Statistical Approach to Ordinal Classification with Monotonicity Constraints

Wojciech Kotłowski[1] and Roman Słowiński[1,2]

[1] Institute of Computing Science, Poznań University of Technology, 60-965 Poznań, Poland
wkotlowski@cs.put.poznan.pl rslowinski@cs.put.poznan.pl
[2] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. In the ordinal classification with monotonicity constraints, it is assumed that the class label of an object does not decrease when evaluations of this object on considered attributes increase. In this paper, we formulate the problem of ordinal classification with monotonicity constraints from statistical point of view, by imposing constraints both on the probability distribution and on the loss function. We propose a procedure for "monotonizing" the data by relabeling objects, based on minimization of the empirical risk in the class of all monotone functions. The procedure is then used as a preprocessing tool, improving the accuracy of the classifiers. We verify these claims in a computational experiment.

1 Introduction

Utilizing the domain knowledge is of fundamental importance in the learning process. In order to generalize, every learning algorithm must be biased, however, when this bias comes from the domain knowledge, the algorithm has a chance to generalize better. A common type of knowledge encountered in applications of machine learning are the statements about orders and about relationships among these orders. Indeed, although the experts may fail in providing quantitative relationships between variables of the system under study, they can usually describe their qualitative characteristic in terms of ordering and monotone dependencies (e.g., "the higher, the better"), which happen to be the relations that are easiest to express.

In the problem of *ordinal classification* (also referred to as *ordinal regression*), the purpose is to predict for a given object $x \in X$ one of K ordered class labels (ranks), $y \in Y = \{1, \ldots, K\}$. It is often assumed that the objects are described in terms of m attributes, so that, without loss of generality, each object is represented by a real-valued vector, $x = (x^1, \ldots, x^m) \in \mathbb{R}^m$. In this paper we consider the ordinal classification problem together with the domain knowledge about the objects expressed by the *monotonicity constraints*, imposed on X. Assume the attributes have ordered domains and it follows from the domain knowledge that a higher value of an object on an attribute, with other values being fixed, should not decrease its class assignment. We define a partial preorder

relation $\succeq$ in the set of objects X, called *dominance relation*, such that an object x dominates x', $x \succeq x'$, if $x^j \geqslant x'^j$ for $j = 1, \ldots, m$. The monotonicity constraints are then expressed in the following way: for any two objects $x, x' \in X$, if $x \succeq x'$, then object x should be labeled with a class not lower than x'.

As an example, consider the customer satisfaction analysis [1], which aims at determining customer preferences in order to optimize decisions about strategies for launching new products or about improving the image of existing products. The monotonicity constraints are of fundamental importance here. Indeed, consider two customers, A and B, and suppose that the evaluations of a product by customer A on a set of attributes are better than the evaluations by customer B. In this case, it is reasonable to expect that also the comprehensive evaluation of this product (its rank) by customer A is better (or at least not worse) than the comprehensive evaluation made by customer B.

As another example, consider the problem of house pricing, i.e. classification of houses with respect to their prices, into one of the following classes: "cheap", "moderate", "expensive", "very expensive". The classification is based on the following attributes: lot size, number of bedrooms, garages, whether house contains air conditioning, basement, etc. [2]. It is apparent that the price of the house A should not be less than of the house B if, for instance, house A has greater number of bedrooms and garages than B, and opposite to B, has basement, and is as good as B on the other attributes.

The problems of ordinal classification with monotonicity constraints are also referred to as *monotone classification* problems, and are commonly encountered in real-life applications, where the ordinal and monotone properties follow from the domain knowledge, e.g. bankruptcy risk prediction [3], medical diagnosis [4], house pricing [5], surveys data [6] and many others. Moreover, such problems are widely considered under a common name *multiple criteria sorting* within multiple criteria decision analysis [7].

Since the partial order can be represented by a directed graph, monotone classification bears resemblance to learning a graph labeling [8]; nevertheless, the meaning of the arcs in a graph labeling is different from the meaning of monotonicity constraints. On the other hand, the considered problem is the constrained case of ordinal classification. Although ordinal classification receives a growing attention in machine learning community [9,10,11,12], the monotonicity constraints are rarely considered. There are only a few methods which take the monotone nature of data into account; let us mention Dominance-based Rough Set Approach (DRSA) [7,13,14], monotone classification trees [15,16,17,5], monotone networks [18], instance-based methods [19,6] or isotonic separation [4]. DRSA provides the most comprehensive theory for ordinal classification problems with monotonicity constraints, based on the rough set approach to knowledge discovery.

Our paper consist of two parts. In the first part, we aim at providing a statistical framework for ordinal classification with monotonicity constraints. We show, how such constraints can be expressed by making general assumptions about the probability distribution. Moreover, we formulate the necessary and

sufficient conditions for the structure of the loss function which ensures the monotonicity of the Bayes classification function. The analysis suggests that some loss functions, such as 0-1 loss, are not suitable for monotone classification.

In the second part of the paper, we propose a general method for incorporating monotonicity constraints into the learning process. The method is called *monotone approximation* and is based on relabeling the training objects to remove the inconsistencies and "monotonize" the data. This is done by empirical risk minimization within the class of all monotone functions. We show how our method can be used as a preprocessing tool in combination with *any* algorithm for ordinal classification. The main idea is to apply our method to the original data, "monotonize" it and pass it to the learning algorithm, in order to improve both accuracy and readability of the classifier. This approach has been verified in computational experiment.

2 Problem Statement

The problem of ordinal classification can be stated as a problem of finding the function $f: X \to Y$, that accurately predicts values of y. The accuracy is measured in terms of the loss function $L(y, f(x))$, which is the penalty for predicting $f(x)$ when the actual value is y. The overall accuracy of the function f is defined as the expected loss (risk) according to the probability distribution $P(x, y)$ of data to be predicted:

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

The loss function in the classification problems has the form of a matrix, hence we denote $l_{yk} = L(y, k)$ where $y, k = 1, \ldots, K$. We assume that $l_{kk} = 0$ and $l_{yk} > 0$ if $y \neq k$. Moreover, in the ordinal classification the loss should 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. Therefore, following [12], we assume the loss matrix is *V-shaped*: for $k \leqslant y$ it holds $l_{y,k-1} \geqslant l_{yk}$, while for $k \geqslant y$ it holds $l_{yk} \leqslant l_{y,k+1}$. Notice that the proposed model is in the spirit of [20,12] and is different from the previous rank loss formulation found in [9].

A *Bayes classification function* is a function f^* minimizing the risk (1). Since $P(x, y)$ is unknown, f^* is unknown and the classification function is learned from a set of n training examples $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ (training set). In order to minimize the value of risk (1), the learning procedure usually performs minimization of empirical risk:

$$R_{\text{emp}}(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)), \tag{2}$$

which is the value of a loss function on the training set. In the paper, we denote objects from the training set D by x_i or x_j, where $i, j = 1, \ldots, n$; any objects from X are denoted by x or x'.

Stochastic dominance. The monotonicity constraints require that if $x \succeq x'$ then x should be assigned a class not lower than x'. In practice, these constraints are not always satisfied, leading to the situations referred to as *inconsistencies*. This suggests that the order relation $\succeq$ does not impose "hard" constraints and the constraints should rather be defined in a probabilistic setting. Let $x, x' \in X$ be such that $x \succeq x'$. Then, the minimal constraint, which should be imposed on the conditional distribution is the following:

$$P(y \geqslant k|x) \geqslant P(y \geqslant k|x'), \tag{3}$$

for each $k = 2, \ldots, K$. Expression (3) states that the probability of an event $\{y \geqslant k\}$ grows monotonically with the dominance relation. It is known as *(first order) stochastic dominance* [21]. We will say that a probability distribution is *monotonically constrained* if it satisfies the stochastic dominance relation (3) for each $x, x' \in X$ such that $x \succeq x'$. The stochastic dominance relation is the core of what we understand by monotonicity constraints. Notice that in [6], stochastic dominance was also used, but to define the properties of an estimator rather than the properties of a probabilistic model. In [9], one assumes that stochastic dominance holds for all $x, x' \in X$, whereas we assume it is induced by the dominance relation.

We now prove a simple lemma related to the stochastic dominance, which will be used later in this paper. The lemma is a basic result in decision theory, but we give the proof for clarity and completeness.

Lemma 1. *Let $x \succeq x'$ so that $P(y|x)$ stochastically dominates $P(y|x')$. Let $z\colon Y \to \mathbb{R}$ be a non-increasing random variable. Then it holds:*

$$\mathbb{E}[z|x] \leqslant \mathbb{E}[z|x'], \tag{4}$$

i.e. the expected value of z according to distribution $P(y|x)$ is always smaller then the expected value of z according to $P(y|x')$.

Proof. Let us denote $p_k = P(y = k|x)$, $q_k = P(y = k|x')$ and $z_k = z(k)$. Then (4) can be rewritten in the following way:

$$\sum_{k=1}^{K} p_k z_k \leqslant \sum_{k=1}^{K} q_k z_k.$$

Let us denote the cumulative distribution as $P_k = \sum_{l=1}^{k} p_l$ and $Q_k = \sum_{l=1}^{k} q_l$ (we assume $P_0 = Q_0 = 0$). From the stochastic dominance it follows that $P_k \leqslant Q_k$ for each k. Then:

$$\begin{aligned}\sum_{k=1}^{K} p_k z_k &= \sum_{k=1}^{K} (P_k - P_{k-1}) z_k = \sum_{k=1}^{K} P_k z_k - \sum_{k=0}^{K-1} P_k z_{k+1} = \\ &= z_K + \sum_{k=1}^{K-1} P_k (z_k - z_{k+1}) \leqslant z_K + \sum_{k=1}^{K-1} Q_k (z_k - z_{k+1}) = \sum_{k=1}^{K} q_k z_k, \end{aligned} \tag{5}$$

where the inequality follows from the fact that $z_k - z_{k+1} \geqslant 0$ for each k.□

3 Monotonicity of the Bayes classification function

Let us call a function $f\colon X \to Y$ *monotone* if for any $x, x' \in X$ it holds $x \succeq x' \to f(x) \geqslant f(x')$. Moreover, let us call vector $\mathbf{v} = (v_1, \ldots, v_n)$ monotone, if for every $i, j = 1, \ldots, n$ it holds that $x_i \succeq x_j \to v_i \geqslant v_j$.

In the classification problem, we aim at finding the classifier which is as close as possible to the Bayes classification function. In other words, the Bayes classification function is our "target function" which we try to approximate. Thus, not surprisingly, we require that in the ordinal classification with monotonicity constraints the Bayes classification function must be monotone[3].

However, although the probability distribution is monotonically constrained, the monotonicity of the Bayes classification function does not always hold. For instance, the Bayes classification function for 0-1 loss (the mode of the distribution) is not monotone under stochastic dominance assumption. It appears that some specific constraints must be imposed on the loss function in order to maintain the monotonicity of the Bayes classification function:

Theorem 1. *Let l_{yk} be V-shaped. The Bayes classification function is a monotone function for every monotonically constrained distribution $P(x, y)$ if and only if the loss function satisfies the following constraints:*

$$\begin{array}{ll} l_{y,k+1} - l_{yk} \geqslant l_{y+1,k+1} - l_{y+1,k} & \text{if } k > y \\ l_{y,k-1} - l_{yk} \geqslant l_{y-1,k-1} - l_{y-1,k} & \text{if } k < y \end{array} \tag{6}$$

Proof. We prove the "if" part. Suppose conditions (6) hold. Let us define δ_{yk} in the following way:

$$\delta_{yk} = \begin{cases} l_{yk} - l_{y,k-1} & \text{for } k > y, \\ l_{yk} - l_{y,k+1} & \text{for } k < y. \end{cases}$$

Let $P(x, y)$ be any monotonically constrained probability distribution and let $x, x' \in X$ be any two points such that $x \succeq x'$. Let us denote $p_k = P(y = k|x)$ and $q_k = P(y = k|x')$. Let u be a predicted class label. The expected loss for the prediction u according to the distribution $P(y|x)$ is as follows:

$$\mathbb{E}[L(y, u)|x] = \sum_{y=1}^{K} p_y l_{yu} = \sum_{y=1}^{u-1} p_y \sum_{k=y+1}^{u} (l_{yk} - l_{y,k-1}) + \sum_{y=u+1}^{K} p_y \sum_{k=u}^{y-1} (l_{yk} - l_{y,k+1}),$$

or by using δ_{yk}:

$$\mathbb{E}[L(y, u)|x] = \sum_{y=1}^{u-1} p_y \sum_{k=y+1}^{u} \delta_{yk} + \sum_{y=u+1}^{K} p_y \sum_{k=u}^{y-1} \delta_{yk}.$$

[3] The Bayes classification function may not be unique, because it is defined only up to a zero measure set. To avoid this problem, we assume that for *every* $\mathbf{x} \in X$, the Bayes function 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.

Consider $\Delta(u|x) = \mathbb{E}[L(y, u+1)|x] - \mathbb{E}[L(y, u)|x]$, the difference between the expected losses for $u+1$ and u:

$$\Delta(u|x) = \sum_{y=1}^{u} p_y \sum_{k=y+1}^{u+1} \delta_{yk} + \sum_{y=u+2}^{K} p_y \sum_{k=u+1}^{y-1} \delta_{yk} - \sum_{y=1}^{u-1} p_y \sum_{k=y+1}^{u} \delta_{yk} - \sum_{y=u+1}^{K} p_y \sum_{k=u}^{y-1} \delta_{yk} =$$

$$= \sum_{y=1}^{u} p_y \delta_{y,u+1} - \sum_{y=u+1}^{K} p_y \delta_{yu} \leqslant \sum_{y=1}^{u} q_y \delta_{y,u+1} - \sum_{y=u+1}^{K} q_y \delta_{yu} = \Delta(u|x'),$$

where the inequality comes from Lemma 1, since the function $z(1) = \delta_{1,u+1}, \ldots, z(u) = \delta_{u,u+1}, z_{u+1} = -\delta_{u+1,u}, \ldots, z(K) = -\delta_{Ku}$ is non-increasing from the assumptions (6). This means that the difference in expected loss for any two contiguous class labels $u+1$ and u does not increase as we move from x to x'. But this means that the difference in expected loss between *any* class labels v and u does not increase.

Now, suppose v is a Bayes classification function for x', i.e.:

$$v = \arg\min_{k \in Y} \mathbb{E}[L(y, k)|x'].$$

Choose some $u < v$. We have:

$$0 > \mathbb{E}[L(y, v)|x'] - \mathbb{E}[L(y, u)|x'] \geqslant \mathbb{E}[L(y, v)|x] - \mathbb{E}[L(y, u)|x],$$

which means that u cannot be the Bayes classification function for x. Thus, Bayes classification function must be monotone.

The "only if" part of the proof is long and quite technical, hence we give only the sketch of the proof. Suppose that one of the conditions (6) is violated; without loss of generality, we may assume that the first one is violated, i.e. $l_{y_0 k_0} - l_{y_0, k_0 - 1} < l_{y_0+1, k_0} - l_{y_0+1, k_0 - 1}$ for some $k_0 > y_0$. We shall find some probability distribution and objects $x \succeq x'$ such that the Bayes classification function violates monotonicity condition, i.e. $f^*(x) < f^*(x')$. However, we can set $P(y = k|x) = 0$ for each $x \in X$, for every class label $k \notin \{y_0, k_0, k_0 - 1\}$. This will effectively eliminate other classes (they never occur in the problem) so that we end up with three-class problem which is much easier to analyze than a general K-class problem. The rest of the proof consist in constructing the distributions $P(y|x)$ and $P(y|x')$ such that $f^*(x) < f^*(x')$. □

From Theorem 1 it follows that conditions (6) are necessary and sufficient for monotonicity of the Bayes classification function. The latter property is desired in the monotone classification, otherwise there would be no point in minimizing the risk within the class of monotone functions. Therefore we will call the loss function satisfying (6) a *monotone loss function*.

Monotone Bayes classification function and convexity. We investigate the conditions (6) for a popular subclass of the loss functions. Let us call a function $c: \mathbb{Z} \to \mathbb{R}$ *convex*[4] if for all $i, j \in \mathbb{Z}$ and for every $\lambda \in [0, 1]$ such that

[4] We denote the set of integers by $\mathbb{Z}$.

$\lambda i + (1-\lambda)j \in \mathbb{Z}$, we have:

$$c(i\lambda + (1-\lambda)j) \leqslant \lambda c(i) + (1-\lambda)c(j)$$

The loss function is very often expressed in the form $l_{yk} = c(y-k)$, with $c(0) = 0$ and $c(k) > 0$ for $k \neq 0$. The loss functions of such type are, for instance, 0-1 loss ($c(k) = 1_{k\neq 0}$)[5], mean 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 $l_{12} = c(-1)$ and $l_{21} = c(1)$.

Theorem 2. *Let $l_{yk} = c(y-k)$ be the V-shaped loss function and $K \geqslant 3$. Then the Bayes classification function $f^*(x)$ is monotone if and only if $c(k)$ is convex.*

Proof. One can show by induction that the function $c\colon \mathbb{Z} \to \mathbb{R}$ is convex if and only if for each $k \in \mathbb{Z}$ it holds:

$$c(k) \leqslant \frac{c(k-1) + c(k+1)}{2}. \tag{7}$$

The conditions (6) can now be expressed as:

$$\begin{array}{ll} c(y-k-1) - c(y-k) \geqslant c(y-k) - c(y-k+1) & \text{if } k > y, \\ c(y-k+1) - c(y-k) \geqslant c(y-k) - c(y-k-1) & \text{if } k < y, \end{array}$$

which is equivalent (along with condition $c(0) \leqslant \frac{c(1)+c(-1)}{2}$ holding for every loss matrix) to the condition (7).□

Corollary 1. *Let $l_{yk} = |y-k|^p$, for $p \geqslant 0$, be the loss function and $K \geqslant 3$. Then the Bayes classification function is monotone if and only if $p \geqslant 1$.*

Corollary 1 explains why 0-1 loss ($p \to 0$) does not lead to the monotone Bayes classification function under the stochastic dominance assumption, while absolute error loss ($p = 1$) and squared-error loss ($p = 2$) do ensure monotonicity. Those results support our opinion that 0-1 is not a proper loss for ordinal classification with $K \geqslant 3$, if one assumes stochastic dominance between the conditional distributions (the stochastic dominance assumption is related not only to the monotonicity constraints – see e.g. [9]). Notice, that for $K = 2$, all loss functions are monotone.

Linear loss function. Let us consider a specific class of the loss functions, called *linear loss functions* [22], defined as:

$$l_{yk} = \begin{cases} \alpha(k-y) & \text{if } k > y \\ (1-\alpha)(y-k) & \text{if } k \leqslant y, \end{cases} \tag{8}$$

where $0 < \alpha < 1$. For $\alpha = \frac{1}{2}$ we have an absolute error loss $l_{yk} = |k-y|$ (up to the proportional constant). The purpose of introducing (8) is to model asymmetric

[5] 1_A is an indicator function equals 1 if A is true, otherwise 0.

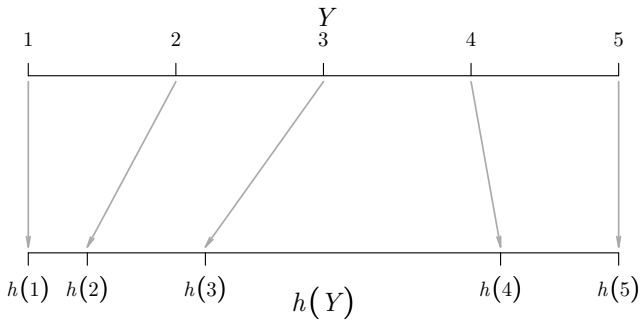


Fig. 1. Example with $K = 5$. The function $h(k)$ changes the position of each class label on the scale.

costs of misclassification: for $\alpha > \frac{1}{2}$, predicting higher class then the actual class y is more penalized than predicting the lower class; for $\alpha < \frac{1}{2}$ we have the opposite case. Such loss function can be useful e.g. in medicine: consider classifying patient into classes according to her/his health condition: "good", "moderate", "bad", "very bad". Then classifying the patient's condition to be better than it really is, will probably be more dangerous to her/his health than regarding the patient to be in worse condition than the real one.

It is easy to check that the linear loss is a monotone loss function. It is also known [22], that such loss function is minimized by $(1-\alpha)$-quantile of the conditional distribution[6] i.e., by such $y_{1-\alpha}$ that $P(y \leqslant y_{1-\alpha}) \geqslant 1-\alpha$ and $P(y \geqslant y_{1-\alpha}) \geqslant \alpha$. For $\alpha = \frac{1}{2}$ this results in the definition of a median.

One can extend the linear loss in the following way:

$$l_{yk} = \begin{cases} \alpha(h(k) - h(y)) & \text{if } k > y \\ (1-\alpha)(h(y) - h(k)) & \text{if } k \leqslant y, \end{cases} \tag{9}$$

where $h(k)\colon Y \to \mathbb{R}$ is a strictly increasing function. It can be interpreted as a "scale changing" function, which positions class labels on the real axis, thus changing the distances between them. Introducing an arbitrary scale may at first look like a strong generalization of (8). Surprisingly, an arbitrary scale will not change anything on the population level:

Theorem 3. *The Bayes classification function for the extended linear loss (9) does not depend on the function $h(k)$. In particular, this implies that Bayes classification function for extended linear loss is the $(1-\alpha)$-quantile of the conditional distribution.*

Proof. First notice, that since $h(k)$ is strictly increasing, it has an inverse $h^{-1}\colon h(Y) \to Y$. Let $y \in Y$ be a random variable according to distribution $P(y|x)$. Let us define the random variable $y' = h(y)$. Moreover, for each $k \in Y$

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

let $k' = h(k)$. Then:

$$\arg\min_k \mathbb{E}\big[L(h(y), h(k))|\mathbf{x}\big] = h^{-1}\left(\arg\min_{k'} \mathbb{E}\big[L(y', k')|\mathbf{x}\big]\right), \tag{10}$$

i.e. minimizing the extended loss (9) is equivalent to first minimizing the expected loss (8) with random variable y' and then taking the h^{-1} inverse of the obtained minimizer. The Bayes classification function for (8) is $y'_{1-\alpha}$, the $(1-\alpha)$-quantile of distribution $P(y'|\mathbf{x})$. Since $P(y' = k|\mathbf{x}) = P(y = h^{-1}(k)|\mathbf{x})$ for every $k = 1, \ldots, K$, then we must have $y'_{1-\alpha} = h(y_{1-\alpha})$. According to (10), the Bayes classification function for (9) is $h^{-1}(y'_{1-\alpha}) = h^{-1}(h(y_{1-\alpha})) = y_{1-\alpha}$.

4 Monotone Approximation

We now propose a general method for incorporating the monotonicity constraints into the learning process. The method is based on relabeling objects from the training set in order to remove the inconsistencies and "monotonize" the data. Let us consider the minimization of the empirical risk (2) within the class of *all* monotone functions. This leads to the following problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{n} L(y_i, d_i) \\ \text{subject to} \quad & x_i \succeq x_j \rightarrow d_i \geqslant d_j \qquad i, j = 1, \ldots, n \\ & d_i \in \{1, \ldots, K\} \qquad i = 1, \ldots, n. \end{aligned} \tag{11}$$

The problem has already been considered in statistics [23] for absolute error loss; in logical analysis of data [24] for the binary case, and in the *isotonic separation* method [4] for any loss function. Although variables d_i are integer, the constraint matrix is totally unimodular [25], so that integer constraints can be relaxed and the problem can be solved efficiently via the linear programming; moreover, its dual is the maximum network flow problem [4] and hence can be solved in $O(n^3)$. The analysis of the problem presented in this paper includes only our own results.

Optimal values of variables d_i can be regarded as new class labels for the training objects. Thus, the problem can be interpreted as follows: relabel (reassign) the objects to make the dataset 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 functions. The set of new labels will be called *monotone approximation*. Notice, that (11) may have more than one optimal solution.

Reduction of the problem size. We are able to reduce the size of the monotone approximation problem by removing some of the variables from the optimization process. Let us call an object x_i *consistent*, if for every x_j, we have $y_i \geqslant y_j$ whenever $x_i \succeq x_j$, and $y_i \leqslant y_j$ whenever $x_i \preceq x_j$, $i, j = 1, \ldots, n$. Let us call the dataset D *monotone* if every object x_i, $i = 1, \ldots, n$, is consistent; this is equivalent to the statement that vector $\mathbf{y} = (y_1, \ldots, y_n)$ is monotone.

For each x_i, let us define the *lower* and the *upper class labels*, respectively as:

$$\begin{aligned} l_i &= \min\{y_j : x_j \succeq x_i, j = 1, \ldots, n\} \\ u_i &= \max\{y_j : x_j \preceq x_i, j = 1, \ldots, n\}. \end{aligned} \tag{12}$$

It always holds that $l_i \leqslant y_i \leqslant u_i$ and $l_i = u_i$ if and only if object x_i is consistent. Moreover, for every x_i, x_j such that $x_i \succeq x_j$, we have $l_i \geqslant l_j$ and $u_i \geqslant u_j$. The lower and upper labels can be used to put constraints on some of the variables in the monotone approximation:

Theorem 4. *Let $\hat{d}_i$, $i = 1, \ldots, n$, be any optimal solution to the problem (11) with arbitrary monotone loss function. Then we have $l_i \leqslant \hat{d}_i \leqslant u_i$.*

Proof. Before we prove the theorem, we must first show that the monotone loss function is always *strictly increasing*, i.e. $l_{yk} > l_{y,k+1}$ if $k < y$, and $l_{y,k-1} < l_{yk}$ if $k > y$. We will show only the first inequality; the second one can be shown analogously. From (6) we have that for $k > y$, $l_{y,k+1} - l_{yk} \geqslant l_{y+1,k+1} - l_{y+1,k}$. Repeating this iteratively, we must finally have

$$l_{y,k+1} - l_{yk} \geqslant l_{y+2,k+1} - l_{y+2,k} \geqslant \ldots \geqslant l_{k,k+1} - l_{kk} > 0,$$

where the last inequality comes from $l_{kk} = 0$ and $l_{yk} > 0$ for $y \neq k$.

Assume we have any optimal solution $\hat{d}_i$, $i = 1, \ldots, n$. Let I be a subset of those i for which $\hat{d}_i < l_i$. Similarly, let J be a subset of those i for which $\hat{d}_i > u_i$. Let us introduce the solution $\tilde{d}_i$ such that $\tilde{d}_i = l_i$ for $i \in I$, $\tilde{d}_i = u_i$ for $i \in J$ and $\tilde{d}_i = \hat{d}_i$ otherwise. As the monotone loss function (6) is always strictly increasing and we know that $l_i \leqslant y_i \leqslant u_i$, it follows that $\tilde{d}_i$ has lower objective value that $\hat{d}_i$, if any of the sets I or J is nonempty. Indeed, for every $i \in I$ and every $i \in J$, the label $\tilde{d}_i$ is surely "closer" than $\hat{d}_i$ to the real label y_i. Therefore it is enough to prove that the solution $\tilde{d}_i$ is feasible. Then, I and J must be empty, because otherwise it would contradict the optimality of $\hat{d}_i$.

To prove the feasibility of $\tilde{d}_i$ in the problem (11), we must show that:

$$x_i \succeq x_j \Longrightarrow \tilde{d}_i \geqslant \tilde{d}_j \quad i, j = 1, \ldots, n \tag{13}$$

Notice that for $i \in I$, $\tilde{d}_i > \hat{d}_i$ and for $i \in J$, $\tilde{d}_i < \hat{d}_i$ Choose any $x_i \succeq x_j$. First we consider $i \in I$, then $i \in J$ and finally the case $i \notin I \cup J$:

1. Case $i \in I$. Then if $j \in I$, $\tilde{d}_i = l_i \geqslant l_j = \tilde{d}_j$. If $j \notin J$, $\tilde{d}_i > \hat{d}_i \geqslant \hat{d}_j \geqslant \tilde{d}_j$.
2. Case $i \in J$. Then $\tilde{d}_i = u_i \geqslant u_j \geqslant \tilde{d}_j$.
3. Case $i \notin I \cup J$. Then if $j \in I$, $\tilde{d}_i \geqslant l_i \geqslant l_j = \tilde{d}_j$. If $j \notin I$, $\tilde{d}_i = \hat{d}_i \geqslant \hat{d}_j \geqslant \tilde{d}_j$.

□

Theorem (4) implies that we can remove consistent objects (for which $l_i = u_i$) from the optimization process, since we know a priori that $\hat{d}_i = y_i$ for such objects. This dramatically reduces the size of the problem: for real-life data we have found that in most cases more than $80 - 90\%$ of objects are removed.

Handling non-uniqueness of the solution. The optimal solution of the problem (11) may not be unique (and, in fact, is not unique in most cases). For instance, consider a very simple binary-class example with only two objects $x_2 \succeq x_1$, but $y_2 = 0$ and $y_1 = 1$. Assume 0-1 loss. Then, either x_2 can be relabeled to $y_2 = 1$ or x_1 to $y_1 = 0$, and both solution has the same cost equal to 1. Thus, the particular solution found by the algorithm is accidental, which is unwanted and doubtful from theoretical point of view.

The problem of non-uniqueness of the optimal solution has been consider in [26,13]. We only report the main results here and refer for the proofs to the cited papers. Although the optimal solution may not be unique, there always exist two optimal solutions $\hat{d}_{*i}$, and $\hat{d}_i^*$, $i = 1, \ldots, n$, such that $\hat{d}_{*i}$ is the smallest and $\hat{d}_i^*$ is the greatest optimal solution. In other words, let $\hat{d}_i$, $i = 1, \ldots, n$, be any optimal solution. Then it holds that:

$$\hat{d}_{*i} \leqslant \hat{d}_i \leqslant \hat{d}_i^*$$

for all $i = 1, \ldots, n$. The smallest and the greatest solutions can be found effectively for linear loss function (8). One can show that the smallest solution $\hat{d}_{*i}$ is obtained by solving the monotone approximation with the value α increased by a sufficiently small amount ϵ. Similarly, the greatest solution $\hat{d}_i^*$ is obtained by decreasing α by ϵ. One can show that the value $\epsilon \leqslant n^{-2}$ is sufficiently small.

Finally, we notice that a modification of the results obtained in [13] will lead to the conclusion that the monotone approximation is not sensitive to the choice of the scale function in extended linear loss (9). In other words, monotone approximation with extended linear loss is the same for every function $h(k)$; in particular, it is the same as for ordinary linear loss function. This shows an unusual property of the linear loss function: being independent of the metric scale. Although linear loss seemingly imposes a distance measure between the class labels, an arbitrary change of the distance measure leads to the same result.

5 Experimental Results

The monotone approximation can be used to improve the training set by removing inconsistencies, contradicting the domain knowledge. This is done by relabeling inconsistent objects in a way which is the safest in the probabilistic meaning. It corresponds to the domain knowledge-based error correction (or outliers detection) method. Our method can be used as a preprocessing tool in combination with any ordinal classification algorithm, by applying it to the original data, "monotonizing" all training examples and passing them to the learning algorithm, in order to improve the accuracy and comprehensibility of the classifier. We thoroughly investigate the method on the artificial data.

We propose the following algorithm for generating artificial data with monotonically constrained probability distribution. We assume for simplicity that $K = 2$ and $Y = \{0, 1\}$. Objects $x = (x^1, \ldots, x^m) \in \mathbb{R}^m$ are generated uniformly on a cube $[0, 1]^m$, i.e., $x \sim U^m(0, 1)$. The most important characteristic of the

data from the prediction point of view is the underlying "target" function, modeling the Bayes classification function $f^*(x)$. We assume that $f^*(x) = 1_{h(x)>0}$, where $h(x)$ is a real-valued function, described below. The noise (non-zero Bayes risk) is introduced to the model in the following way: the observed label y equals $f^*(x)$ with probability $1-\gamma$ and equals $1-f^*(x)$ with probability γ. In other words, the noise causes random errors in label assignment with probability γ. One can easily show that the Bayes risk $R^* = \gamma$. The conditional distribution made this way is monotonically constrained as long as the function $h(x)$ is monotone.

The target function $h(x)$ has the following form:

$$h(x) = \sum_{t=1}^{T} a_t r_t(x) - \theta, \tag{14}$$

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

$$r_t(x) = \prod_{s=1}^{m_s} 1_{x^{j_s} \geqslant b^s} \qquad r_t(x) = -\prod_{s=1}^{m_s} 1_{x^{j_s} \leqslant b^s},$$

where $j_s \in \{1, \ldots, m\}$ and $b^s \in [0,1]$. One can show that $h(x)$ is a monotone function. Moreover, every monotone function can be approximated arbitrarily close (with respect to an L_p norm) by functions of the form (14). Notice, that $r_t(x)$ have the form of a hyperrectangle.

All the parameters of the model, apart from T, R^* and θ, are chosen at random: we set $a_t, b_s \sim U(0,1)$; each j_s is 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 $50 \times m$ in the experiment. The threshold θ is chosen so that the prior probabilities of both classes are equal: $P(y=1) = P(y=0)$. The parameters to be changed are: sample size n, dimensionality m and Bayes risk R^*. We choose $n = 1000$, $m \in \{4, 6, 8, 10\}$ and $R^* \in \{0.1, 0.2, 0.3, 0.4\}$. For each combination of these parameters we generate 20 models of the form (14). For each model, we train the method on 10 separate sets of size 1000 and test on 10 separate sets of the same size.

The problem is binary, so we can use any binary classification algorithm minimizing 0-1 loss, since all symmetric loss functions (0-1, absolute error, squared error, etc.) are equivalent for binary classification and are monotone loss functions, as described in Section 2. We choose C4.5 [27], AdaBoost [28] with C4.5 as a base learner (20 iterations), logistic regression and RankBoost [11] with stump as a base learner (100 iterations). Notice that RankBoost is designed to minimize the rank loss, but can be easily adapted to deal with ordinal classification for any loss function, as described in [20]. We use 0-1 loss as a measure of accuracy.

Each classifier was learned in two copies, either with or without monotone approximation (in monotone approximation, we always choose the greatest solution $\hat{d}_i^*$ without loss of generality). In other words, one copy of the classifier was learned on the original data, while another copy – on the monotonized data,

Table 1. Results on the artificial data. For each classifier, two values are shown: the first is the testing error (in %) when the classifier is trained on the original data (ORIG), while the second concerns testing error (in %) with training on the monotonized data (MON), i.e. with monotone approximation. The significantly higher result is marked with bold.

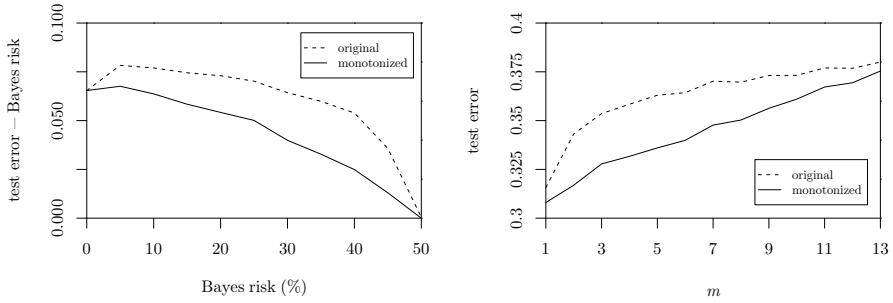
DATASET		C4.5		ADABOOST		LOGISTIC		RANKBOOST	
R^*	m	ORIG	MON	ORIG	MON	ORIG	MON	ORIG	MON
10%	4	20.7	**19.4**	20.4	**16.7**	14.2	14.1	16.8	**14.7**
	6	25.1	**24.7**	22.5	**19.6**	14.1	14.0	17.5	**16.0**
	8	28.7	**28.4**	23.4	**22.1**	14.5	14.4	18.1	**17.3**
	10	31.6	31.6	24.5	**23.8**	14.7	14.6	18.6	**18.2**
20%	4	29.7	**27.8**	29.7	**26.0**	23.4	**23.2**	26.5	**24.1**
	6	33.6	**32.5**	33.1	**29.3**	23.8	23.7	27.2	**25.3**
	8	36.0	**35.3**	34.8	**32.5**	24.1	**23.9**	27.6	**26.2**
	10	37.2	37.2	35.8	**34.5**	24.3	**24.1**	28.0	**27.2**
30%	4	38.6	**36.0**	38.3	**34.9**	32.9	32.7	35.9	**33.3**
	6	41.2	**39.2**	40.5	**37.7**	33.3	**33.0**	36.5	**33.9**
	8	42.0	**41.2**	41.3	**39.9**	33.9	**33.4**	37.0	**34.9**
	10	43.5	**42.9**	42.6	**41.6**	34.1	**33.6**	37.3	**35.9**
40%	4	46.5	**43.5**	46.4	**42.9**	42.4	**41.7**	44.7	**42.0**
	6	47.5	**45.2**	47.4	**44.3**	43.0	**42.0**	45.3	**42.4**
	8	47.9	**46.1**	47.8	**45.4**	43.4	**42.1**	45.5	**42.9**
	10	47.8	**46.9**	47.8	**46.6**	43.7	**42.7**	45.6	**43.8**

with inconsistencies removed. Each classifiers was run on 20 models, For each model, the average accuracy (0-1 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 implementation of RankBoost was taken from [20]. Implementations of other algorithms were taken from the Weka software [29]. The results of the experiment are shown in Table 1. They unquestionably show that removing inconsistencies can only improve the accuracy. The strength of the improvement, however, depends on the learning algorithm and properties of the dataset. The highest improvement is gained by the most complex classifiers which can closely fit to the data, AdaBoost and RankBoost. Removing the inconsistencies may help to push the learning process in the right direction and avoid overfitting. The smallest improvement is gained by the logistic regression; this is caused by the fact that linear classifiers are more stable and "weaker" (in the sense that they cannot push the training error down to zero) than tree or stump ensembles. This result is consistent with [30], where it is proven that a linear fit can determine monotonicity direction for a wide class of probability distributions.

It also follows from the experiment, that the amount of the improvement increases with increasing Bayes risk, mostly due the fact that monotone approximation works as error correction based on the domain knowledge about the monotonicity. When the level of noise in the data is high, using the knowledge

Fig. 2. Left chart: test error of RankBoost decreased by Bayes risk as a function of Bayes risk for $m = 6$. Right chart: test error of RankBoost as a function of m for Bayes risk 0.3.



about the probability distribution (monotonicity constraints) to relabel the objects towards Bayes classification function, increases the quality of the dataset. Notice, that the improvement decreases with the number of attributes m. This can be explained by observing that the dominance relation becomes sparse in high dimensional space and only few objects are then relabeled. The relationships mentioned above are shown in greater details in Figure 2 for RankBoost, with and without monotonizing the data. The left chart shows the loss in accuracy compared to the Bayes classification function (test error minus Bayes risk) as a function of Bayes risk R^*. The improvement on monotone approximation increases with the R^* up to $R^* = 0.45$; for $R^* = 0.5$, surprisingly, both methods detect that there is no dependency between y and x and equate their risks with R^*. The right chart shows the influence of the number of attributes m on the testing error. At first, the improvement on monotone approximation increases with m, but later, due to the sparseness of the dominance relation it starts to decrease.

Another issue related to data monotonization is that the classifiers tend to be more comprehensible when the domain knowledge is incorporated in the learning process. We leave this problem for further research.

6 Summary

We presented a statistical theory for the problem of ordinal classification in the presence of monotonicity constraints. Although the constraints follow from domain knowledge and often appear in the real data, they are rarely taken into account and were not deeply considered from the theoretical point of view in machine learning. We considered the problem in its most general formulation, when the knowledge about the objects is expressed only through the dominance relation $\succeq$. We introduced a probabilistic model for monotone classification, based on the concept of stochastic dominance, and investigated the possible loss functions in this setting.

We also proposed a general method for incorporating the monotonicity constraints into the learning process, called monotone approximation. The method is based on relabeling the training objects in order to remove inconsistencies and monotonize the training data, and can be used as a preprocessing tool in combination with any algorithm for ordinal classification. Our approach was thoroughly verified on the artificial data and proved to increase the accuracy of many classifiers. The highest gain in accuracy is achieved when the classifier is complex and the level of noise (Bayes risk) is high.

References

1. Greco, S., Matarazzo, B., Słowiński, R.: Rough set approach to customer satisfaction analysis. Volume 4259 of LNCS., Springer (2006) 284–295
2. Koop, G.: Analysis of Economic Data. John Wiley and Sons (2000)
3. Greco, S., Matarazzo, B., Słowiński, R.: A new rough set approach to evaluation of bankruptcy risk. In Zopounidis, C., ed.: Operational Tools in the Management of Financial Risks. Kluwer Academic Publishers, Dordrecht (1998) 121–136
4. Chandrasekaran, R., Ryu, Y.U., Jacob, V.S., Hong, S.: Isotonic separation. INFORMS Journal on Computing **17**(4) (2005) 462–474
5. Potharst, R., Feelders, A.J.: Classification trees for problems with monotonicity constraints. SIGKDD Explorations **4**(1) (2002) 1–10
6. Cao-Van, K., De Baets, B.: An instance-based algorithm for learning rankings. In: Proceedings of Benelearn. (2004) 15–21
7. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research **129** (2001) 1–47
8. Agarwal, S.: Ranking on graph data. In: ICML '06. (2006) 25–32
9. Herbrich, R., Graepel, T., Obermayer, K.: Regression models for ordinal data: A machine learning approach. Technical report, Technical University of Berlin (1999)
10. Shashua, A., Levin, A.: Ranking with large margin principle: Two approaches. Advances in Neural Information Processing System **15** (2003)
11. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. J. of Machine Learning Research **4** (2003) 933–969
12. Lin, H.T., Li, L.: Ordinal regression by extended binary classifications. Advances in Neural Information Processing Systems **19** (2007) 865–872
13. Dembczyński, K., Greco, S., Kotłowski, W., Słowiński, R.: Statistical model for rough set approach to multicriteria classification. In: PKDD '07. Volume 4702 of LNCS., Springer (2007) 164–175
14. Greco, S., Matarazzo, B., Słowiński, R.: Rough set based decision support. In: Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. Springer (2005) 475–527
15. Ben-David, A.: Monotonicity maintenance in information-theoretic machine learning algorithms. Machine Learning **19**(1) (1995) 29–43
16. Giove, S., Greco, S., Matarazzo, B., Słowiński, R.: Variable consistency monotonic decision trees. Volume 2475 of LNAI., Springer (2002) 247–254
17. Cao-Van, K., De Baets, B.: Growing decision trees in an ordinal setting. International Journal of Intelligent Systems **18** (2003) 733–750
18. Sill, J.: Monotonic networks. In: Advances in Neural Information Processing Systems. Volume 10., Denver, USA, The MIT Press (1998) 661–667

19. BenDavid, A., Sterling, L., Pao, Y.H.: Learning and classification of monotonic ordinal concepts. Computational Intelligence **5**(1) (1989) 45–49
20. Lin, H.T., Li, L.: Large-margin thresholded ensembles for ordinal regression: Theory and practice. Lecture Notes in Artificial Intelligence **4264** (2006) 319–333
21. Levy, H.: Stochastic Dominance. Kluwer Academic Publishers (1998)
22. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. Springer (1993)
23. Dykstra, R., Hewett, J., Robertson, T.: Nonparametric, isotonic discriminant procedures. Biometrica **86**(2) (1999) 429–438
24. Boros, E., Hammer, P.L., Hooker, J.N.: Boolean regression. Annals of Operations Research **58**(3) (1995)
25. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization. Dover (1998)
26. Dembczyński, K., Greco, S., Kotłowski, W., Słowiński, R.: Optimized generalized decision in dominance-based rough set approach. In: RSKT. Volume 4481 of Lecture Notes in Computer Science. (2007) 118–125
27. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
28. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences **55**(1) (1997) 119–139
29. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco (2005)
30. Magdon-Ismail, M., Sill, J.: A linear fit gets the correct monotonicity directions. Machine Learning **70**(1) (2008) 21–43