Ordinal Classification Using Partially Ordered Feature Sets

I. D. Papay[1], V. V. Strijov[1,1]

Abstract

This paper presents a method to solve an ordinal classification problem where the objects are described with a set of partially ordered features. To construct an ordinal classification model we aggregate partial orders of features by weighting the incidence matrices of partial order graphs. To describe set of possible solutions we use partial order cones corresponding to the incidence matrices and find a solution of the classification problem as the projection of a vector of response variables to a superposition of the cones. We propose a method of optimum parameter estimation for the ordinal classification model. The method is applied to the Football Player List categorization problem.

Keywords: ordinal classification, alternatives ranking, preference learning, partial order, monotonic constraints

1. Introduction

We consider an ordinal classification problem using partially ordered feature sets. This problem generalizes the well-known ordinal classification problem with monotonic constraints [1, 2, 3]. The objective is to construct a monotonic function, further calling a classification model, mapping an object set to a class label set such that the class label set is strictly totally ordered. The requirement of the classification model monotonicity is a form of prior knowledge of a relation between a response and predictor variables [4, 5, 6]. The motivation for this type of problem statement arises in areas of support decision theory [7, 8, 9], information retrieval [10, 11, 12] and preference learning [13, 14].

[*] Corresponding author

Email addresses: papayid1503@gmail.com (I. D. Papay), strijov@ccas.ru (V. V. Strijov)

The task is set as follows. There are a number of experts ranking some partially ordered set of objects. Our goal is to teach the model to predict the rank [15] of an individual object for any other expert [16]. At the same time, all that is known about an object is a finite-dimensional vector of its features.

In this paper we consider a general case of classification model monotonicity requirement: the set of values for each feature is a partially ordered set, that is, a partial order relation is defined over the feature values set [17]. The goal is to construct a classification model satisfying monotonicity of relation between partially ordered features and a response variable. To construct a classification model defined over the Cartesian product of the partially ordered sets, we use incidence matrices of the graphs corresponding to the partial orders. We introduce a partial order cone corresponding to the incidence matrix [18, 19]. We regard a superposition of the partial order cones as the set of possible solutions of the ordinal classification problem. In the paper we prove that the constructed cone superposition describes a broad class of monotonic transformations.

As a solution of the ordinal classification problem we find a point in the superposition of the cones, maximizing its correlation with the expert-given target vector. We propose a method of parametrization of the cone superposition to find optimal parameters of the classification model. Using this parametrization, the method of parameter estimation consists of the two independent stages. In the first stage we construct a matrix of pairwise dominance of the objects. This type of matrices is used in alternatives ranking problems [20, 21]. In the paper [22] the authors solve a problem of the linear order construction using the pairwise dominance matrix. In the second stage we solve an ordinal classification problem wherein using the columns of the dominance matrix as the new predictor variables.

In this paper we investigate properties of the partial order cone generators. We show that the generator vectors are equal to the columns of the incidence matrix corresponding to the partial order. This novel fact proves that the proposed method of ordinal classification finds an optimal function in a sufficiently broad class of monotonic transformations.

As methods competing with the proposed method, that solves our problem, we assume decision oblique trees and isotonic regression.

Decision tree can be explained a series of nested if-then-else statements. Each non-leaf node has a predicate associated, testing an attribute of data. Terminal node denotes class, or category. To classify a data ,we have to traverse down the tree by starting from root node ,testing predicates(test attribute) and taking branches labelled with corresponding value.

Isotonic regression is a technique used to fit a non-decreasing function to a set of data points. We can think of isotonic regression as a generalization of linear regression where the function is constrained to be non-decreasing.

The proposed method is compared with oblique decision tree [23] and isotonic regression [24] on synthetic and real data. As a real data we consider a problem of the Football Player List categorization, also described in. The goal is to rank FIFA players by their world reputation ranks basing on their characteristics. The following categories are considered: least popular, rarely heard, normally heard, popular, extremely popular. This categorization is monotonic. The partially ordered features (e.g., mobility, strength) are given by the experts. The proposed method shows better results in comparison with the alternative approaches.

Table 1: Comparative analysis of basic solution to my problem.

Solution	Strengths	Weakness
Partial Orders	no contradictions in the partial order	vulnerable to noise
Isotonic regression	working fast	inaccurately predictions
Oblique decision trees	learns non-linearity dependence	vulnerable to retraining
Stochastic Partial Orders	no vulnerability to noise	demand higher dimension of features

2. Ordinal classification with monotonicity constraints

Partially ordered feature set. Let $\mathfrak{D}$ be a sample consisting of the pairs

$$\mathfrak{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m},$$

where $\mathbf{x}_i = [x_{i1}, ..., x_{ij}, ..., x_{in}]^\mathsf{T}$ is an object to be classified, y_i is a class label. An object $\mathbf{x}_i$ is an n-dimensional vector, each component of which belongs to the corresponding partially ordered set X_j. In other words, an element j of the vector $\mathbf{x}_i$ (which is also reffered to as value of the feature j for the object $\mathbf{x}_i$) belongs to the set X_j with the given partial order relation $\succeq$ with the following properties:

- reflexivity, $\forall a \in X \ (a \succeq a)$,
- antisymmetry, $\forall a, b \in X$, $(a \succeq b) \wedge (b \succeq a) \Rightarrow (a = c)$,
- transitivity, $\forall a, b, c \in X \ (a \succeq b) \wedge (b \succeq c) \Rightarrow (a \succeq c)$.

Thereby the object set X is a Cartesian product of the partially ordered sets $X_1, ..., X_n$,

$$X = X_1 \times X_2 \times \cdots \times X_n,$$

the sets $X_1, ..., X_n$ are the feature values sets.

Every partial order $\succeq$, defined on a set X_j, described with a binary function $z_j(\mathbf{x}_i, \mathbf{x}_k)$ such that

$$z_j(\mathbf{x}_i, \mathbf{x}_k) = \begin{cases} 1, & \text{if} \quad x_{ij} \succeq x_{kj}, \\ 0, & \text{if} \quad x_{ij} \not\succeq x_{kj}, \end{cases}$$

where the function $z_j(\cdot, \cdot)$ satisfies the conditions of reflexivity, antisymmetry and transitivity.

Let us define a partial order matrix $\mathbf{Z}_j$ for the sample $\mathfrak{D}$ and every set X_j such that the matrix describes binary relation between each pair of the sample elements,

$$\mathbf{Z}_j(i, k) = z_j(\mathbf{x}_i, \mathbf{x}_k),$$

where j is a feature index, and i, k are the object indices. Note that for the matrix $\mathbf{Z}_j$, as well as for the function z_j, elements $\mathbf{Z}_j(i, k)$ and $\mathbf{Z}_j(k, i)$ do not depend on each other. For example, the objects i, k can be incomparable with each other for the feature j. In this case for the matrix $\mathbf{Z}_j$ the following equalities hold:

$$\mathbf{Z}_j(i, k) = \mathbf{Z}_j(k, i) = 0.$$

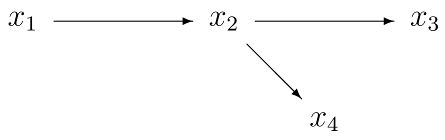
On the other hand, the objects i, k can be equal for the feature j. In this case, the following equalities hold:

$$\mathbf{Z}_j(i,k) = \mathbf{Z}_j(k,i) = 1.$$

Example 1. *The partially ordered set X_j consists of four objects with the following binary relation,*

$$X_j = \{x_1, x_2, x_3, x_4 | \; x_1 \succeq x_2, x_2 \succeq x_3, x_2 \succeq x_4\}.$$

The partial order on the objects x_1, x_2, x_3, x_4 is described by a partial order graph,



This graph corresponds to a matrix $\mathbf{Z}_j$:

$$\mathbf{Z}_j = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The class label set

$$Y = \{l_1, l_2, ..., l_K\}, \quad y_i \in Y \tag{1}$$

is a finite set with the specified strict linear order relation, $l_1 \prec l_2, ..., \prec l_K$, where K is a number of the class labels. Similarly to the matrices $\mathbf{Z}_1, ..., \mathbf{Z}_n$, construct a partial order matrix $\mathbf{Z}_0$ corresponding to the vector of class labels $\mathbf{y} = [y_1, ..., y_m]^\mathsf{T}$ of the sample $\mathfrak{D}$:

$$\mathbf{Z}_0(i,k) = \begin{cases} 1, & \text{if } y_i \succeq y_k, \\ 0, & \text{otherwise.} \end{cases}$$

Partial order cone. Introduce a polyhedral cone $\mathcal{X}$, corresponding to the partially ordered set of feature values.

Definition 1. *A polyhedral cone in $\mathbb{R}^m$ is a set $\mathcal{X}$ such that*

$$\mathcal{X} = \{\boldsymbol{\chi} \mid \mathbf{A}\boldsymbol{\chi} \leq \mathbf{0}, \; \boldsymbol{\chi} \in \mathbb{R}^m\},$$

fore some matrix $\mathbf{A}$.

By matrix A in our article, from here on, we consider the matrix

$$\mathbf{A}_m = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 \\ 0 & 0 & 0 & \dots & 0 & -1 \end{pmatrix},$$

For each set X_j define a cone $\mathcal{X}_j$ in a space $\mathbb{R}^m_+$:

$$\mathcal{X}_j = \{\boldsymbol{\chi}_j \in \mathbb{R}^m_+ | \quad x_{ij} \succeq x_{kj} \to \chi_{ij} \geq \chi_{kj} \quad \forall i, k = 1, ..., m\}, \tag{2}$$

where x_{ij} and x_{kj} are values of the feature j on the sample elements i and k, respectively. For the cone $\mathcal{X}$ (index j is omitted for the reasons of convenience) the following theorem holds.

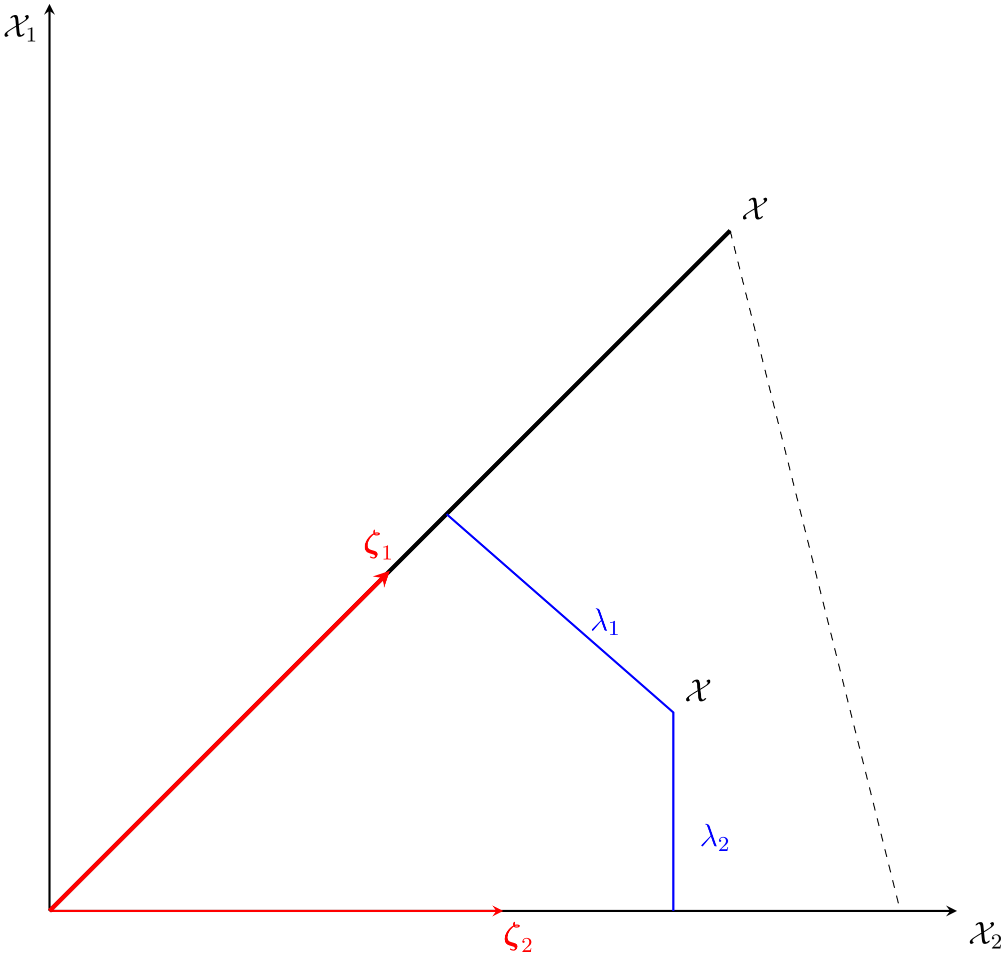
Theorem 1. *A vector $\boldsymbol{\chi}$ belonging to the cone $\mathcal{X}$ can be represented as the nonnegative combination of cone generators,*

$$\boldsymbol{\chi} = \sum_{k=1}^{m} \lambda_k \boldsymbol{\zeta}_k, \quad \lambda_k \geq 0,$$

where $\boldsymbol{\zeta}_k$ is a generatrix of the cone $\mathcal{X}$,

$$\boldsymbol{\zeta}_k(i) = \begin{cases} 1, & \textit{if} \quad x_i \succeq x_k, \\ 0, & \textit{if} \quad x_i \not\succeq x_k, \end{cases}$$

and this decomposition is unique.



The figure illustrates vector $\boldsymbol{\chi}$ decomposition on the generators $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2$ of the cone $\mathcal{X}$. The vector $\boldsymbol{\chi} \in \mathcal{X}$ is a nonnegative combination of the generators $\boldsymbol{\zeta}_1 = [1; 1]$ and $\boldsymbol{\zeta}_2 = [1; 0]$ with the coefficients $\lambda_1, \lambda_2 \geq 0$.

Theorem 1 establishes the following connection between the cone $\mathcal{X}_j$ and the partial order matrix $\mathbf{Z}_j$: the generatrix $\boldsymbol{\zeta}_{jk}$ of the cone $\mathcal{X}_j$ is a column k of the matrix $\mathbf{Z}_j$.

Ordinal classification with monotonicity constraints. In this section we formulate the problem of ordinal classification with monotonicity constraints; the constraints monotonicity means satisfying the given object binary relations.

As it was mentioned before (1), the set of class labels $Y = \{l_1, l_2, ..., l_K\}$ is a finite set with the specified strict linear order relation, $l_1 \prec l_2, ..., \prec l_K$. The goal is to construct a monotonic function $f : X \to Y$ minimizing a given loss function,

$$S(\mathfrak{D}) = \sum_{i=1}^{m} s(f(\mathbf{x}_i), y_i) \to \min, \tag{3}$$

where $s(f(\mathbf{x}_i), y_i)$ is a loss value of the function f on the object $\mathbf{x}_i$. We construct a monotonic function f as the following superposition,

$$f(\mathbf{x}) = \phi(u(\mathbf{x})), \tag{4}$$

where $u : X \to \mathbb{R}$ is a utility function *mapping object set X to the real axis $\mathbb{R}$, and $\phi : \mathbb{R} \to Y$ is a* decision rule *dividing the real axis into the sets of correspondence to the class labels $l_1, ..., l_K$.*

Using conic description of the partially ordered sets (2), find a solution of the problem (4) as follows. Find a vector $\hat{\mathbf{y}} \in \mathbb{R}^m$ that belongs to the superposition of the cones $\mathcal{X}_1, ..., \mathcal{X}_n$ and minimizes the loss function (3),

$$S(\mathfrak{D}) = \sum_{i=1}^{m} s(\phi(\hat{y}_i), y_i) \to \min, \quad \hat{\mathbf{y}} \in \sum_{i=1}^{n} \mathcal{X}_i,$$

where $\sum\limits_{i=1}^{n} \mathcal{X}_i$ denotes a Minkowski sum of the sets $\mathcal{X}_1, ..., \mathcal{X}_n$:

$$\sum_{i=1}^{n} \mathcal{X}_i = \{\boldsymbol{\chi}_1 + ... + \boldsymbol{\chi}_n \mid \boldsymbol{\chi}_1 \in \mathcal{X}_1, ..., \boldsymbol{\chi}_n \in \mathcal{X}_n\}.$$

Note that though a Minkoswski sum of the cones $\mathcal{X}_1, ..., \mathcal{X}_n$ is a special case of a superposition of the sets $\mathcal{X}_1, ..., \mathcal{X}_n$, it still defines a sufficiently broad class of monotonic transformations. The concept of Minkowski sum generalizes linear model *idea in the case of ordered feature sets.*

As long as target vector $\hat{\mathbf{y}}$ must belong to the Minkowski sum of the cones, $\hat{\mathbf{y}} \in \sum\limits_{i=1}^{n} \mathcal{X}_i$, from Theorem 1 it follows that the vector $\hat{\mathbf{y}}$ decomposes in a linear combination of generators of the cones $\mathcal{X}_1, ..., \mathcal{X}_n$,

$$\hat{\mathbf{y}} = \sum_{j=1}^{n} \sum_{k=1}^{m} \lambda_{jk} \boldsymbol{\zeta}_{jk}, \quad \lambda_{jk} \in \mathbb{R}_+,$$

where $\boldsymbol{\zeta}_{jk}$ is a k -generatrix of the cone $\mathcal{X}_j$. The vector $\boldsymbol{\zeta}_{jk}$ is also a k column of the partial order matrix $\mathbf{Z}_j$. Therefore the utility function u from the formula (4) for any object $\mathbf{x} \in X$ is as follows,

$$u(\mathbf{x}) = \sum_{j=1}^{n} w_j \sum_{k=1}^{m} \lambda_{jk} z_j(\mathbf{x}, \mathbf{x}_k), \tag{5}$$

where the parameters w_j are feature weights.

Note that the problem statement (5) is equivalent to a canonical problem of ordinal classification with monotonic constraints. The problem is to find a utility function u as a linear combination of monotonic functions $u_1, ..., u_n$:

$$u(\mathbf{x}) = \sum_{j=1}^{n} w_j u_j(x_j), \tag{6}$$

where every function u_j is monotonic due to its argument $x \in X_j$,

$$x_2 \succeq x_1 \quad \to \quad u_j(x_2) \geq u_j(x_1). \tag{7}$$

For the further considerations make some restrictions over a class of functions $u(\mathbf{x})$. Rewrite a formula (5) as follows,

$$u(\mathbf{x}) = \sum_{k=1}^{m} \lambda_k \sum_{j=1}^{n} w_j z_j(\mathbf{x}, \mathbf{x}_k) = \sum_{k=1}^{m} \lambda_k \Psi(\mathbf{x}, \mathbf{x}_k). \tag{8}$$

The model (8) has equal parameter values λ_k for all the features j. In this case the parameters λ_k are the object weights for the sample $\mathfrak{D}$. The function $\Psi(\mathbf{x}_i, \mathbf{x}_k)$ describes partial order relation between the objects $\mathbf{x}_i$ and $\mathbf{x}_k$ and takes values in the unit interval $[0; 1]$,

$$\Psi(\mathbf{x}_i, \mathbf{x}_k) \in [0, 1].$$

The function $\Psi(\mathbf{x}_i, \mathbf{x}_k)$ can be interpreted as the pairwise dominance degree between the objects $\mathbf{x}_i$ and $\mathbf{x}_k$, $\mathbf{x}_i \succeq \mathbf{x}_k$.

Loss function minimization for the optimal parameters search. To solve the problem of ordinal classification with monotonic constraints (3), let us to formulate a problem of finding the optimal parameters of the model (8). According to formula (3), the optimal parameters $\hat{\mathbf{w}}, \hat{\boldsymbol{\lambda}}$ minimize the loss function

$$(\hat{\mathbf{w}}, \hat{\boldsymbol{\lambda}}) = \arg\min_{\mathbf{w}, \boldsymbol{\lambda}} \left(S(\mathfrak{D}|\mathbf{w}, \boldsymbol{\lambda})\right) = \arg\min_{\mathbf{w}, \boldsymbol{\lambda}} \left(\sum_{i=1}^{m} s(f_{\mathbf{w},\boldsymbol{\lambda}}(\mathbf{x}_i), y_i) \right),$$

where $s(f_{\mathbf{w},\boldsymbol{\lambda}}(\mathbf{x}_i), y_i)$ — is the value of loss on an object $\mathbf{x}_i$, and a monotonic function

$$f_{\mathbf{w},\boldsymbol{\lambda}}(\mathbf{x}_i) = \phi \left(\sum_{k=1}^{m} \lambda_k \Psi(\mathbf{x}_i, \mathbf{x}_k) \right) \tag{9}$$

is defined by the expressions (4) and (8). The explicit form of the loss function S, of the decision rule ϕ and of the binary relation function Ψ will be considered in the next section.

3. Classification methods

3.1. Isothonic regression

More formally suppose that one collects n sample points $y_1, \ldots, y_n$ and has weights $w_1, \ldots, w_n$ associated with each data point.

When no a priori weights are given, we can set $w_i = 1$ for all $i \in \overline{1, n}$.

Then, the isotonic regression problem can be formulated as follows:

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} w_i (x_i - y_i)^2 \tag{10}$$

$$s.t. \quad x_1 \leq x_2 \leq \ldots \leq x_n$$

The formulation of the isotonic regression problem given in (10) is a quadratic program (QP) with linear constraints. Using a matrix notation, introducing the matrix

$$A = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & 1 & -1 \end{pmatrix} = \begin{pmatrix} a_1^\top \\ \vdots \\ a_{n-1}^\top \end{pmatrix} \in \mathbb{R}^{n-1 \times n} \quad ,$$

the isotonic regression problem can be formulated as follows:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{n} w_i (x_i - y_i)^2 \\ s.t. \quad Ax \leq 0 \end{aligned} \tag{11}$$

In what follows we will write $W = (w_1, \ldots, w_n)$, assuming that $w_i > 0$ for all $i \in 1, n$, so $W^{-1} = (1/w_1, \ldots, 1/w_n)$. Hence, we can rewrite the isotonic regression problem as

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} (x - y)^\top W (x - y) \tag{12}$$

$$s.t. \quad Ax \leq 0 \quad . \tag{13}$$

Algorithm 1 PAVA

```
    $y \in \mathbb{R}^n, w \in \mathbb{R}^n_+ r \leftarrow y$ $W \leftarrow y$ $J = [\{1\}, \ldots, \{n\}]$ {lists of blocks} $i = 1$ (index of list
    start at 0 here) while $i < n$ do
    if $r_i < r_{i-1}$ then
        {Find adjacent violators and merge groups} $r_i \leftarrow \frac{W_i r_i + W_{i-1} r_{i-1}}{W_i + W_{i-1}}$ $W_i \leftarrow W_i + W_{i-1}$
        $J_i \leftarrow J_i \cup J_{i-1}$ Remove $r_{i-1}$, $W_{i-1}$ and $J_{i-1}$ from the listsif $i > 1$ then
        $i \leftarrow i - 1$
      end if
    else
      $i \leftarrow i + 1$
    end if
  end while
  for $i = 1$ to len($J$) do
    $r_{J_i} \leftarrow \bar{y}_{J_i} \mathbf{1}_{J_i}$ {Set the block to the average value}
  end for
  return $r, J$
```

Hence, the isotonic regression problem can be formulated as a convex quadratic program with linear constraints.

The intuition behind the PAVA algorithm is to merge adjacent blocks of constant values in the primal vector, as the targeted solution is simply the average of the observe signal over each block. Hence, the algorithm aims at creating the blocks of constant values in the primal vector. The dual variables can be inferred from the solution from the previous lemma.

3.2. Decision oblique trees

Univariate and multivariate decision tree. Multivariate decision trees differ from univariate decision trees in the way they test the attributes. Univariate decision trees test single attribute at internal node. Multivariate decision tree several attribute participate in single node split test. The limitation to one attribute reduces the ability of expressing concepts, due to its disability in three forms. Splits could only be orthogonal to axes, subtrees may be replicated and fragmentation.

Univariate test using feature x_i *can only split a space with a boundary that is orthogonal to the* x_i *axis. This results in larger trees and poor generalization. The multivariate decision tree-constructing algorithm selects not the best attribute but the best linear combination of the attributes:* $\sum_{i=1}^{f} w_i x_i > 0$*.* w_i *are the weights associated with each feature* x_i *and* w_0 *is the threshold to be determined from the data. Multivariate decision trees differ from univariate trees as the symbolic features are converted into numeric features. And all splits are binary, final weighted sum is numeric.*

We examine decision trees that test a linear combination of the attributes at each internal node. More precisely, let an example take the form $X = x1; x2; \ldots xd; C_j$ *where* C_j *is a class label and the* x_i *are real-valued attributes. The test at each node will then have the form:*

$$\sum_{i=1}^{d} a_i x_i + a_{d+1} > 0. \tag{14}$$

Here $a_1; \ldots; a_{d+1}$ *are real-valued coefficients. Because these tests are equivalent to hyperplanes at an oblique orientation to the axes, we call this class of decision trees oblique decision trees.*

Tree induction algorithms create decision trees that take into account only a single attribute at a time. For each node of the decision tree an attribute is selected from the feature space of the dataset which brings maximum information gain by splitting the data on its distinct values. The information gain is calculated as the difference between the entropy of the initial dataset and the sum of the entropies of each of the subsets after the split. Algorithm selects at each node the split on the attribute which gives the biggest gain Such trees make splits parallel to the axis in the feature space of the dataset.

On the other hand, oblique decision trees split the feature space by considering combinations of the attribute values, be them linear or otherwise. Though these methods can find the optimal linear discriminants for specific goodness measures, the size of the linear program grows very fast with the number of instances and the number of attributes.

Algorithm 2 CART

```
{To induce a split at node T of the decision tree:} {Normalize values for all d attributes.} $L = 0$while TRUE do
  $L = L + 1$ {Let the current split $s_L$ be $u \le c$, where $u = \sum_{i=1}^{d} a_i x_i$}
  for $i = 1$ to $d$ do
    for $\gamma = -0.25$ to $0.25$ do
      {Search for the $\sigma$ that maximizes the goodness of the split $u - \sigma(a_i + \gamma) \le c$}
    end for{Let $\sigma^*, \gamma^*$ be the settings that result in highest goodness in these 3 searches}
    $a_i = a_i - \sigma^*$
    $c = c - \sigma^* \gamma^*$
  end for{Perturb c to maximize the goodness of $s_L$, keeping $a_1, \dots, a_d$ constant} {If —goodness($s_L$) - goodness($s_{L-1}$)—$\le \epsilon$ exit while loop}
end while{Eliminate irrelevant attributes in $\{a_1; \dots; a_d\}$ using backward elimination}
{Convert $s_L$ to a split on the un-normalized attributes} {Return the better of $s_L$ and the best axis-parallel split as the split for $T$}
```

3.3. An ordinal classification algorithm

To estimate parameters of the model (9) we propose a two-stage algorithm. In the first stage estimate the partial order matrix $\mathbf{\Psi}$, where the elements of the matrix $\mathbf{\Psi}$ equal to the values of function Ψ on the objects of the sample $\mathfrak{D}$,

$$\mathbf{\Psi}(i,k) = \Psi(\mathbf{x}_i, \mathbf{x}_k).$$

In the second stage estimate parameters λ_k of a linear combination $\sum\limits_{k=1}^{m} \lambda_k \Psi(\mathbf{x}_i, \mathbf{x}_k)$ from (8).

Partial order matrix estimation. Estimate matrix $\mathbf{\Psi}$ of the pairwise objects dominance using matrices $\mathbf{Z}_1, ..., \mathbf{Z}_n$ corresponding to the partially ordered feature sets. Let every element of a matrix $\mathbf{\Psi}$ be a linear combination of elements of the matrices $\mathbf{Z}_1, ..., \mathbf{Z}_n$, according to (6):

$$\mathbf{\Psi}(i,k) = \sum_{j=1}^{n} w_j \mathbf{Z}_j(i,k).$$

The optimal parameters $\hat{\mathbf{w}}$ minimize a loss function

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{m} \sum_{k=1}^{m} \left(\mathbf{Z}_0(i,k) - \sum_{j=1}^{n} w_j \mathbf{Z}_j(i,k) \right)^2,$$

where $\mathbf{Z}_0$ is a partial order matrix corresponding to the class label vector $\mathbf{y}$. Since the target variable $\mathbf{Z}_0(i,k)$ is binary, we propose to estimate an optimal parameter vector $\hat{\mathbf{w}}$ using one of the standard two-class classification methods. In this paper we use the logistic regression method.

Example 2. *Give an example of a linear combination of matrices* $\mathbf{Z}_1, \mathbf{Z}_2$ *corresponding to the agreed partial orders. Consider a set consisting of the four elements* $X = \{x_1, x_2, x_3, x_4\}$ *and the matrix* $\mathbf{Z}_1$ *from Example 1,*

$$\mathbf{Z}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

corresponding to the partial order

$$\begin{tikzcd} x_1 \arrow[r] & x_2 \arrow[r] \arrow[dr] & x_3 \\ & & x_4 \end{tikzcd}$$

As a second matrix $\mathbf{Z}_2$*, consider*

$$\mathbf{Z}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

corresponding to the linear order

$$\begin{tikzcd} x_1 \arrow[r] & x_2 \arrow[r] & x_3 \arrow[r] & x_4 \end{tikzcd}$$

Construct a matrix $\mathbf{\Psi}$ *as a linear combination of the matrices* $\mathbf{Z}_1, \mathbf{Z}_2$ *with the parameters* $w_1 = w_2 = \frac{1}{2}$*:*

$$\mathbf{\Psi} = \frac{1}{2}\mathbf{Z}_1 + \frac{1}{2}\mathbf{Z}_2 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Since the matrices $\mathbf{Z}_1$ *and* $\mathbf{Z}_2$ *correspond to the almost equivalent object orderings, the only element of the matrix* $\mathbf{\Psi}$ *equals* $\frac{1}{2}$*. This fact can be interpreted as uncertainty in the dominance of the only pair of objects* x_3 *and* x_4*.*

Example 3. *Give an example of matrices* $\mathbf{Z}_1, \mathbf{Z}_2$ *of the disagreed partial orders. Consider the matrix* $\mathbf{Z}_1$ *from the previous example,*

$$\mathbf{Z}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

and as a matrix $\mathbf{Z}_2$ consider

$$\mathbf{Z}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix},$$

corresponding to the linear order

$$x_1 \longleftarrow x_2 \longleftarrow x_3 \longleftarrow x_4$$

Similarly to the previous example, construct a matrix $\mathbf{\Psi}$ as a liner combination of the matrices $\mathbf{Z}_1, \mathbf{Z}_2$ with the parameters $w_1 = w_2 = \frac{1}{2}$:

$$\mathbf{\Psi} = \frac{1}{2}\mathbf{Z}_1 + \frac{1}{2}\mathbf{Z}_2 = \begin{pmatrix} 1 & 1/2 & 1/2 & 1/2 \\ 1/2 & 1 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1 & 0 \\ 1/2 & 1/2 & 1/2 & 1 \end{pmatrix}.$$

Since the matrices $\mathbf{Z}_1$ and $\mathbf{Z}_2$ correspond to the almost opposite object orderings, almost all elements of the matrix $\mathbf{\Psi}$ equal $\frac{1}{2}$. This fact can be interpreted as uncertainty in the dominance of the almost all pairs of objects.

Object weights estimation. Estimate parameters λ_k of the function (9) using the partial order matrix $\mathbf{\Psi}$ estimation,

$$f(\mathbf{x}_i) = \phi\left(\sum_{k=1}^{m} \lambda_k \Psi(\mathbf{x}_i, \mathbf{x}_k)\right).$$

To estimate the parameters λ_k use a logistic regression method. Classify an object $\mathbf{x}_i$ as follows,

$$f(\mathbf{x}_i) = \begin{cases} y_1 \text{ if } u(\mathbf{x}_i) \leq \mu_1, \\ y_2 \text{ if } \mu_1 < u(\mathbf{x}_i) \leq \mu_2, \\ ..., \\ y_K \text{ if } \mu_K < u(\mathbf{x}_i), \end{cases}$$

where the utility function

$$u(\mathbf{x}_i) = \sum_{k=1}^{m} \lambda_k \Psi(\mathbf{x}_i, \mathbf{x}_k).$$

Here $\mu_1, ..., \mu_K$ are the decision rule parameters dividing the real axis $\mathbb{R}$ to the into the sets of correspondence to the class labels $l_1, ..., l_K$. The optimal parameters are estimated by minimization of the loss function S,

$$S(\boldsymbol{\lambda}, \boldsymbol{\mu}) = -\sum_{i=1}^{m} \log\big(\sigma(\mu_{y_i} - \boldsymbol{\lambda}^T \boldsymbol{\psi}_i) - \sigma(\mu_{y_{i-1}} - \boldsymbol{\lambda}^T \boldsymbol{\psi}_i)\big) \to \min,$$

where σ is a sigmoid function,

$$\sigma(t) = 1/\big(1 + \exp(-t)\big),$$

and $\boldsymbol{\psi}_i$ is a column i of the matrix $\mathbf{\Psi}$.

3.4. Stochastic ordinal classification algorithm

To estimate parameters of the model (9) we propose a two-stage algorithm, like we have said before. In the first stage we estimate the partial order matrix $\mathbf{\Psi}$, where the elements of the matrix $\mathbf{\Psi}$ equal to the values of function Ψ on the objects of the sample $\mathfrak{D}$,

$$\mathbf{\Psi}(i,k) = \Psi(\mathbf{x}_i, \mathbf{x}_k).$$

In the second stage estimate parameters λ_k of a linear combination $\sum\limits_{k=1}^{m} \lambda_k \Psi(\mathbf{x}_i, \mathbf{x}_k)$ from (8).

Partial order matrix estimation. Estimate matrix $\mathbf{\Psi}$ of the pairwise objects dominance using initial approximation as the matrix P. By matrix P we consider

$$P_0 = \begin{pmatrix} P(x_1 \succeq x_1|\mu_1 \succeq \mu_1) & P(x_1 \succeq x_2|\mu_1 \succeq \mu_2) & & & \\ & \ddots & P(x_i \succeq x_j|\mu_i \succeq \mu_j) & & \\ & & \ddots & \ddots & \\ & & & & P(x_n \succeq x_n|\mu_n \succeq \mu_n) \end{pmatrix} \in \mathbb{R}^{n\times n}$$

This matrix can be easily calculated by using sampling methods. Let every element of a matrix $\mathbf{\Psi}$ be a linear combination of elements of the matrices $\mathbf{Z}_1, ..., \mathbf{Z}_n$, according to (6):

$$\mathbf{\Psi}(i,k) = \sum_{j=1}^{n} w_j \mathbf{Z}_j(i,k).$$

Here we have to assume the probability of existing the noise within sample data. So it is necessary to use learning without teacher - that is how will be used matrix P. The optimal parameters $\hat{\mathbf{w}}$ minimize a loss function

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{m} \sum_{k=1}^{m} \left(P_0(i,k) - \sum_{j=1}^{n} w_j \mathbf{Z}_j(i,k) \right)^2,$$

here is considered to be partial order matrix corresponding to the class labels vector $\mathbf{y}$. Since the target variable $P_0(i,k)$ is $\in [0,1]$, we propose to estimate an probability distribution vector $\hat{\mathbf{w}}$ using one of the standard optimization methods. In this paper we use the Mean Least Squares method.

Object weights estimation. Estimate parameters λ_k of the function (9) using the partial order matrix $\mathbf{\Psi}$ estimation,

$$f(\mathbf{x}_i) = \phi\left(\sum_{k=1}^{m} \lambda_k \Psi(\mathbf{x}_i, \mathbf{x}_k) \right).$$

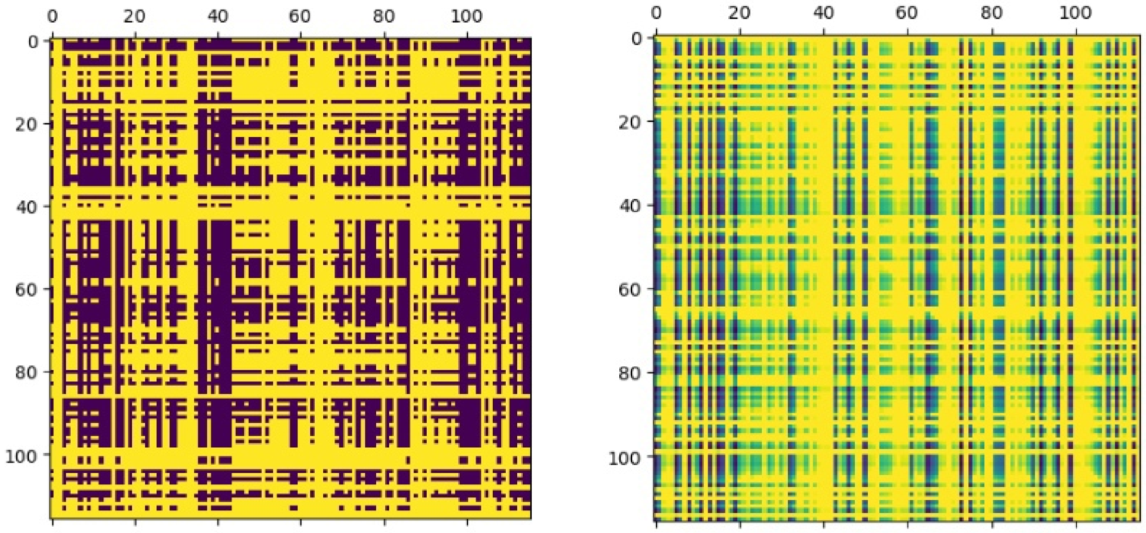
To estimate the parameters λ_k use a Mean Least Squares Method instead of logistic regression to avoid retraining. Furthermore by classifying an object $\mathbf{x}_i$ as we did before. The optimal parameters are estimated by minimization of the loss function S,

$$S(\boldsymbol{\lambda}, \boldsymbol{\mu}, P) = \sum_{i=1}^{m} \left(\mu_{y_i} - \mu_{y_{i-1}} - \boldsymbol{\lambda}^T \boldsymbol{\psi}_i\right) \to \min,$$

and $\boldsymbol{\psi}_i$ is a column i of the matrix $\boldsymbol{\Psi}$. By finding optimal parameters we get the proper estimation of $\boldsymbol{\Psi}$.

4. Computational experiment

Int this section we illustrate the algorithm for the problem of the Football Player List categorization and compare results with the alternative approaches. The fig. 1 shows the first stage of the proposed algorithm, an estimation of the matrix $\boldsymbol{\Psi}$ using the matrices $\mathbf{Z}_0, \mathbf{Z}_1, ..., \mathbf{Z}_n$. The fig. 1 shows the original $\mathbf{Z}_0$ and the estimated $\boldsymbol{\Psi}$ matrices for the Football Player List categorization problem. For this demonstration we construct a binary matrix $\mathbf{Z}_0$ shown at fig. 1(a). The fig. 1(b) shows the second stage of the algorithm for the Football Player List sample. The estimated matrix $\boldsymbol{\Psi}$ is calculated without teacher. Every element of the matrix $\boldsymbol{\Psi}(i,k)$ shows pairwise dominance of the objects $\mathbf{x}_k$ and $\mathbf{x}_i$.



(a) Original $\mathbf{Z}_0$

(b) Estimated $\boldsymbol{\Psi}$

Figure 1: An example of the original matrix $\mathbf{Z}_0$ and the estimated matrix $\boldsymbol{\Psi}$ for the problem of the Football Player List categorization

Algorithms comparison. The table 2 shows comparison of the proposed method with the method of decision oblique trees and with the isotonic regression. The loss function is Hamming distance between the class labels,

$$s(y, \hat{y}) = |y - \hat{y}|.$$

Table 2: Algorithm comparison on the Football Player dataset. The loss function is Hamming distance between class labels.

Algorithm	Learn error	Test error
Partial Orders	1.14	1.69
Isotonic Regression	0.98	1.28
Oblique Decision Trees	0.47	1.06
Stochastic Partial Orders	1.48	1.32

5. *Conclusion*

The proposed method of the ordinal classification problem uses partially ordered sets of expert estimations of features. We considered an alternative problem statement for the ordinal classification with monotonic constraints. Each partially ordered feature set corresponds to the partial order cone and to the matrix of the partial order graph. We find the solution of an ordinal classification problem as the projection to the cones superposition. The proposed method is compared with alternative approaches on the Football Player List dataset. The quality of classification was improved using the ideas considered in the present paper.

References

[1] Wojciech Kotlowski and Roman Slowinski. On nonparametric ordinal classification with monotonicity constraints. IEEE Transactions on Knowledge and Data Engineering*, 25(11):2576–2589, 2013. https://doi.org/http://doi.ieeecomputersociety.org/10.1109/TKDE.2012.204 doi:http://doi.ieeecomputersociety.org/10.1109/TKDE.2012.204.*

[2] Salvatore Corrente, Salvatore Greco, Milosz Kadzinski, and Roman Slowinski. Robust ordinal regression in preference learning and ranking. Machine Learning*, 93:381–422, 2013.*

[3] Diane Ahrens Ali Fallah Tehrani, Marc Strickert. A class of monotone kernelized classifiers on the basis of the choquet integral. Expert Systems*, 37(3), 2020. URL: http://dx.doi.org/10.1111/exsy.12506, https://doi.org/10.1111/exsy.12506 doi:10.1111/exsy.12506.*

[4] *Wouter Duivesteijn and Ad Feelders. Nearest neighbour classification with monotonicity constraints. In Walter Daelemans, Bart Goethals, and Katharina Morik, editors,* Machine Learning and Knowledge Discovery in Databases*, volume 5211 of* Lecture Notes in Computer Science*, pages 301–316. Springer Berlin Heidelberg, 2008. URL: http://dx.doi.org/10.1007/978-3-540-87479-* $9_38, https : //doi.org/10.1007/978-3-540-87479-9_38doi : 10.1007/978-3-540-87479-9_38$

[5] *Jorge M. Arevalillo. Data science methods for response, incremental response and rate sensitivity to response modelling in banking.* Expert Systems*, 41(10), 2024. URL: http://dx.doi.org/10.1111/exsy.13644, https://doi.org/10.1111/exsy.13644 doi:10.1111/exsy.13644.*

[6] *Viviane M. Lelis Eduardo Guzmán, María-Victoria Belmonte. Ensemble methods for meningitis aetiology diagnosis.* Expert Systems*, 39(8), 2022. URL: http://dx.doi.org/10.1111/exsy.12996, https://doi.org/10.1111/exsy.12996 doi:10.1111/exsy.12996.*

[7] *Juan A. Aledo Jose A. Gámez Enrique G. Rodrigo, Juan C. Alfaro. Efficient ensembles of distance-based label ranking trees.* Expert Systems*, 41(4), 2023. URL: http://dx.doi.org/10.1111/exsy.13525, https://doi.org/10.1111/exsy.13525 doi:10.1111/exsy.13525.*

[8] *Pablo Barreiro Roi Durán Rosa Crujeiras María Loureiro Eduardo Sánchez Ameed Almomani, Paula Saavedra. Application of choice models in tourism recommender systems.* Expert Systems*, 40(3), 2022. URL: http://dx.doi.org/10.1111/exsy.13177, https://doi.org/10.1111/exsy.13177 doi:10.1111/exsy.13177.*

[9] *Baruch Keren Yossi Hadad. A decision-making support system module for customer segmentation and ranking.* Expert Systems*, 40(2), 2022. URL: http://dx.doi.org/10.1111/exsy.13169, https://doi.org/10.1111/exsy.13169 doi:10.1111/exsy.13169.*

[10] *J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems.* Lecture Notes in Computer Science*, 4321:291–324, 2007.*

[11] *Andrew Trotman. Learning to rank.* Information Retrieval*, 8:381, 2005.*

[12] *Nikita Spirin and Konstantin Vorontsov. Learning to rank with nonlinear monotonic ensemble. In Carlo Sansone, Josef Kittler, and Fabio Roli, editors,* Multiple Classifier Systems*, volume 6713 of* Lecture Notes in Computer Science*, pages 16–25. Springer Berlin Heidelberg, 2011. URL: http://dx.doi.org/10.1007/978-3-642-21557-* $5_4, https : //doi.org/10.1007/978-3-642-21557-5_4doi : 10.1007/978-3-642-21557-5_4$.

[13] *Johannes Fuernkranz and Eyke Huellermeier.* Preference learning. *Springer, 2011.*

[14] Yibo Miao Zefan Cai Zhe Yang Liang Chen Helan Hu Runxin Xu Qingxiu Dong Ce Zheng Shanghaoran Quan Wen Xiao Ge Zhang Daoguang Zan Keming Lu Bowen Yu Dayiheng Liu Zeyu Cui Jian Yang Lei Sha Houfeng Wang Zhifang Sui Peiyi Wang Tianyu Liu Baobao Chang Bofei Gao, Feifan Song. Towards a unified view of preference learning for large language models: A survey. Semantic Scholar*, 2024. URL: http://dx.doi.org/10.48550/arXiv.2409.02795, https://doi.org/10.48550/arXiv.2409.02795 doi:10.48550/arXiv.2409.02795.*

[15] Marco F. Huber Nadia Burkart, Sebastian Robert. Are you sure? prediction revision in automated decision-making. Expert Systems*, 38(1), 2020. URL: http://dx.doi.org/10.1111/exsy.12577, https://doi.org/10.1111/exsy.12577 doi:10.1111/exsy.12577.*

[16] Riccardo Ceccato Elena Barzizza, Nicolò Biasetton. Multi-aspect permutation tests for model selection. Expert Systems*, 41(3), 2023. URL: http://dx.doi.org/10.1111/exsy.13492, https://doi.org/10.1111/exsy.13492 doi:10.1111/exsy.13492.*

[17] Weiwei Cheng, Michael Rademaker, Bernard De Baets, and Eyke Huellermeier. Predicting partial orders: Ranking with abstention. Machine Learning and Knowledge Discovery in Databases*, 6321:215–230, 2010. https://doi.org/10.1007/978-3-642-15880-*$3_20doi: 10.1007/978-3-642-15880-3_20$.

[18] M.P. Kuznetsov and V.V. Strijov. Methods of expert estimations concordance for integral quality estimation. Expert Systems with Applications*, 41(4):1988–1996, March 2014. URL: http://www.sciencedirect.com/science/article/pii/S0957417413007173, https://doi.org/http://dx.doi.org/10.1016/j.eswa.2013.08.095 doi:http://dx.doi.org/10.1016/j.eswa.2013.08.095.*

[19] Akram Dehnokhalaji, Pekka J. Korhonen, Murat Köksalan, Nasim Nasrabadi, and Jyrki Wallenius. Convex cone-based partial order for multiple criteria alternatives. Decision Support Systems*, 51(2):256 – 261, 2011. Multiple Criteria Decision Making and Decision Support Systems. URL: http://www.sciencedirect.com/science/article/pii/S0167923610002009, https://doi.org/http://dx.doi.org/10.1016/j.dss.2010.11.019 doi:http://dx.doi.org/10.1016/j.dss.2010.11.019.*

[20] Souhila Kaci. Working with Preferences: Less Is More*. Springer Berlin Heidelberg, 2011.*

[21] R. Busa-Fekete, B. Szoreny, P. Weng, W. Cheng, and E. Huellermeier. Top-k selection based on adative sampling of noisy preferences. Proc. ICML-13, 30th International Conference on Machine Learning (JMLR)*, 28(3):1130–1138, 2013.*

[22] *William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things.* J. Artif. Int. Res.*, 10(1):243–270, May 1999. URL: http://dl.acm.org/citation.cfm?id=1622859.1622867.*

[23] *Sebastian Litzinger Wilhelm Rödder, Andreas Dellnitz. Analysing terrorist networks – an entropy-driven method.* Expert Systems*, 39(10), 2021. URL: http://dx.doi.org/10.1111/exsy.12720, https://doi.org/10.1111/exsy.12720 doi:10.1111/exsy.12720.*

[24] *Samuel Vaiter Pierre C Bellec, Joseph Salmon. A sharp oracle inequality for graph-slope.* Electronic journal of electronics*, 11 2017. https://doi.org/10.48550/1706.06977 doi:10.48550/1706.06977.*