On Nonparametric Ordinal Classification with Monotonicity Constraints

Wojciech Kotłowski and Roman Słowiński, *Senior Member, IEEE*

Abstract—We consider the problem of ordinal classification with monotonicity constraints. It differs from usual classification by handling background knowledge about ordered classes, ordered domains of attributes, and about a monotonic relationship between an evaluation of an object on the attributes and its class assignment. In other words, the class label (output variable) should not decrease when attribute values (input variables) increase. Although this problem is of great practical importance, it has received relatively low attention in machine learning. Among existing approaches to learning with monotonicity constraints, the most general is the nonparametric approach, where no other assumption is made apart from the monotonicity constraints assumption. The main contribution of this paper is the analysis of the nonparametric approach from statistical point of view. To this end, we first provide a statistical framework for classification with monotonicity constraints. Then, we focus on learning in the nonparametric setting, and we consider two approaches: the "plug-in" method (classification by estimating first the class conditional distribution) and the direct method (classification by minimization of the empirical risk). We show that these two methods are very closely related. We also perform a thorough theoretical analysis of their statistical and computational properties, confirmed in a computational experiment.

Index Terms—Machine learning, monotonicity constraints, ordinal classification, ordinal regression, preference learning, nonparametric methods, isotonic regression, isotonic classification, monotone functions

1 Introduction

Using background knowledge is of fundamental importance in the learning process. A common type of such knowledge, concerning data describing decision problems, is a monotone relationship between input and output variables. Consider evaluations of objects on ordinal attributes, and decision about their class assignment to some ordered classes. It is often rational to assume that the better the evaluation of an object on considered attributes (input variables), the better its class assignment (output variable). For instance, "the higher the debt ratio of a company, the higher its level of bankruptcy risk"; "the better education and experience level of a candidate for a job, the higher his/her position in the selection process." This assumption, commonly accepted in decision analysis, is called *dominance principle*. It is just a kind of background knowledge referring to so-called *ordinal classification with monotonicity constraints*. Such background knowledge is particularly important in the decision problems involving preferences, like social choice, multiple criteria decision making, or decision under risk and uncertainty. In these decision problems, attributes are called *criteria* [1], [2]; learning from such data is called *preference learning* [3]. However, dominance principle is not limited to preference learning, as other problems may need to obey this principle, for instance, discovering laws in physics, like "the greater the mass and the smaller the distance, the higher the gravity."

Indeed, monotonicity constraints are frequently encountered in data analysis. For example, in the customer satisfaction analysis [4], the overall evaluation of a product by a customer should increase with increasing evaluations of the product on a set of attributes. In the house pricing problem [5], the selling price of the house should increase with, for example, lot size, number of rooms, and decrease with, for example, crime rate or pollution concentration in the area. Monotonicity also occurs in such domains as bankruptcy risk prediction [6], option pricing [7], medical diagnosis [8], [9], credit approval [10], survey data [11], and many others.

The need of handling background knowledge about ordinal evaluations and monotonicity constraints in the learning process led to the development of new algorithms. The examples of such algorithms are Dominance-based Rough Set Approach (DRSA) [1], [12] along with decision rule induction [13], [14], [15], rule ensembles [16], monotone classification trees [5], [17], [18], [19], monotone networks [20], instance-based methods [11], [21], [22] or isotonic separation [23]. Related problems, where the classes are ordered, but there are no monotonicity relationship between attribute evaluation and class assignment, are considered in machine learning and statistics under the name *ordinal regression* [24], [25].

The approaches mentioned above assume some particular class of monotone classification functions (e.g., sets of rules, neural networks, decision trees, etc.) and pick one function from that class, based on the training data (e.g., by

- W. Kotłowski is with the Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland. E-mail: wkotlowski@cs.put.poznan.pl.
- R. Słowiński is with the Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland, and the Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland. E-mail: roman.slowinski@cs.put.poznan.pl.

Manuscript received 14 June 2012; accepted 14 Sept. 2012; published online 15 Oct. 2012.
Recommended for acceptance by B.C.Ooi.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2012-06-0424.
Digital Object Identifier no. 10.1109/TKDE.2012.204.

minimizing the training error). Interestingly, when the monotonicity constraints are present, it is possible to efficiently work with a very general class of functions, which put no constraints on the function's shape other than the monotonicity constraint [22], [26], [27]. In other words, it is possible to efficiently do inference in the class of *all* monotone functions. We refer to such an approach as *nonparametric approach*. It has the advantage of not making any additional, ill-founded assumptions about the model apart from the only justified background knowledge: the monotonicity constraints. Moreover, even if the learning algorithm deals with restricted class of monotone functions (e.g., decision rules, nearest neighbors), the nonparametric approach can be used as an initial preprocessing procedure for relabeling the data set and making it obey the monotonicity constraints; this serves as the nonparametric error correction, based solely on the monotonicity constraints assumption—such an approach was used, for example, in [16], [22], [23], [26].

Although several methods for learning with monotonicity constraints were proposed, there exists no theoretical framework for their analysis. In this paper, we provide such a framework, based on the statistical approach to the problem. Then, we analyze the computational and statistical properties of the nonparametric approach in this framework. Thus, the main contribution of this paper is twofold. First, we formalize the approach to learning with monotonicity constraints from statistical point of view. We show how such constraints can be handled by making general assumptions about probability distribution generating the data. We also formulate necessary and sufficient conditions imposed on the structure of the loss function, under which the optimal Bayes classifier is monotone.

Second, we analyze nonparametric classification methods. We consider two general approaches to classification: the "plug-in" approach (classification by estimating the class conditional distribution) and the direct approach (classification by minimization of the empirical risk). The plug-in approach is based on the multiple *isotonic regression*. Although isotonic regression was extensively studied in the context of regression problems, hypothesis testing, and probability estimation in the Bernoulli model [28], [29], [30], application of isotonic regression to multiclass ordinal classification is a new result. The direct approach (to which we refer as *isotonic classification*) is based on a linear program. We show how the general K-class problem can be decomposed into a series of binary-class subproblems. We also show how to speed up learning by exploiting some properties of the data, which allow us to remove some of the variables from the problem. We analyze the relationship between plug-in and direct approaches, and show that they coincide for a large class of loss function. We also investigate their asymptotic consistency. Finally, we verify the methods in the extensive computational experiment.

The introduced statistical framework is the original contribution of the paper, following some preliminary results from [16], [26]. The nonparametric methods presented here were initially considered in [22], [26], [27], but the multiple isotonic regression, its relationship to isotonic classification, and the analysis of computational and statistical properties of the two methods are new.

2 Problem Statement

Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ be an object-label pair generated according to some unknown distribution $P(x, y)$, where y is a class label from a finite set of *ordered* labels $\mathcal{Y} = \{1, \dots, K\}$, x is the object's description, and $\mathcal{X}$ is the input space.[1] The goal is to find a *classifier* $h : \mathcal{X} \to \mathcal{Y}$ that accurately predicts value of y using the knowledge about the object expressed by means of x. The accuracy of a single prediction $\hat{y}$ is measured in terms of a *loss function* $L(y, \hat{y})$, which is a penalty for predicting $\hat{y}$ when the actual value is y. The overall accuracy of classifier h is defined as the expected loss (risk) according to the probability distribution $P(x, y)$ of the data:

$$L(h) = \mathbb{E}[L(y, h(x))]. \tag{1}$$

A *Bayes classifier* is a function h^* minimizing the expected loss, $h^* = \arg\min_h L(h)$. The minimum $L^* = L(h^*)$ is called the *Bayes risk*. It follows that [31]

$$h^*(x) = \arg\min_{k \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} L(y, k) P(y|x). \tag{2}$$

In other words, $h^*(x)$ minimizes the expected loss at x, where the expectation is with respect to conditional class distribution $P(y|x)$. Using Bayes classifier is the best one can do. Unfortunately, $P(x, y)$ is usually unknown, thus h^* is also unknown and the goal is to learn a good approximation of h^* using an i.i.d. sample of n training examples $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, called a *training set*. There are two general approaches to this problem. The first approach, referred to as *plug-in method*, is based on estimating the conditional distribution $P(y|x)$ by a set of K functions $\hat{p}_1(x), \dots, \hat{p}_K(x)$, such that $\hat{p}_k(x)$ is the estimator of $P(y = k|x)$. Then, the classifier $\hat{h}$ is obtained by plugging the probability estimators into the definition of the Bayes classifier (2), i.e.,

$$\hat{h}(x) = \arg\min_{k \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} L(y, k) \hat{p}_y(x). \tag{3}$$

The second approach is to directly learn a classifier by minimizing the *empirical risk*:

$$L_D(h) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(x_i)),$$

within some family of classification functions $\mathcal{H}$, i.e.,

$$\hat{h} = \arg\min_{h \in \mathcal{H}} L_D(h). \tag{4}$$

This approach will be referred to as *direct method*.

It is reasonable to assume that $L(k, k) = 0$ and $L(y, k) > 0$ if $y \neq k$ for $y, k = 1, \dots, K$. The loss should also be consistent with the order between class labels, in the sense that the loss should not decrease, as the predicted value moves away from the true value. Hence, following [25], we assume the loss matrix is *V-shaped*: for $k \leq y$ it holds $L(y, k-1) \geq L(y, k)$, while for $k \geq y$ it holds $L(y, k) \leq L(y, k+1)$.

[1] Throughout the text, we denote by lower case letters both random variables and their actual values, which, hopefully, should not lead to a confusion.

Notice that until now, the proposed framework bears close resemblance to the problem of ordinal regression. What makes it specific is the presence of monotonicity constraints. We assume that objects are described in terms of m input variables with *ordered* domains. Therefore, without loss of generality, $\mathcal{X} \subseteq \mathbb{R}^m$ and object x is an m-dimensional vector $x = (x^1, \dots, x^m)$. We exploit the order properties of $\mathcal{X}$ by using the available knowledge given in terms of *dominance relation*. We say that x *dominates* x', denoted by $x \succeq x'$, if each coordinate (input variable) of x is not smaller than the respective coordinate of x', $x^s \geq x'^s, s = 1, \dots, m$. Notice that dominance relation is a partial order on $\mathcal{X}$. The monotonicity constraints require that if $x \succeq x'$, then x should be assigned a class label greater or equal to x'.

In the nonparametric procedures considered here, the dominance relation is the only information about $\mathcal{X}$ which is taken into account: no other properties of x (such as description in terms of the input variables) are used. Therefore, one can even consider a more general setting, in which the dominance relation is a primal concept given in the problem, while the internal structure of objects is not specified. In other words, we are only given a partially ordered set $(\mathcal{X}, \succeq)$ and i.i.d. sample D of n elements from $\mathcal{X}$ (along with their class labels). All the methods considered in this paper are directly applicable to such a general case.

In the paper, we denote objects from the training set D by x_i or x_j, where $i, j = 1, \dots, n$; if we consider any object from the whole space $\mathcal{X}$, we denote it by x or x'. Indices i and j always run in the set $\{1, \dots, n\}$, while indices k and y in the set $\{1, \dots, K\}$. We use bold symbols only for n-dimensional vectors, for example, $\mathbf{y} = (y_1, \dots, y_n)$.

Let us call a function $h : \mathcal{X} \to \mathcal{Y}$ *monotone* if for any $x, x' \in \mathcal{X}$ it holds $x \succeq x' \to h(x) \geq h(x')$. Let us call a vector $\mathbf{v} = (v_1, \dots, v_n) \in \mathcal{Y}^n$ *monotone*, if for each $i, j = 1, \dots, n$ it holds: $x_i \succeq x_j \to v_i \geq v_j$. In other words, monotone vectors are defined on the training set D, while monotone functions on the whole space $\mathcal{X}$.

3 Monotonicity Constraints

In this section, we formalize the concept of monotonicity constraints from statistical point of view, by imposing constraints on the probability distribution generating the data. Then, we formulate necessary and sufficient conditions imposed on the structure of the loss function, under which the optimal Bayes classifier is monotone.

3.1 Stochastic Dominance

The monotonicity constraints require that if $x \succeq x'$ then x should be assigned a class not lower than that of x'. In practice, these constraints are not always satisfied, which suggests that the order relation $\succeq$ does not impose "hard" constraints, so that the constraints should rather be defined in a probabilistic setting. Consider two points $x, x' \in \mathcal{X}$, such that $x \succeq x'$. The core of the concept of monotonicity constraints lies in the following assumption: fix $k \in \{1, \dots, K\}$; then, the probability that x will get class label equal to at least k should not be smaller than the probability that x' will get class label equal to at least k:

$$P(y \geq k|x) \geq P(y \geq k|x'). \tag{5}$$

In other words, the probability of the event $\{y \geq k\}$ is a monotone function for every k. Since $P(y \geq k) = 1 - P(y < k)$, (5) is equivalent to $P(y \leq k|x) \leq P(y \leq k|x')$ for each k. Notice that (5) is a relation between two probability distributions, conditioned at x and x', respectively. This relation is known as *(first order) stochastic dominance* [32]. We overload the symbol $\succeq$ and use it also to denote the stochastic dominance relation between distributions. Therefore, (5) can be concisely written as

$$x \succeq x' \Longrightarrow P(y|x) \succeq P(y|x'), \tag{6}$$

where $P(y|x)$ and $P(y|x')$ denote the class conditional distributions at x and x', respectively. We will call a probability distribution *monotonically constrained* if it satisfies (6). Notice that in [11], [33] stochastic dominance was also used, but to define the properties of the estimator, not the properties of the probability distribution.

3.2 Monotone Bayes Classifier

In the classification problem, we aim at finding the classifier which is as close as possible to the Bayes classifier. In other words, the Bayes classifier is our "target function" that we try to approximate. Thus, not surprisingly, we require that in the classification with monotonicity constraints the Bayes classifier must be monotone.[2] Remark, however, that although the probability distribution is monotonically constrained, the monotonicity of the Bayes classifier does not always hold. For instance, the Bayes classifier for 0-1 loss (the mode of the distribution) is not monotone under stochastic dominance assumption. Indeed, consider the following 3-class counterexample: let $x \succeq x'$ and let us define the conditional distributions at x and x' as $P(y|x) = (0.1, 0.5, 0.4)$ and $P(y|x') = (0.3, 0.3, 0.4)$. Although $P(y|x) \succeq P(y|x')$, the mode of $P(y|x')$ (output of the Bayes classifier) is 3, while the mode of $P(y|x)$ is 2.

It appears that some specific constraints must be imposed on the loss function to maintain the monotonicity of the Bayes classifier. They are given in the following theorem:

Theorem 1. *Let the loss function $L(y, k)$ be V-shaped. The Bayes classifier is monotone for every monotonically constrained distribution $P(x, y)$ if and only if the loss function satisfies for every $y, k \in \{1, \dots, K-1\}$, the following condition:*

$$L(y, k+1) - L(y, k) \geq L(y+1, k+1) - L(y+1, k). \tag{7}$$

The proof can be found in the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2012.204. It follows from Theorem 1 that condition (7) is necessary and sufficient for monotonicity of the Bayes classifier. The latter property is desired in the classification with monotonicity constraints, otherwise there would be no point in minimizing the risk within the class of monotone functions. Therefore, we will call the loss function satisfying (7) a *monotone loss function*.

[2] The Bayes classifier may not be unique, because it is defined only up to a zero measure set. To avoid this problem, we assume that for *every* $x \in \mathcal{X}$, the Bayes classifier returns the class label k with the smallest conditional risk $\mathbb{E}[L(y, k)|x]$; in case of ties on the conditional risk, the lowest label is always chosen.

3.3 Monotone Bayes Classifier and Convex Loss

We analyze condition (7) in case of a popular subclass of the loss functions. Namely, the loss function is very often expressed in the form $L(y,k) = c(y-k)$, with $c(0) = 0$ and $c(k) > 0$ for $k \neq 0$. The loss functions of such type are, for instance, 0-1 loss ($c(k) = 1_{k \neq 0}$),[3] absolute error loss ($c(k) = |k|$), or squared error loss ($c(k) = k^2$). Moreover, every binary loss has this form, because it is determined by two values $L(1,2) = c(-1)$ and $L(2,1) = c(1)$.

Let $\bar{\mathcal{Y}} = \{-(K-1), \dots, -1, 0, 1, \dots, K-1\}$. Let us call a function $c : \bar{\mathcal{Y}} \rightarrow \mathbb{R}$ *convex* if for every k, such that $-(K-1) < k < K-1$, we have

$$c(k) \leq \frac{c(k-1) + c(k+1)}{2}. \tag{8}$$

This is a natural definition of convexity for a function defined over integer-valued domain. The convexity turns out to be a crucial property determining the monotonicity of the Bayes classifier:

Theorem 2. *Let $L(y,k) = c(y-k)$ be the V-shaped loss function. Then, the Bayes classifier $h^*(x)$ is monotone if and only if $c(k)$ is convex.*

Proof. Condition (7) can now be expressed as

$$c(y-k-1) - c(y-k) \geq c(y-k) - c(y-k+1),$$

which is equivalent to condition (8). □

Corollary 1. *Let $L(y,k) = |y-k|^p$, for $p \geq 0$, be the loss function, and let $K \geq 3$. Then, the Bayes classifier is monotone if and only if $p \geq 1$.*

Proof. Function $f(z) = |z|^p$ is convex for $p \geq 1$ and strictly concave for $p < 1$. Therefore, condition (8) holds if and only if $p \geq 1$. □

We assumed $K \geq 3$, since when $K = 2$, every loss function is convex (and thus monotone). Corollary 1 implies that 0-1 loss ($p \rightarrow 0$) does not result in the monotone Bayes classifier, while absolute error loss ($p = 1$) and squared-error loss ($p = 2$) do ensure monotonicity. This suggests that 0-1 loss is not a proper loss function for ordinal classification for $K \geq 3$, if one assumes stochastic dominance between the conditional distributions.

3.4 Linear Loss Function

In this paper, we focus on a specific class of the loss functions, called *linear loss functions* [34], defined as

$$L(y,k) = \begin{cases} \alpha(k-y) & \text{if } k > y \\ (1-\alpha)(y-k) & \text{if } k \leq y, \end{cases} \tag{9}$$

where $0 < \alpha < 1$. From Corollary 1, we immediately have that the linear loss is a monotone loss function. For $\alpha = \frac{1}{2}$, we have an absolute error loss $l_{yk} = |k-y|$ (up to the proportional constant). The purpose of introducing (9) is to model asymmetric cost of misclassification: for $\alpha > \frac{1}{2}$, predicting higher class than the actual class y is more penalized than predicting the lower class; for $\alpha < \frac{1}{2}$, we have the opposite situation. Such a loss function can be useful, for example, in medicine: consider classifying patient into classes according to her/his health condition: "good," "moderate," "bad," "very bad." Then, predicting the patient's condition to be better than it really is is usually more dangerous than predicting the condition to be worse than it is. It is known [34], that such a loss function is minimized by $(1-\alpha)$-quantile of the conditional distribution,[4] i.e., by such $y_{1-\alpha}$ that $P(y \leq y_{1-\alpha}) \geq 1-\alpha$ and $P(y \geq y_{1-\alpha}) \geq \alpha$. For $\alpha = \frac{1}{2}$, we obtain the median of the distribution. Minimization of the linear loss has a very important property of being independent of the particular encoding of the class labels. Indeed, the quantile of a distribution is invariant under any strictly monotonic (order-preserving) transformation of the domain of the distribution.

4 The Plug-in Approach

For the rest of the paper, we will consider nonparametric methods of classification. They are called "nonparametric," because they exploit the class of *all* monotone functions. The nonparametric approach has the advantage of not making any additional assumptions about the model apart from the monotonicity constraints.

In this section, we consider the plug-in approach to nonparametric classification. Since it follows from (3) that the classifier is determined by the estimators $\hat{p}_1(x), \dots, \hat{p}_K(x)$ of the conditional class distribution, we will need to construct a good method for estimating $P(y|x)$. Knowing $P(y|x)$ has two main advantages: first, the conditional distribution allows determination of the optimal prediction for *any* loss function, according to (2); second, the conditional distribution measures the confidence of prediction. Our estimation method is based on isotonic regression. This approach has been first proposed by Dembczyński et al. [35], [36], and, independently, by Feelders [37]. Here, we analyze it for the first time in the statistical framework for classification with monotonicity constraints.

Note that as a direct corollary from Theorem 1 we have:

Corollary 2. *Let $\hat{h} : \mathcal{X} \rightarrow \mathcal{Y}$ be a plug-in classifier defined by (3) and let $L(y,k)$ be a monotone loss functions. Then, $\hat{h}$ is monotone, provided that for every $k = 2, \dots, K$, function:*

$$\hat{P}_k(x) := \sum_{k'=k}^{K} \hat{p}_{k'}(x) \tag{10}$$

is monotone, where $\hat{p}_1(x), \dots, \hat{p}_K(x)$ are estimators of the conditional class distribution.

Proof. We only need to apply Theorem 1 with distribution $P(y \geq k|x) := \hat{P}_k(x)$, which, according to (6), is monotonically constrained if $\hat{P}_k(x)$ is monotone for every $k = 2, \dots, K$. □

Corollary 2 provides us with a natural constraint (10), which should be satisfied by reasonable estimators of the conditional class distribution.

4.1 Binary-Class Problem and Isotonic Regression

Let us first restrict to the binary case ($K = 2$), and for the sake of clarity, let us use the set of class labels $\mathcal{Y} = \{0, 1\}$.

[3] 1_A is an indicator function, equal to 1 if A is true, and 0 otherwise.

[4] We remind that, in general, p-quantile of probability distribution $P(x)$ is defined as a value x_p such that $P(x \leq x_p) \geq p$ and $P(x \geq x_p) \geq 1-p$.

For each x_i, we have a single estimator $\hat{p}_1(x_i)$, because $\hat{p}_0(x_i) = 1 - \hat{p}_1(x_i)$. Let us concisely denote $\hat{p}_1(x_i)$ as $\hat{p}_i$. In the plug-in method for binary class problem, we propose to use a vector conditional density estimators $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_n)$, which is an isotonic regression of the vector of labels $\mathbf{y} = (y_1, \dots, y_n)$.

Let us call a vector $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_n)$ an *isotonic regression* of $\mathbf{y} = (y_1, \dots, y_n)$ if $\hat{\mathbf{p}}$ is the solution of the following problem:

$$\begin{aligned} &\text{minimize: } \sum_{i=1}^{n} (y_i - p_i)^2 \\ &\text{subject to: } x_i \succeq x_j \Longrightarrow p_i \geq p_j \qquad i, j = 1, \dots, n, \end{aligned} \tag{11}$$

so that $\hat{\mathbf{p}}$ minimizes the squared error in the set of all monotone vectors $\mathbf{p} = (p_1, \dots, p_n)$. The constraints in (11) are necessary, as they correspond to monotonicity constraints (10) which in turn guarantee monotonicity of the classifier. Although choosing the squared error loss as a measure of error seem arbitrary, it can be shown that minimizing many other error functions yields to the same solution $\hat{\mathbf{p}}$ [30]. In particular, it can be shown that the isotonic regression is the maximum-likelihood estimator of probabilities given monotonicity constraints (for details, and proof, see [30]).

The isotonic regression is a quadratic optimization problem with linear constraints, and therefore can be solved efficiently by most of general-purpose optimization solvers. Burdakov et al. [38] proposed a heuristic algorithm giving results close to optimum in $O(n^2)$. The exact algorithm works in $O(n^4)$ [39]. However, the size of the problem can usually be significantly reduced beforehand. To show this, we introduce an important property of the isotonic regression, which will be used several times in this paper. A subset $L \subseteq \{x_1, \dots, x_n\}$ is a *lower set* if $x_i \in L$ implies $x_j \preceq x_i \rightarrow x_j \in L$. Similarly, a subset U is an *upper set* if $x_i \in U$ implies $x_j \succeq x_i \rightarrow x_j \in U$. For any vector $\mathbf{f}$, let us denote by $Av(\mathbf{f}, A) = \frac{1}{|A|} \sum_{i \in A} f_i$ the average value of $\mathbf{f}$ on a set A. Then, the following theorem holds:

Theorem 3 [30]. *Let $\hat{p}$ be the isotonic regression of $\hat{y}$. Then,*

$$\hat{p}_i = \min_{L: x_i \in L} \max_{U: x_i \in U} Av(\mathbf{y}, L \cap U),$$

where the minimum and the maximum are taken over all lower and upper sets, respectively.

Using Theorem 3, one can show that the isotonic regression problem can be significantly reduced, because the optimal values of some of the variables are known a priori. Let us call object x_i *consistent* if for every $j = 1, \dots, n$, it holds: $x_i \succeq x_j \rightarrow y_i \geq y_j$ and $x_i \preceq x_j \rightarrow y_i \leq y_j$. We can use consistency property to significantly reduce the size of the problem:

Theorem 4. *Let $\hat{\mathbf{p}}$ be the isotonic regression of* $\mathbf{y}$. *Then, $\hat{p}_i = y_i$ if and only if object x_i is consistent.*

Proof. We consider the case $y_i = 1$ (the case $y_i = 0$ is analogous). If x_i is consistent, then for every x_j, such that $x_j \succeq x_i$, we have $y_j = 1$. Thus, the upper set $U_i = \{x_j : x_j \succeq x_i\}$ includes only objects with $y_j = 1$. But then, for every lower set L, such that $x_i \in L$, $Av(\mathbf{y}, L \cap U_i) = 1$, which by Theorem 3 implies $\hat{p}_i = 1$. Conversely, if x_i is not consistent, then there exists x_j, such that $x_j \succeq x_i$ and $y_j = 0$; but then, for every upper set U, such that $x_i \in U$, we must have $x_j \in U$. Choose any of such sets and a trivial lower set $L_0 = \{x_1, \dots, x_n\}$, and notice that $Av(\mathbf{y}, L_0 \cap U) < 1$, which by Theorem 3 implies $\hat{p}_i < 1$. □

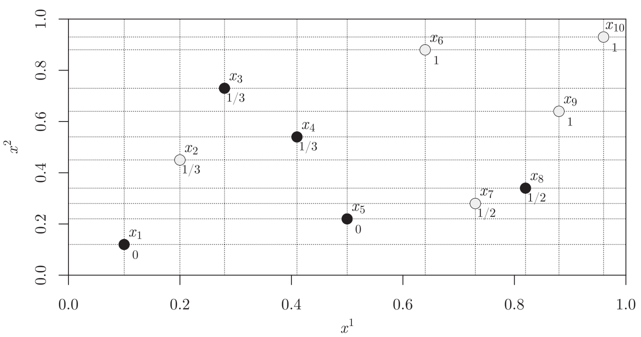

Fig. 1. Binary-class example with two input variables. Objects with $y = 0$ are dark, while with $y = 1 -$ light. The estimate of probability of class 1, $\hat{p}_i$, is shown. Notice that for consistent objects $(x_1, x_5, x_6, x_9, x_{10})$ it holds $y_i = \hat{p}_i$.

Thus, only consistent objects have optimal values equal to 1 or 0. We can set $\hat{p}_i = y_i$ for each consistent object x_i and optimize (11) only for inconsistent objects, which usually gives a large reduction of the problem size (number of variables). A simple example of isotonic regression is shown in Fig. 1.

4.2 Multiclass Problem

In the multiclass case, we propose an estimator based on the multiple isotonic regression. The idea is to decompose the K-class problem into a sequence of binary classification problems and apply isotonic regression to each of the binary problems. Although each of the problems is tackled separately, we prove that the final estimator is a consistent conditional probability distribution.

Let $\mathcal{Y} = \{1, \dots, K\}$, and for a given x_i let us define $K - 1$ dummy values $y_{ik} = 1_{y_i \geq k}$, $k = 2, \dots, K$. We can think of solving the general K-class problem in terms of solving $K - 1$ binary problems. In the kth binary problem, dummy values y_{ik} play the role of class labels with $\mathcal{Y} = \{0, 1\}$, while variables of the problem correspond to estimating the probabilities $P(y \geq k | x_i)$. Let us fix $k = 2, \dots, K$. We define the vector of estimators $\hat{\mathbf{q}}_k = (\hat{q}_{1k}, \dots, \hat{q}_{nk})$ of the probabilities $P(y \geq k | x_i)$, as the isotonic regression of vector $\mathbf{y}_k = (y_{1k}, \dots, y_{nk})$, i.e., the optimal solution to the problem:

$$\begin{aligned} &\text{minimize} \quad \sum_{i=1}^{n} (y_{ik} - p_i)^2 \\ &\text{subject to} \quad x_i \succeq x_j \Longrightarrow p_i \geq p_j \qquad i, j = 1, \dots, n. \end{aligned} \tag{12}$$

Having obtained the solution of (12) for each $k = 2, \dots, K$, we construct the estimators $\hat{p}_{ik}$ of $P(y = k | x_i)$ as

$$\hat{p}_{ik} = \begin{cases} \hat{q}_{ik} & \text{if } k = K, \\ \hat{q}_{ik} - \hat{q}_{i,k+1} & \text{if } 2 \leq k \quad < K, \\ 1 - \hat{q}_{i,k+1} & \text{if } k = 1. \end{cases} \tag{13}$$

These estimators are unique because the isotonic regression is unique. They boil down to the previous problem (11) in the binary-class case. However, as the $K - 1$ problems (12)

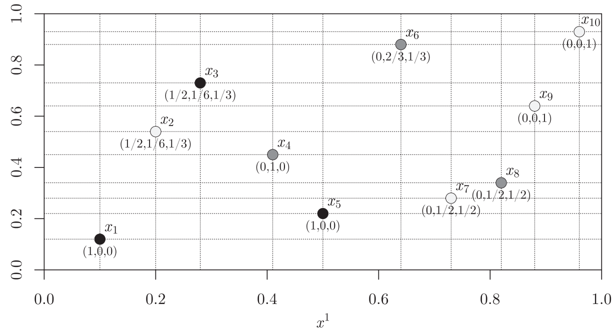

Fig. 2. Three-class example with two input variables. Objects with $y = 1$ are black, with $y = 2$ are gray, and with $y = 3$ are light. The vector of probability estimates $(\hat{p}_{i1}, \hat{p}_{i2}, \hat{p}_{i3})$ is shown below each objects. Notice that for consistent objects $(x_1, x_4, x_5, x_9, x_{10})$, the probability concentrates on a single class y_i.

are solved independently, we must guarantee that $\hat{q}_{ik} < \hat{q}_{i,k+1}$ will never happen, because otherwise we would have negative probabilities $\hat{p}_{ik}$:

Theorem 5. *For each $i = 1, \dots, n$, estimators $\{\hat{p}_{i1}, \dots, \hat{p}_{iK}\}$ form a probability distribution, i.e., $\sum_{k=1}^{K} \hat{p}_{ik} = 1$, and for each k, $\hat{p}_{ik} \geq 0$.*

Proof. It immediately follows from definition (13) that

$$\sum_{k=1}^{K} \hat{p}_{ik} = 1 - \hat{q}_{i,2} + \sum_{k=2}^{K-1} \left(\hat{q}_{ik} - \hat{q}_{i,k+1} \right) + \hat{q}_{iK} = 1.$$

Now, we prove the nonnegativity of $\hat{p}_{ik}$. First, notice that the isotonic regression (11) is bounded between 0 and 1. This follows from Theorem 3, because $y_{ik} \in \{0, 1\}$ and thus $Av(\mathbf{y}, A) \in [0, 1]$ for any subset A. This shows that $\hat{p}_{i1} \geq 0$ and $\hat{p}_{iK} \geq 0$. To show that $\hat{p}_{ik} \geq 0$ for $k = 2, \dots, K-1$, we must show that $\hat{q}_{ik} - \hat{q}_{i,k+1} \geq 0$. Since $y_{ik} = 1_{y_i \geq k} \geq 1_{y_i \geq k+1} = y_{i,k+1}$, we have that $Av(\mathbf{y}_k, A) \geq Av(\mathbf{y}_{k+1}, A)$ for every subset A. Thus, for *any* lower set L, such that $x_i \in L$, we must have

$$\max_{U : x_i \in U} Av(\mathbf{y}_k, L \cap U) \geq \max_{U : x_i \in U} Av(\mathbf{y}_{k+1}, L \cap U).$$

Then, however, it follows from Theorem (3), that $\hat{q}_{ik} \geq \hat{q}_{i,k+1}$. □

Thus, the problem of probability estimation for multiclass case is decomposed into $K - 1$ isotonic regression problems (12). The probability estimators are obtained each time from the optimal solution by (13). They always form a proper probability distribution, i.e., they are nonnegative and sum to unity. Theorem 4 applies for each $k = 2, \dots, K$.

A simple example of three-class problem is shown in Fig. 2.

4.3 Extension Beyond the Training Set

So far, we got estimators of the conditional probability distributions only for objects from the training set D, i.e., at the points x_i, $i = 1, \dots, n$. One can, however, simply extend the estimated probabilities to the whole space $\mathcal{X}$. Consider first the binary problem $\mathcal{Y} = \{0, 1\}$ and probability estimate $\hat{p}_i$ for object x_i. Since the estimates were obtained by solving the isotonic regression (11), it must hold $x_i \succeq x_j \rightarrow \hat{p}_i \geq \hat{p}_j$. The monotonicity constraint (6) for $K = 2$ states that the probability $p(x) = P(y = 1|x)$ is a monotone function. Therefore, a valid extension $\hat{p}(x)$ of the vector of estimators $\hat{\mathbf{p}} = (\hat{p}_1, \dots, \hat{p}_n)$ must satisfy two conditions:

1. $\hat{p}(x) = \hat{p}_i$ (the extension is consistent with respect to the estimators).
2. For every $x, x' \in \mathcal{X}$ it holds $x \succeq x' \rightarrow \hat{p}(x) \geq \hat{p}(x')$ (the extension is monotone).

Potharst and Feelders [5] considered extending the monotone functions from the training set to the whole space $\mathcal{X}$. They showed that there is a minimal and a maximal extension, defined as

$$\hat{p}^{\min}(x) = \max\{\hat{p}_i : x_i \preceq x\},$$
$$\hat{p}^{\max}(x) = \min\{\hat{p}_i : x_i \succeq x\},$$

and every valid extension $\hat{p}(x)$ satisfies $\hat{p}^{\min}(x) \leq \hat{p}(x) \leq \hat{p}^{\max}(x)$ for every $x \in \mathcal{X}$. Moreover, every monotone function satisfying the above condition is a valid extension. Therefore, we will consider a simple, yet sufficient, generalization of those two, defined as

$$p^{\lambda}(x) = \lambda \hat{p}^{\min}(x) + (1 - \lambda) \hat{p}^{\min}(x), \tag{14}$$

for $\lambda \in [0, 1]$. For $\lambda \in \{0, 1\}$, we retrieve back the minimal and maximal extensions.

In multiclass case, the situation is analogous. Instead of $\hat{p}_i$, we have the estimators $\hat{q}_{ik}$ of $P(y \geq k|x_i)$, for $k = 2, \dots K$, obtained from (12). The stochastic dominance principle is equivalent to stating that $P(y \geq k|x_i)$ must be a monotone function for each k. Therefore, the extensions $\{\hat{q}_2(x), \dots, \hat{q}_K(x)\}$ must be monotone. We can proceed analogously as before, defining the minimal, the maximal, and λ-parametrized extension $\hat{q}_k^{\min}(x)$, $\hat{q}_k^{\max}(x)$, and $\hat{q}_k^{\lambda}(x)$.

5 The Direct Method

In this section, we consider the direct approach to classification, defined in (4), based on the empirical risk minimization within the class of all monotone functions. Although such a class of functions cannot be described with a finite number of parameters, minimization can be done effectively because we are only interested in values of monotone functions at points from D. It is not hard to verify that a monotone function minimizing the empirical risk can be obtained by solving the following optimization problem:

$$\begin{aligned} \text{minimize:} \quad & \sum_{i=1}^{n} L(y_i, d_i) \\ \text{subject to:} \quad & x_i \succeq x_j \rightarrow d_i \geq d_j i, j = 1, \dots, n, \\ & d_i \in \{1, \dots, K\} i = 1, \dots, n, \end{aligned} \tag{15}$$

where d_i are variables of the problem (values of the optimal monotone function at points from D). The problem has one more interesting interpretation: relabel the objects to make the data set monotone, such that new class labels are as close as possible to the original class labels, where the closeness is measured in terms of the loss function. The new labels in (15) are the optimal values of variables d_i. Therefore, replacing the original labels by the new labels can be used in the preprocessing phase along with any classification algorithm, and corresponds to the nonparametric error correction. Since the nonparametric classification problem resembles isotonic regression (except that a discrete output variable is now considered), it will be called *isotonic classification*, and its optimal solution $\hat{\mathbf{d}}$ will be called isotonic classification of $\mathbf{y}$.

The problem (15) was already considered in [23], [40]. It was shown that the problem can be solved either by linear programming or by maximum network flow in $O(n^3)$. In both cases, (15) must be transformed to a more useful form. Let d_{ik}, for $k = 2, \dots, K$, be binary variables with the following interpretation: "$d_{ik} = 1$ iff new class label of object x_i is at least k." Such interpretation implies that $d_{ik} \geq d_{i,k+1}$; for instance, for $K = 5$, $d_i = 1$ is encoded as $[d_{i1} = 0, d_{i2} = 0, d_{i3} = 0, d_{i4} = 0]$, d_2 is encoded as $[1, 0, 0, 0]$, d_3 as $[1, 1, 0, 0]$, d_4 as $[1, 1, 1, 0]$, and d_5 as $[1, 1, 1, 1]$. Thus, the new label of object x_i can be obtained from $d_i = 1 + \sum_{k=2}^{K} d_{ik}$. The monotonicity of new labels implies that for any $x_i \succeq x_j$ we must have $d_{ik} \geq d_{jk}$ for each $k = 2, \dots, K$. Finally, the loss function can be reformulated as

$$\begin{aligned} L(y_i, d_i) &= \sum_{k=2}^{y_i} (L(y_i, k-1) - L(y_i, k))(1 - d_{ik}) \\ &\quad + \sum_{k=y_i+1}^{K} (L(y_i, k) - L(y_i, k-1)) d_{ik} \\ &= \sum_{k=2}^{K} (L(y_i, k) - L(y_i, k-1)) d_{ik} \\ &\quad + \sum_{k=y_i+1}^{K} (L(y_i, k-1) - L(y_i, k)), \end{aligned} \tag{16}$$

and the second sum on the right-hand side can be dropped, because it is constant. Thus, denoting $\delta(y, k) = L(y, k) - L(y, k-1)$, we transformed problem (15) into the following problem:

$$\begin{aligned} \text{minimize:} \quad & \sum_{i=1}^{n} \sum_{k=2}^{K} \delta(y_i, k) d_{ik} \\ \text{subject to:} \quad & x_i \succeq x_j \Longrightarrow d_{ik} \geq d_{jk} \quad i, j = 1, \dots, n, \\ & \qquad\qquad\qquad\qquad\quad k = 2, \dots, K, \\ & d_{i,k} \geq d_{i,k+1} \qquad\quad i = 1, \dots, n, \\ & \qquad\qquad\qquad\qquad\quad k = 2, \dots, K-1, \\ & d_{ik} \in \{0, 1\} \qquad\quad i = 1, \dots, n, \\ & \qquad\qquad\qquad\qquad\quad k = 2, \dots, K. \end{aligned} \tag{17}$$

This is a linear program with integer variables. However, the integer condition (last constraint) can be relaxed to $0 \leq d_{ik} \leq 1$ and we end up with an ordinary linear program. The relaxation of this constraint follows from the fact that the matrix of coefficients of the constraints is *totally unimodular* and the right hand sides of the constraints are integer. In such a case, every feasible (hence also every optimal) basic solution is always integer. Therefore, we do need to impose integer constraints, because we will obtain an integer solution anyway (see [23], [41] for more details).

Here, we will focus on two particular cases: binary classification problem, and multiclass classification with linear loss function (9). We will analyze computational and statistical properties of the method. We will also show how to cope with a drawback of isotonic classification: in most cases the optimal solution is not unique.

5.1 Reduction of the Problem Size

In this section, we provide a general method for reduction of the size of the isotonic classification problem. Chandrasekaran et al. [23] proposed a reduction method for binary classification. Here, we show how to reduce problem (15) in the most general case. For each x_i, let us define the *lower* and *upper class labels*, respectively, as

$$\begin{aligned} l_i &= \min\{y_j : x_j \succeq x_i, j = 1, \dots, n\} \\ u_i &= \max\{y_j : x_j \preceq x_i, j = 1, \dots, n\}. \end{aligned} \tag{18}$$

The following, simple lemma (proof of which can be found in the Appendix, available in the online supplemental material) states that lower and upper labels appears to be indicators of inconsistencies in the data set:

Lemma 1. *The following holds:*

1. $l_i \leq y_i \leq u_i$.
2. $u_i = l_i$ *if and only if* x_i *is consistent.*
3. *For each* $x_i \succeq x_j$, *we have* $l_i \geq l_j$ *and* $u_i \geq u_j$.

The isotonic classification problem, formulated as (17), has $n \times (K - 1)$ variables. Removing any of these variables is very desirable. Given the above lemma, we are able to obtain a priori the optimal values of some of the variables:

Theorem 6. *Let* $\hat{\mathbf{d}}$ *be any isotonic classification of* $\mathbf{y}$. *Then, we have* $l_i \leq \hat{d}_i \leq u_i$ *for each* $i = 1, \dots, n$.

The proof is in the Appendix, available in the online supplemental material. Theorem (6) implies that we can remove consistent objects from the optimization process, because we know a priori that $\hat{d}_i = y_i$ for such objects. Moreover, for other objects, we can bound the values of the variables d_i to a smaller range (l_i, u_i), which will speed up the optimization process. In many cases, this can dramatically reduce the size of the problem [26].

5.2 Binary Isotonic Classification

Let us consider the simplest problem of isotonic classification, when $K = 2$ and $\mathcal{Y} = \{0, 1\}$. Since $L(0, 0) = L(1, 1) = 0$ and the loss function is invariant under multiplication of every value by a constant factor, there is only one "degree of freedom" $\alpha = \frac{L(0,1)}{L(0,1)+L(1,0)}$. One can easily show that the Bayes classifier $h^*(x)$ has one of the following forms:

$$h^*(x) = 1_{P(y=1|x) \geq \alpha}, \qquad h^*(x) = 1_{P(y=1|x) > \alpha}, \tag{19}$$

or can be any monotone function between these two. The isotonic classification problem (15) can be presented in the simplified form: since we have $K = 2$, we can omit index k for the variables; moreover, by introducing

$$w_0 = \alpha, \qquad w_1 = 1 - \alpha, \tag{20}$$

we can write (15) as

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} w_{y_i} |y_i - d_i| \\ \text{subject to} \quad & x_i \succeq x_j \Longrightarrow d_i \geq d_j \qquad i, j = 1, \dots, n. \end{aligned} \tag{21}$$

As it was mentioned before, the integer constraint $d_i \in \{0,1\}$ can be relaxed to $0 \le d_i \le 1$ (due to unimodularity). Moreover, the relaxed constraint can further be dropped, because if there were any $d_i \ge 1$ (or $d_i \le 0$) in any feasible solution, we could decrease their values down to 1 (or increase up to 0), obtaining a new feasible solution with a smaller value of the objective function.

We transformed problem (15) into (21) to show that it strongly resembles isotonic regression (11). In the isotonic regression problem, we minimize L_2-norm (sum of squares) between vectors $\mathbf{y}$ and $\mathbf{p}$, while in (21) we minimize L_1-norm (sum of absolute values) between $\mathbf{y}$ and $\mathbf{d}$. In fact, both problems are closely connected. To investigate this issue in a greater detail, let us present a useful property of isotonic regression. Suppose $\hat{\mathbf{p}}$ is the isotonic regression of $\mathbf{y}$. By a *level set* of $\hat{\mathbf{p}}$, denoted by $[\hat{p} = a]$, we mean the subset of $\{1, \dots, n\}$ on which $\hat{\mathbf{p}}$ has constant value a, i.e., $[\hat{p} = a] = \{i : \hat{p}_i = a\}$. The following theorem holds:

Theorem 7 [30]. *Suppose $\hat{\mathbf{p}}$ is the isotonic regression of $\mathbf{y}$. If a is any real number such that the level set $[\hat{p} = a]$ is not empty, then $a = Av(\mathbf{y}, [\hat{p} = a])$.*

The following theorem holds (proven in the Appendix, available in the online supplemental material):

Theorem 8. *Let $\hat{\mathbf{p}}$ be the isotonic regression of $\mathbf{y}$. Then, the vectors $\hat{\mathbf{d}}_*$ given by $\hat{d}_{*i} = 1_{\hat{p}_i > \alpha}$, and $\hat{\mathbf{d}}^*$ given by $\hat{d}^*_i = 1_{\hat{p}_i \ge \alpha}$, are the isotonic classifications of $\mathbf{y}$. Moreover, if $\hat{\mathbf{d}}$ is any isotonic classification, it must hold $\hat{d}_{*i} \le \hat{d}_i \le \hat{d}^*_i$, for all $i = 1, \dots, n$. In particular, if $\hat{\mathbf{d}}_* = \hat{\mathbf{d}}^*$, then the isotonic classification is unique.*

Let us call $\hat{\mathbf{d}}^*$ the greatest, and $\hat{\mathbf{d}}_*$ the smallest isotonic classification of $\mathbf{y}$. Theorem 8 states that if the MLE estimator (isotonic regression) $\hat{p}_i$ is greater (or smaller) than α, then the optimal value for the corresponding variable $\hat{d}_i$ in the binary isotonic regression problem (21) is 1 (or 0). In other words, the functions $1_{\hat{p}_i \ge \alpha}$ and $1_{\hat{p}_i > \alpha}$, or any monotone function in between, minimize the loss function on the training set D. However, the above functions are counterpart of the Bayes classifier (19) and, thus, are the classifiers produced by the plug-in method. This means that *the plug-in method, and the direct method coincide for binary classification*.

It follows from Theorem 8 that the isotonic classification may be nonunique. In particular, the nonunique variables are exactly those d_i, for which $\hat{p}_i = \alpha$. This suggests a procedure of finding the greatest and the smallest isotonic classification. The idea is to solve (21) twice, first with perturbing α by a small positive amount ϵ, and then with perturbing α by a small negative amount ϵ. A solution to the first problem is the greatest isotonic classification, while a solution to the second problem—the smallest isotonic classification. Indeed, Theorem 7 states that for a given x_i, $\hat{p}_i$ is equal to the average of y_j over all the objects x_j having the same value $\hat{p}_j = \hat{p}_i$. In our case, since $y_i \in \{0,1\}$, every $\hat{p}_i$ must be of the form $\frac{r}{r+s}$, where $r, s \le n$. When α is emphnot of the form $\frac{r}{r+s}$ for some integers $r, s \le n$, then the binary isotonic classification is, thus, unique (there will be no $\hat{p}_i = \alpha$). On the other hand, if α *is* of the form $\frac{r}{r+s}$, increasing α by sufficiently small ϵ, such that $\alpha + \epsilon$ is not of the form $\frac{r}{r+s}$, and there is no other number $\gamma = \frac{r}{r+s}$ for some $r, s \le n$ such that $\alpha < \gamma < \alpha + \epsilon$, will give a unique isotonic classification equal the greatest α-binary isotonic classification, $\hat{\mathbf{d}}^*$ (because there is no $\hat{p}_i$ such that $\alpha < \hat{p}_i \le \alpha + \epsilon$). One can show that $\epsilon \le n^{-2}$ is sufficient. Similarly, decreasing α by ϵ will lead us to the smallest isotonic classification $\hat{\mathbf{d}}^*$. Thus, we have proved the following theorem.

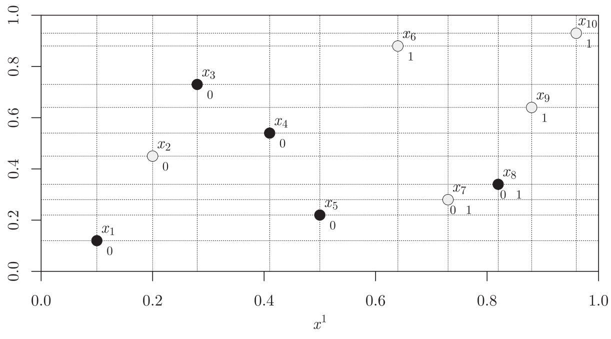

Fig. 3. Binary-class isotonic classification with $\alpha = \frac{1}{2}$; the data set is the same as in Fig. 1. The new class label is shown below on the right side of each object. The isotonic classification is not unique; two objects have class labels 0-1, which means that they are assigned label 0 in the smallest isotonic classification, and label 1 in the greatest isotonic classification.

Theorem 9. *If α is not of the form $\frac{r}{r+s}$ for some $r, s \le n$, the α-binary isotonic classification is unique. Otherwise, the greatest α-binary isotonic classification $\hat{\mathbf{d}}^*$ can be found by increasing the value of α by $\epsilon \le n^{-2}$ and solving problem (21). Similarly, the smallest α-binary isotonic classification $\hat{\mathbf{d}}_*$ can be found by decreasing the value of α by $\epsilon \le n^{-2}$ and solving again (21).*

A simple example of binary isotonic classification is shown in Fig. 3. Comparison with Fig. 1 shows relation between new labels and probability estimates, as stated in Theorem 8.

5.3 Linear Isotonic Classification

Let us analyze the problem of isotonic classification (15) with the linear loss (9), which will be called *linear isotonic classification*. We have already shown in Theorem 8 that there exists a correspondence between binary isotonic regression and binary isotonic classification, and therefore the direct and plug-in methods coincide. In the forthcoming theorem, we will show that an analogous relationship takes place between multiple isotonic regression and linear isotonic classification:

Theorem 10. *Let $\hat{\mathbf{q}}_k$ be the isotonic regression of $\mathbf{y}_k$, $k = 2, \dots, K$. Then, vectors $\hat{\mathbf{d}}_*$ and $\hat{\mathbf{d}}^*$ defined as $\hat{d}_{*i} = 1 + \sum_{k=2}^{K} 1_{\hat{q}_{ik} > \alpha}$ and $\hat{d}^*_i = 1 + \sum_{k=2}^{K} 1_{\hat{q}_{ik} \ge \alpha}$ are the linear isotonic classifications of $\mathbf{y}$. Moreover, every other linear isotonic classification $\hat{\mathbf{d}}$ satisfies $\hat{d}_{*i} \le \hat{d}_i \le \hat{d}^*_i$, $i = 1, \dots, n$.*

See the Appendix, available in the online supplemental material, for the proof. There are several important conclusions following from Theorem 10. First of all, the problem of isotonic classification (15) for extended linear loss can be solved by solving a sequence of $K - 1$ simple

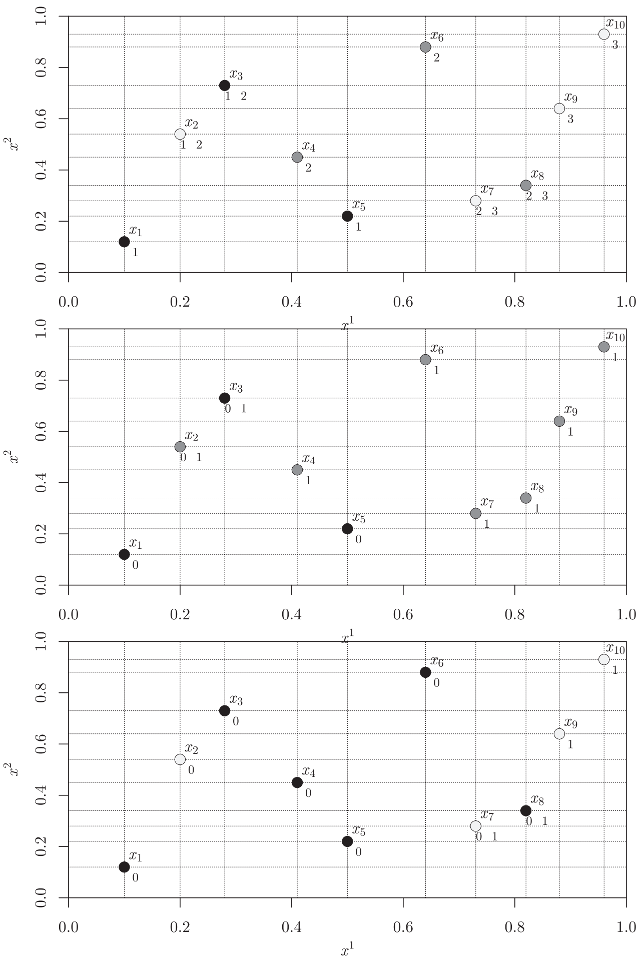

Fig. 4. Linear isotonic classification for $\alpha = \frac{1}{2}$. In the top figure, the 3-class data set is shown with new labels assigned; in the middle and bottom figures, there are shown the data sets used in two binary subproblems (with new labels). For example, label 2-3 means that the object is assigned label 2 in the lowest isotonic classification and label 3 in the greatest isotonic classification.

weighted binary problems. Although it seems that we now have $K-1$ problem instead of one problem, from the computational point of view this is a great gain, because we decomposed the problem with $(K-1) \times n$ variables into $K-1$ subproblems with n variables each.

Moreover, a closer look at how $\hat{\mathbf{d}}_*$ and $\hat{\mathbf{d}}^*$ are defined reveals an interesting fact: for each $i = 1, \ldots, n$, every $\hat{d}_i$ such that $\hat{d}_{*i} \leq \hat{d}_i \leq \hat{d}^*_i$ is the $(1-\alpha)$-quantile of the probability distribution $\{\hat{p}_{i1}, \ldots, \hat{p}_{iK}\}$, obtained in (13) from the multiple isotonic regression. Since the Bayes classifier for linear loss is the $(1-\alpha)$-quantile of the conditional probability distribution, $\hat{\mathbf{d}}_*$ and $\hat{\mathbf{d}}^*$ are the classifiers produced by the plug-in method. This means that *the plug-in method, and the direct method coincide for linear loss function*.

Finally, notice that similarly to the case of binary classification, we can give a simple procedure for finding the greatest and the smallest solutions. The idea is to find the greatest and the smallest solutions for each binary problem, by perturbing α by small amount $\pm\epsilon$, and then combine to separate solutions together. Due to space limit, we omit the details.

As an example, consider the example shown in Fig. 4, illustrating how the three-class problem is transformed to two binary problems. For simplicity we assume that $\alpha = \frac{1}{2}$, i.e., the loss function is an ordinary (symmetric) absolute error. Notice that for any object, the new class label in the top figure can be obtained from $\hat{d}_i = 1 + \sum_{k=2}^{K} \hat{d}_{ik}$, i.e., by summing up new class labels in middle and bottom figures and adding 1. Moreover, new class labels in the middle figure are always greater or equal than those on the bottom figure. This is exactly the conclusion of Theorem 10.

5.4 Extension Beyond the Training Set

The linear isotonic classification $\hat{\mathbf{d}}$ is defined only at training points x_i, $i = 1, \ldots, n$. Since we are dealing with a class of all monotone functions, we can extend isotonic classification to $\mathcal{X}$ by using any monotone function $\hat{h}: \mathcal{X} \to \mathcal{Y}$, such that $\hat{h}(x_i) = \hat{d}_i$, for each $i = 1, \ldots, n$. Similarly as in Section 4.3, we define the following minimal and maximal extensions:

$$\begin{aligned} \hat{h}^{\min}(x) &= \max\{\hat{d}_{*i} : x_i \preceq x\}, \\ \hat{h}^{\max}(x) &= \min\{\hat{d}^*_i : x_i \succeq x\}. \end{aligned} \tag{22}$$

We will now prove that every valid extension is bounded by $\hat{h}^{\min}(x)$ from below and by $\hat{h}^{\max}(x)$ from above:

Theorem 11. *Let $\hat{h}: \mathcal{X} \to \mathcal{Y}$ be monotone and suppose there exists an isotonic classification $\hat{\mathbf{d}}$ such that $\hat{h}(x_i) = \hat{d}_i$, for each $i = 1, \ldots, n$ (i.e., $\hat{h}(x)$ is a valid extension of the isotonic classification). Then, for each $x \in \mathcal{X}$, $\hat{h}^{\min}(x) \leq \hat{h}(x) \leq \hat{h}^{\max}(x)$.*

Proof. We know that for each i, $\hat{d}_{*i} \leq \hat{d}_i \leq \hat{d}^*_i$. For every $x \in \mathcal{X}$, since $\hat{h}(x)$ is monotone, we must have that if $x_i \preceq x$, then $\hat{h}(x) \geq \hat{h}(x_i) = \hat{d}_i$. But this means that $\hat{h}(x) \geq \max\{\hat{d}_i : x_i \preceq x\} \geq \max\{\hat{d}_{*i} : x_i \preceq x\} = \hat{h}^{\min}(x)$. Analogously, we can show that $\hat{h}(x) \leq \hat{h}^{\max}(x)$. □

6 Asymptotic Consistency

Let us consider the sequence of classifiers $(\hat{h}_1, \hat{h}_2, \ldots)$, denoted by $\hat{h}_n$, where each $\hat{h}_n$ is trained on a data set D_n of size n. We say that $\hat{h}_n$ is *strongly consistent* [31] if

$$\lim_{n \to \infty} L(h_n) = L^*,$$

with probability one, i.e., for almost every sequence of data sets $(D_1, D_2, \ldots)$. In other words, as the size of the training set increases, the risk of the classifier approaches the Bayes risk, i.e., $\hat{h}_n$ approaches the best possible classifier $\hat{h}^*$. In this section, we will consider the consistency of nonparametric classification with monotonicity constraints.

It is easy to show that the methods described in previous sections are *not* consistent for every distribution $P(x, y)$. Consider, for instance, the binary-class problem, let the input space be the unit square, $\mathcal{X} = [0, 1]^2$, so that $x = (x^1, x^2)$, and let the distribution be such that $P(x)$ puts all its mass uniformly on the diagonal of the square, i.e., on the points x, for which $x^1 = 1 - x^2$ (see Fig. 5). We can set $P(y = 1|x) = 1$ for x such that $x^1 \geq 1/2$, otherwise $P(y = 1|x) = 0$ (the distribution is then monotonically constrained), so that the

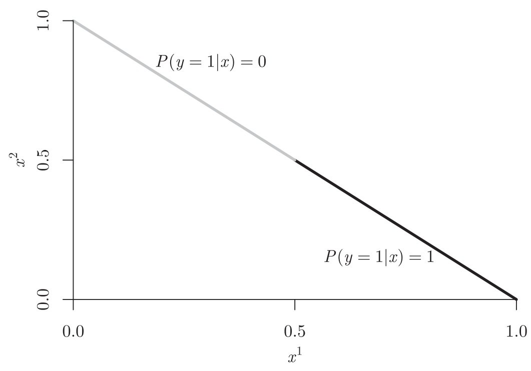

Fig. 5. A distribution, for which nonparametric classification is inconsistent.

Bayes risk is 0. Due to the form of $P(x)$, with probability one, none of the objects dominates any other object (they all lie at the diagonal). This means that to every point x at the diagonal, apart from a finite number of training data points, *any* extension of isotonic classification $\hat{h}_n$ (or plug-in classifier based on the isotonic regression) will assign the same output value, because, with probability one, there exists no x_i such that $x \succeq x_i$ or $x \preceq x_i$. Therefore, $L(\hat{h}_n) = 1/2$ for all n.

The above example shows the extreme case, in which inconsistency follows from the fact, that with probability one, the objects are incomparable and monotonicity constraints do not apply. This suggests that the properties of $P(x)$ play the most important role in establishing the consistency for nonparametric classification. This is indeed the case and, as the theorems below show, it is enough to assume that $P(x)$ has density on $\mathcal{X}$ (with respect to the Lebesgue measure) to make sure that nonparametric procedures are consistent.

We will use this result to show consistency of nonparametric methods of classification with monotonicity constraints. We first address the consistency of isotonic classification.

Theorem 12. *Let $\mathcal{X} = \mathbb{R}^m$ and assume $P(x, y)$ is monotonically constrained and $P(x)$ has density on $\mathcal{X}$. Let $\hat{h}_n$ be any valid extension of the linear isotonic classification trained on data set D of size n. Then, $\hat{h}_n$ is strongly consistent.*

The proof can be found in the Appendix, available in the online supplemental material. Our next results concern the consistency of plug-in classification methods. It comes as no surprise that it crucially depends on the behavior of isotonic regression for large sample sizes. Below, we show that under similar assumptions as before, isotonic regression is consistent in the mean squared error sense.

Lemma 2. *Let $\mathcal{X} = \mathbb{R}^m$, $\mathcal{Y} = \{0, 1\}$ and assume $P(x, y)$ is monotonically constrained and $P(x)$ has density on $\mathcal{X}$. Let $\hat{p}_n : \mathcal{X} \to [0, 1]$ be any valid extension of isotonic regression of the data set D of size n. Let $\eta(x) := P(y = 1|x)$. Then,*

$$\lim_{n \to \infty} \mathbb{E}[(\hat{p}_n(x) - \eta(x))^2 | D_n] = 0,$$

with probability one.

Lemma 2 will be used to show consistency of the plug-in classifier based on the isotonic regression:

Theorem 13. *Let $\mathcal{X} = \mathbb{R}^m$ and assume $P(x, y)$ is monotonically constrained and $P(x)$ has density on $\mathcal{X}$. Let $\hat{p}_{kn} : \mathcal{X} \to [0, 1]$, $k = 2, \ldots, K$, be any valid extension of multiple isotonic regression of the data set D of size n, and let $\hat{h}_n : \mathcal{X} \to \mathcal{Y}$ be the plug-in classifier defined as*

$$\hat{h}_n(x) = \arg\min_{k \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} L(y, k) \hat{p}_{yn}(x). \tag{23}$$

Then, $\hat{h}_n$ is strongly consistent.

Note that Theorem 13 does not put any constraints on the loss function. Theorem 13 is thus, in some sense, stronger than Theorem 12, as the latter assumes linear loss function, while the former makes no such assumption. Nevertheless, because isotonic classification is computationally easier than isotonic regression, and both direct and plug-in methods coincide for linear loss function, we advise that, in general, isotonic classification should be preferred in applications.

7 Experimental Results

In this section, we verify the performance of isotonic classification in the extensive computational experiment on both artificial and real data. We focus on both the prediction accuracy, and the computational speedup due to introduced decomposition methods.

7.1 Artificial Data

Isotonic classification replaces the original labels with new, monotone labels. It can, therefore, be used as an initial preprocessing procedure for relabeling the data set and making it consistent with the monotonicity constraints. In the experiment, we tested a few standard classification algorithms (such as decision trees or linear classifiers) on the data sets with monotonicity constraints. We wished to check if preprocessing the data with isotonic classification ("monotonizing" the training set) can increase the predictive performance of subsequently applied classification algorithm.

We performed the experiment on numerous artificial data sets. Working with artificial data allows a controlled variation of all parameters (number of attributes, objects, Bayes risk) and estimation of the predictive performance with a high accuracy. The data sets have been generated as follows: Objects $x = (x^1, \ldots, x^m) \in \mathbb{R}^m$ were generated uniformly on a cube $[0, 1]^m$. From the prediction point of view, the most important characteristic of the data is the underlying "target" function—the Bayes classification function $h^*(x) \in \{1, \ldots, K\}$. We assumed that $h^*(x) = k$ iff $\theta_{k-1} \le f(x) \le \theta_k$, where $f(x)$ is a real-valued function, described below, and $-\infty = \theta_0 < \theta_1 < \cdots < \theta_K = \infty$ are $K + 1$ thresholds. The noise (nonzero Bayes risk) was introduced to the model by randomly relabeling the objects from the Bayes label $h^*(x)$ to a randomly chosen label. The relabeling probability was set to get the prespecified Bayes risk R^*. The conditional distribution made in this way is monotonically constrained as long as the function $f(x)$ is monotone. The function $f(x)$ has the following form:

TABLE 1
The Average Accuracy (Absolute Error) of Classification Obtained on the Artificial Data

Dataset		C4.5		AdaBoost		Logistic		RankBoost	
R^*	m	orig	mon	orig	mon	orig	mon	orig	mon
0.1	4	0.464	**0.454**	0.365	**0.353**	0.215	**0.211**	0.303	**0.295**
	6	0.684	**0.677**	0.52	0.516	0.221	**0.219**	0.351	**0.345**
	8	0.841	0.838	0.613	0.613	0.223	0.223	0.387	0.386
	10	0.955	0.951	0.682	0.682	0.231	**0.231**	0.419	**0.417**
0.2	4	0.575	**0.542**	0.486	**0.453**	0.312	**0.304**	0.396	**0.377**
	6	0.78	**0.759**	0.62	**0.605**	0.321	**0.315**	0.435	**0.424**
	8	0.928	**0.919**	0.705	0.703	0.328	**0.324**	0.471	**0.465**
	10	1.024	1.024	0.774	0.768	0.34	**0.339**	0.506	0.505
0.3	4	0.688	**0.632**	0.602	**0.55**	0.414	**0.4**	0.486	**0.464**
	6	0.878	**0.839**	0.718	**0.694**	0.424	**0.413**	0.523	**0.508**
	8	1.013	**0.994**	0.803	**0.789**	0.437	**0.429**	0.558	**0.552**
	10	1.093	1.089	0.853	0.852	0.44	**0.436**	0.58	0.58
0.4	4	0.795	**0.711**	0.719	**0.641**	0.521	**0.5**	0.578	**0.549**
	6	0.975	**0.922**	0.824	**0.787**	0.529	**0.514**	0.614	**0.596**
	8	1.09	**1.062**	0.891	**0.872**	0.538	**0.526**	0.638	**0.63**
	10	1.162	**1.152**	0.938	**0.932**	0.544	**0.537**	0.663	**0.659**

The significantly higher result is marked with bold.

$$f(x) = \sum_{t=1}^{T} a_t r_t(x), \tag{24}$$

where a_t is positive and each $r_t(x)$ has one of the two following forms:

$$r_t(x) = \prod_{s=1}^{m_s} 1_{x^{j_s} \geq b^s} \qquad \text{or} \qquad r_t(x) = -\prod_{s=1}^{m_s} 1_{x^{j_s} \leq b^s},$$

where $j_s \in \{1, \ldots, m\}$ and $b^s \in [0, 1]$. One can show that $f(x)$ is a monotone function. Moreover, every monotone function can be approximated arbitrarily close (with respect to an L_p norm) by a function of the form (24). All parameters of the model, apart from T, R^*, and θ_k, were chosen at random: we set $a_t, b_s \sim U(0, 1)$; each j_s was chosen randomly from $\{1, \ldots, m\}$; values m_s are chosen according to the exponential law $P(m_s = j) = 2^{-j}$. Any of the two forms of $r_t(x)$ is equally likely. Parameter T models the "smoothness" of the function and is set to $100 \times m$ in the experiment. The thresholds θ_k were chosen so that the prior probabilities of all classes were equal: $P(y = k) = 1/K$. The parameters still to be determined were: number of classes K, sample size n, dimensionality m, and Bayes risk R^*. We chose $K = 5$, $n = 1{,}000$, $m \in \{4, 6, 8, 10\}$ and $R^* \in \{0.1, 0.2, 0.3, 0.4\}$ (Bayes risk is measured by the absolute error loss). For each combination of these parameters, we generated 20 models of the form (24). For each model, we trained the method on 10 separate sets of size 1,000 and tested on 10 separate sets of the same size.

We chose four state-of-the-art classification methods: C4.5 [42], AdaBoost [43] with C4.5 as a base learner (20 iterations), logistic regression, and RankBoost [44] with stump as a base learner (50 iterations). We used absolute error loss as a measure of accuracy. Since all the algorithms, except RankBoost, produce a conditional class distribution as an output, we adapted each algorithm to the absolute error loss by predicting with the median of the conditional distribution. RankBoost is designed to minimize the rank loss, but can be also easily adapted to deal with ordinal classification for any loss function, as described in [45]. Each classifier was learned in two copies, either with or without preprocessing with isotonic classification. In other words, one copy of the classifier was learned on the original data, while another copy—on the monotonized data, with inconsistencies removed. Each classifiers was run on 20 models, For each model, the average accuracy (absolute error) over 10 testing sets was calculated for both copies of the classifier. We then performed a sign test between models to verify if any of the methods is significantly better. We chose the significance level $\alpha = 0.05$, which means that one copy of the classifier must outperform another copy on at least 15 out of 20 models.

The results of the experiment are shown in Table 1. They unquestionably show that removing inconsistencies can only improve the accuracy. There is not even a single combination of parameters, for which monotonization would lead to a worse performance. The strength of the improvement, however, depends on the properties of the data set. The highest improvement is gained for high Bayes risk (large amount of noise), mostly due the fact that isotonic classification works as error correction based on the domain knowledge about the monotonicity. When the level of noise in the data is high, using knowledge about the probability distribution (monotonicity constraints) is especially beneficial, and relabeling the objects significantly decreases the amount of noisy labels. The improvement also decreases with the number of attributes m. This can be explained by observing that the dominance relation becomes sparse in high dimensions and only few objects are then relabeled.

In Table 2, we present the running times for isotonic classification on a standard laptop, with algorithms written in Java, using an open-source linear programming solver *lp_solve*. The running times were given for exemplary combinations of parameters (R^*, n, and m). They only concern the monotonization by isotonic classification, *not* the running times of classification algorithms. They are given in two versions, without and with the decomposition methods introduced in this paper (based on Theorem 6 and Theorem 10). It is clear that using our methods speeds up

TABLE 2
Computational Times (in Seconds) without and with Applying the Decomposition Methods

Dataset		$n = 200$		$n = 500$		$n = 1000$		$n = 2000$		$n = 5000$	
R^*	m	without	with	without	with	without	with	without	with	without	with
0.1	6	0.086	0.014	0.358	0.024	2.314	0.069	18.791	0.269	291.200	2.528
	8	0.024	0.003	0.127	0.015	0.605	0.057	4.343	0.220	79.080	1.724
0.2	6	0.042	0.004	0.466	0.016	3.091	0.089	23.969	0.405	385.611	7.255
	8	0.021	0.003	0.157	0.016	0.836	0.058	5.952	0.231	109.142	1.754

the computations at least several times, and often even by orders of magnitude.

7.2 Real Data

In the artificial data experiment, we have shown that isotonic classification can significantly improve prediction accuracy of standard classification algorithms when used as a preprocessing tool. In this section, we show on the real data sets that isotonic classification also works well as a stand-alone classifier, outperforming standard classification algorithms, which do not take ordinal properties of the data into account.

We used eight data sets, for which it is known from a domain knowledge that monotonicity constraints are present; three survey data sets (ESL, SWD and LEV) were obtained from [21], and five data sets (Housing, Breast cancer Wisconsin, Breast cancer Ljulbijana, Car, CPU) were collected from UCI repository [46]. A detailed characteristic of all data sets is shown in Table 3. We compared isotonic classification with three standard "off-the-shelf" classification algorithms used in machine learning: decision trees (C4.5), Naive Bayes and Support Vector Machines (SVM) [47]. The measure of error was the mean absolute error (MAE) and, as before, we adapted each algorithm to the absolute error loss by predicting with the median of the conditional distribution. The error of each classifier was estimated by a 10-fold cross validation, repeated 10 times to improve the replicability of the experiment. The results (average MAE and its standard deviation) are given in Table 4. The results show that isotonic classification, a simple classifier exploiting solely the dominance relation in the data, outperforms the standard classification algorithm in most of the cases.

8 Conclusions

We presented a statistical theory for ordinal classification with monotonicity constraints. While background knowledge often suggests consideration of these constraints with respect to real data sets, they are rarely taken into account in machine learning research. We considered the problem in its most general formulation, when the only knowledge about the set of objects is expressed solely through the dominance relation $\succeq$. We introduced a probabilistic model for ordinal classification with monotonicity constraints, based on the concept of stochastic dominance, and we investigated the possible loss functions in this setting. Our analysis suggests that convex losses are most suitable for ordinal classification with monotonicity constraints.

We also analyzed nonparametric classification methods. We considered both classification by estimating the class conditional distribution ("plug-in" method), and classification by minimization of the empirical risk (direct method). The plug-in approach is based on the multiple isotonic regression, while the empirical risk minimization approach (isotonic classification) is based on a linear program, and thus is computationally easier. We have shown that both approaches are closely related to each other, and they coincide for a linear loss function. We proposed how to decompose the general K-class classification problem into several binary-class subproblems. We have also shown how to speed up learning by exploiting some properties of the data. We investigated the asymptotic consistency of the considered methods. Finally, we verified our approach in the extensive computational experiment.

Theoretical analysis performed in this paper provides a necessary foundation for nonparametric methods of ordinal classification with monotonicity constraints. It also provides arguments which permit to claim that the proposed approach is computationally feasible. We hope that our results will be of interest for machine learning community, because monotone relationships are frequently encountered in the applications, and because nonparametric methods considered here play important role in many learning algorithms for ordinal classification with monotonicity constraints.

TABLE 3
Data Sets Used in Experiments

Data set	#attributes	#objects	#classes
ESL	4	488	8
SWD	10	1000	4
LEV	4	1000	5
Housing	8	506	4
Wisconsin	9	699	2
Ljubljana	8	286	2
Car	6	1728	4
CPU	6	209	4

TABLE 4
Results of the Experiment: Average MAE ± Standard Error of MAE

Dataset	IsoSep	J48	SVM	NB
ESL	**0.328**±0.023	0.369±0.022	0.355±0.023	**0.333**±0.024
SWD	**0.442**±0.018	**0.442**±0.016	**0.435**±0.016	0.457±0.016
LEV	**0.398**±0.017	0.415±0.018	0.444±0.016	0.441±0.017
Housing	**0.286**±0.02	0.332±0.023	0.314±0.025	0.506±0.033
CPU	**0.099**±0.02	**0.1**±0.019	0.371±0.03	0.18±0.033
Ljubljana	**0.241**±0.024	**0.259**±0.021	0.299±0.023	**0.252**±0.025
Wisconsin	**0.03**±0.007	0.046±0.009	**0.03**±0.007	0.037±0.007
Car	**0.045**±0.006	0.09±0.008	0.078±0.007	0.177±0.008

For each data set, results within one standard error from the best are marked with bold.

ACKNOWLEDGMENTS

The authors wish to acknowledge financial support from the Ministry of Science and Higher Education, grant N519 441939. The authors also wish to acknowledge helpful remarks on the previous version of this paper from three anonymous reviewers.

REFERENCES

[1] S. Greco, B. Matarazzo, and R. Słowiński, "Rough Sets Theory for Multicriteria Decision Analysis," *European J. Operational Research,* vol. 129, pp. 1-47, 2001.

[2] S. Greco, B. Matarazzo, and R. Słowiński, "Dominance-Based Rough Set Approach to Decision under Uncertainty and Time Preference," *Annals of Operations Research,* vol. 176, pp. 41-75, 2010.

[3] *Preference Learning,* J. Fürnkranz and E. Hüllermeier, eds. Springer, 2010.

[4] S. Greco, B. Matarazzo, and R. Słowiński, "Rough Set Approach to Customer Satisfaction Analysis," *Proc. Fifth Int'l Conf. Rough Sets and Current Trends in Computing,* pp. 284-295, 2006.

[5] R. Potharst and A.J. Feelders, "Classification Trees for Problems with Monotonicity Constraints," *SIGKDD Explorations,* vol. 4, no. 1, pp. 1-10, 2002.

[6] S. Greco, B. Matarazzo, and R. Słowiński, "A New Rough Set Approach to Evaluation of Bankruptcy Risk," *Operational Tools in the Management of Financial Risks,* C. Zopounidis, ed., pp. 121-136, Kluwer Academic Publishers, 1998.

[7] D. Gamarnik, "Efficient Learning of Monotone Concepts via Quadratic Optimization," *Proc. Conf. Computational Learning Theory,* pp. 134-143. 1998,

[8] J. Sill and Abu-Mostafa, "Monotonicity Hints," *Advances in Neural Information Processing Systems,* vol. 9, pp. 634-640, MIT Press, 1997.

[9] Y.U. Ryu, R. Chandrasekaran, and V. Jacob, "Data Classification Using the Isotonic Seeparation Technique: Application to Breast Cancer Prediction," *European J. Operational Research,* vol. 181, no. 2, pp. 842-854, 2007.

[10] A. Feelders and M. Pardoel, "Pruning for Monotone Classification Trees," *Proc. Fifth Int'l Symp. Advances in Intelligent Data Analysis V,* 2003.

[11] K. Cao-Van and B. De Baets, "An Instance-Based Algorithm for Learning Rankings," *Proc. Benelearn,* pp. 15-21, 2004.

[12] R. Słowiński, S. Greco, and B. Matarazzo, "Rough Sets in Decision Making," *Encyclopedia of Complexity and Systems Science,* R. Meyers, ed., pp. 7753-7786, Springer-Verlag, 2009.

[13] K. Dembczyński, W. Kotłowski, and R. Słowiński, "Ensemble of Decision Rules for Ordinal Classification with Monotonicity Constraints," *Proc. Third Int'l Conf. Rough Sets and Knowledge Technology,* pp. 260-267, 2008.

[14] J. Błaszczyński, R. Słowiński, and M. Szelag, "Sequential Covering Rule Induction Algorithm for Variable Consistency Rough Set Approaches," *Information Sciences,* vol. 181, pp. 987-1002, 2011.

[15] R. Słowiński and D. Vanderpooten, "A Generalized Definition of Rough Approximations Based on Similarity," *IEEE Trans. Knowledge and Data Eng.,* vol. 12, no. 2, pp. 331-336, Mar./Apr. 2000.

[16] W. Kotłowski and R. Słowiński, "Rule Learning with Monotonicity Constraints," *Proc. 26th Int'l Conf. Machine Learning (ICML '09),* pp. 537-544, 2009.

[17] A. Ben-David, "Monotonicity Maintenance in Information-Theoretic Machine Learning Algorithms," *Machine Learning,* vol. 19, no. 1, pp. 29-43, 1995.

[18] S. Giove, S. Greco, B. Matarazzo, and R. Słowiński, "Variable Consistency Monotonic Decision Trees," *Proc. Third Int'l Conf. Rough Sets and Current Trends in Computing,* pp. 247-254, 2002.

[19] K. Cao-Van and B. De Baets, "Growing Decision Trees in an Ordinal Setting," *Int'l J. Intelligent Systems,* vol. 18, pp. 733-750, 2003.

[20] J. Sill, "Monotonic Networks," *Advances in Neural Information Processing Systems,* vol. 10, pp. 661-667, MIT Press, 1998.

[21] A. BenDavid, L. Sterling, and Y.-H. Pao, "Learning and Classification of Monotonic Ordinal Concepts," *Computational Intelligence,* vol. 5, no. 1, pp. 45-49, 1989.

[22] W. Duivesteijn and A. Feelders, "Nearest Neighbour Classification with Monotonicity Constraints," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD '08),* pp. 301-316, 2008.

[23] R. Chandrasekaran, Y.U. Ryu, V.S. Jacob, and S. Hong, "Isotonic Separation," *INFORMS J. Computing,* vol. 17, no. 4, pp. 462-474, 2005.

[24] R. Herbrich, T. Graepel, and K. Obermayer, "Regression Models for Ordinal Data: A Machine Learning Approach," technical report, Technical Univ. of Berlin, 1999.

[25] H.-T. Lin and L. Li, "Ordinal Regression by Extended Binary Classifications," *Proc. Advances in Neural Information Processing Systems,* vol. 19, pp. 865-872, 2007.

[26] W. Kotłowski and R. Słowiński, "Statistical Approach to Ordinal Classification with Monotonicity Constraints," *Proc. Preference Learning ECML/PKDD '08 Workshop,* 2008.

[27] N. Barile and A. Feelders, "Nonparametric Monotone Classification with MOCA," *Proc. IEEE Eighth Int'l Conf. Data Mining (ICDM '08),* pp. 731-736, 2008.

[28] H.D. Brunk, "Maximum Likelihood Estimates of Monotone Parameters," *Annals of Math. Statistics,* vol. 26, no. 4, pp. 607-616, 1955.

[29] M. Ayer, H.D. Brunk, G.M. Ewing, W.T. Reid, and E. Silverman, "An Empirical Distribution Function for Sampling with Incomplete Information," *Annals of Math. Statistics,* vol. 26, no. 4, pp. 641-647, 1955.

[30] T. Robertson, F.T. Wright, and R.L. Dykstra, *Order Restricted Statistical Inference.* John Wiley & Sons, 1998.

[31] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition,* first ed. Springer, 1996.

[32] H. Levy, *Stochastic Dominance.* Kluwer Academic Publishers, 1998.

[33] K. Cao-Van, "Supervised Ranking, from Semantics to Algorithms," PhD dissertation, Ghent Univ., 2003.

[34] J.O. Berger, *Statistical Decision Theory and Bayesian Analysis.* Springer, 1993.

[35] K. Dembczyński, S. Greco, W. Kotłowski, and R. Słowiński, "Statistical Model for Rough Set Approach to Multicriteria Classification," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML PKDD '07),* pp. 164-175, 2007.

[36] W. Kotłowski, K. Dembczyński, S. Greco, and R. Słowiński, "Stochastic Dominance-Based Rough Set Model for Ordinal Classification," *Information Sciences,* vol. 178, no. 21, pp. 3989-4204, 2008.

[37] A. Feelders, "A New Parameter Learning Method for Bayesian Networks with Qualitative Influences," *Proc. 23rd Conf. Uncertainty in Artificial Intelligence (UAI '07),* pp. 117-124, 2007.

[38] O. Burdakov, O. Sysoev, A. Grimvall, and M. Hussian, "An $o(n^2)$ Algorithm for Isotonic Regression," *Large-Scale Nonlinear Optimization,* series Nonconvex Optimization and Its Applications, vol. 83, pp. 25-33, Springer, 2006.

[39] W.L. Maxwell and J.A. Muchstadt, "Establishing Consistent and Realistic Reorder Intervals in Production-Distribution Systems," *Operations Research,* vol. 33, pp. 1316-1341, 1985.

[40] R. Dykstra, J. Hewett, and T. Robertson, "Nonparametric, Isotonic Discriminant Procedures," *Biometrica,* vol. 86, no. 2, pp. 429-438, 1999.

[41] C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization.* Dover, 1998.

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

[43] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[44] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer, "An Efficient Boosting Algorithm for Combining Preferences," *J. Machine Learning Research,* vol. 4, pp. 933-969, 2003.

[45] H.-T. Lin and L. Li, "Large-Margin Thresholded Ensembles for Ordinal Regression: Theory and Practice," *Proc. 17th Int'l Conf. Algorithmic Learning Theory,* pp. 319-333, 2006.

[46] A. Asuncion and D. Newman, "UCI Machine Learning Repository," http://www.ics.uci.edu/mlearn/MLRepository.html, 2007.

[47] B.E. Boser, I. Guyon, and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proc. Fifth Ann. Workshop Computational Learning Theory (COLT),* pp. 144-152, 1992.

Wojciech Kotłowski received the BSc, MSc, and PhD degrees in computer science from Poznań University of Technology, Poland, in 2002, 2004, and 2009, respectively, and the MSc degree in physics from Adam Mickiewicz University, Poland, in 2006. In 2009-2012, he was a postdoctoral fellow at Centrum Wiskunde & Informatica (CWI), in the Netherlands. Meanwhile, he held several research appointments at the University of California at Santa Cruz. He is currently employed as an assistant professor at Poznań University of Technology. His scientific interests include online machine learning, preference learning, and statistical decision theory.

Roman Słowiński is a professor and the founding head of the Laboratory of Intelligent Decision Support Systems within the Institute of Computing Science, Poznań University of Technology. As a member of the Polish Academy of Sciences (PAS), he is currently the president of the Poznań Branch of the PAS. His area of expertise covers multiple criteria decision aiding, preference modeling, rough set theory, and knowledge discovery. He is the author or coauthor of 14 books and more than 200 papers in major scientific journals. He is a laureate of the EURO Gold Medal (1991), and Doctor *Honoris Causa* of Polytech Mons (2000), University Paris Dauphine (2001), and Technical University of Crete (2008). He received, moreover, the Edgeworth-Pareto Award, by International Society on Multiple Criteria Decision Making (1997) and the 2005 Prize of the Foundation for Polish Science, regarded as the most prestigious scientific award in Poland. Since 1999, he has been the editor-in-chief of the *European Journal of Operational Research*. He is a senior member of the IEEE.