K	5	.6307	.6015	.6335	.6404	.6358	.6585	.6717	.5969	.5854	**.5580**	.5780
5	20K	5	.5705	.5626	.5922	.5753	.6084	.6350	.6879	.5357	.5228	**.4879**	.5157
5	10K	10	.6088	.5911	.5944	.6234	.6070	.6326	.6258	.5318	**.5304**	.5342	.5418
5	20K	10	.5516	.5406	.5598	.5552	.5565	.5947	.6425	.4770	.4782	**.4462**	.4659

Table 10: Mean TMP for different classification rules, number of groups K, predictors d and training sample sizes n, for the second set of simulations.

K	n	d	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	(A)SILB	(A)MILB	CSILB	CMILB
2	10K	5	.4482	.4066	.4693	.4505	.4421	.4531	.4711	.3741	**.3584**		
2	20K	5	.4139	.4029	.3936	.4091	.4335	.4411	.4571	.3552	**.3271**		
2	10K	10	.4113	.3826	.4025	.4349	.4123	.4224	.4543	**.3232**	.3297		
2	20K	10	.3596	.3573	.3594	.3796	.3636	.3893	.4519	.2998	**.2883**		
3	10K	5	.6573	.6171	.6673	.6862	.6657	.6868	.7385	.5505	.5792	**.5483**	.5741
3	20K	5	.5943	.5736	.5937	.5751	.6418	.6464	.7881	.5101	.4661	.4594	**.4499**
3	10K	10	.6438	.6131	.6118	.6507	.6393	.6317	.7119	**.4821**	.5094	.4963	.4894
3	20K	10	.5303	.5117	.5281	.5453	.5328	.5629	.6724	.4292	.4367	**.3865**	.4353
5	10K	5	.8658	.8015	.9020	.9138	.8852	.9895	1.0752	.7788	.8043	**.7054**	.7445
5	20K	5	.7264	.7045	.7699	.7290	.8247	.9008	1.0910	.6478	.6515	**.5564**	.6177
5	10K	10	.8542	.8156	.8293	.8743	.8561	.9296	.9331	.6843	.7000	**.6613**	.6838
5	20K	10	.6747	.6568	.6889	.6824	.6896	.8105	.9013	.5471	.5679	**.4959**	.5307

Table 11: Mean MAE for different classification rules, number of groups K, predictors d and training sample sizes n, for the second set of simulations.

K	n	d	LDA	RLDA	RF	SVM	LOGIT	LGB	MONOXGB	(A)SILB	(A)MILB	CSILB	CMILB
2	10K	5	.5861	.6412	.5535	.4807	.5894	.5655	.5216	.6798	**.6914**		
2	20K	5	.6195	.6378	.6446	.4593	.5732	.5888	.5352	.7175	**.7492**		
2	10K	10	.6290	.6768	.6426	.5546	.6286	.6083	.5677	**.7517**	.7485		
2	20K	10	.6977	.7043	.6910	.6220	.6925	.6500	.5650	.7761	**.7855**		
3	10K	5	.6511	.6806	.6369	.5408	.6506	.6082	.5752	.7126	.7111	**.7344**	.7210
3	20K	5	.6819	.6951	.6812	.6204	.6434	.6344	.5497	.7370	.7740	.7651	**.7768**
3	10K	10	.6682	.6858	.6732	.6158	.6697	.6452	.6078	.7558	.7493	**.7608**	.7582
3	20K	10	.7273	.7372	.7212	.7025	.7201	.6783	.6170	.7907	.7948	**.8213**	.7953
5	10K	5	.7257	.7429	.7107	.7216	.7146	.6370	.6278	.7575	.7544	**.7585**	.7453
5	20K	5	.7717	.7834	.7452	.7616	.7041	.6603	.6035	.8054	.8128	**.8293**	.8088
5	10K	10	.7437	.7544	.7434	.7154	.7435	.6569	.6833	.8033	**.8078**	.7783	.7778
5	20K	10	.7991	.8053	.7814	.7846	.7913	.6854	.6721	.8452	.8419	**.8495**	.8377

Table 12: Mean AUC for different classification rules, number of groups K, predictors d and training sample sizes n, for the second set of simulations.