
A NOVEL LARGE-SCALE ORDINAL REGRESSION MODEL*

A PREPRINT

Yong Shi

School of Economics and Management
University of Chinese Academy of Sciences
Beijing, China
yshi@ucas.ac.cn

Huadong Wang

Research Center on Fictitious Economy & Data Science
University of Chinese Academy of Sciences
Beijing, China
wanghuadong14@mails.ucas.ac.cn

Xin Shen

Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Hong Kong, China
xshen@se.cuhk.edu.hk

Lingfeng Niu[†]

School of Economics and Management
University of Chinese Academy of Sciences
Beijing, China
niulf@ucas.ac.cn

December 21, 2018

ABSTRACT

Ordinal regression (OR) is a special multiclass classification problem where an order relation exists among the labels. Recent years, people share their opinions and sentimental judgments conveniently with social networks and E-Commerce so that plentiful large-scale OR problems arise. However, few studies have focused on this kind of problems. Nonparallel Support Vector Ordinal Regression (NPSVOR) is a SVM-based OR model, which learns a hyperplane for each rank by solving a series of independent sub-optimization problems and then ensembles those learned hyperplanes to predict. The previous studies are focused on its nonlinear case and got a competitive testing performance, but its training is time consuming, particularly for large-scale data. In this paper, we consider NPSVOR’s linear case and design an efficient training method based on the dual coordinate descent method (DCD). To utilize the order information among labels in prediction, a new prediction function is also proposed. Extensive contrast experiments on the text OR datasets indicate that the carefully implemented DCD is very suitable for training large data.

Keywords Ordinal regression · Linear classification · Dual coordinate descent method

1 Introduction

Ordinal regression (OR) is a supervised learning problem that aims to learn a rule to predict labels of an ordinal scale, i.e., labels from a discrete but ordered set [1]. In contrast to metric regression, it features finite ranks, among which the metric distances are not defined. Compared with ranks in classification, those in OR are different from the labels of multiple classes in classification problems due to the order information. Its prediction usually requires the predicted label as close as possible to the true label. OR is generally considered as an intermediate problem between metric regression and multiclass classification.

OR is widely employed in many domains [2–13]. In [2], a novel classification algorithm for ordinal data is adopted to objectively evaluate the aesthetic result of breast cancer conservative treatment. Fernandez-Navarro et al. [3] addresses the sovereign rating problem by an ordinal regression approach as a result of its ordinal nature of dependent variable. In [4], P.A. Gutiérrez et al. deem wind speed as a discrete variable and divides it into four levels, and thus it turns out to be a problem of classification. Yoon et al. [5] propose an algorithm for adaptive, sequential classification in systems

*This paper has been submitted to Neural Computing and Applications

[†]Corresponding author

with unknown labeling errors for biomedical application of Brain Computer Interfacing. In facial recognition, Yan [10] proposes a cost-sensitive OR approach to predict the beauty information of face images captured in unconstrained environments which behaves like human beings; Rudovic et al. [11] uses an ordinal manifold structure to encode temporal dynamics of facial expressions, ordinal relationships between their intensities and intrinsic topology of multidimensional continuous facial data; Rudovic et al. [12] developed Context-sensitive Conditional Ordinal Random Field model in estimating intensity levels of spontaneously displayed facial expressions; Chang et al. [13] uses relative order of age labels to estimate age group and even precisely predict one's age only by facial images. All these works prove that utilizing the existing order information among labels in modeling can significantly improve the prediction performance.



Figure 1: An example of restaurant reviews (from Yelp Dataset Challenge)

With the development of Internet and mobile communication technology, more and more people can freely express their opinions and personal preferences on all kinds of entities such as products and services. These reviews are useful for other users to make more sensible decisions, and for merchants to improve their service. A notable example is the restaurant's reviews which is shown in Figure 1 (selected from Yelp Dataset Challenge³). The reviewer give review and corresponding rating about the restaurant by {one star, two star, ..., five star}, depending on how much he or she likes it. The number of stars represent the different levels of preference. In Yelp dataset competition, one of the tasks is to predict the overall rating based on the review text, which can be used in the recommendation system. Obviously, this example can be treated as an OR problem. Under such circumstances, the data of OR problem is usually in text type so that the feature vector is already high-dimensional which is extracted from text using n-gram words, such as word frequency vector or TF-IDF. To our knowledge, few researches focus on large-scale high dimensional OR problems, [14] has done some related works but it limited in feature selection. For OR problems, there still lacks relevant methods to be proposed and this paper aims to solve large-scale OR.

So far, a great deal of methods about OR have been proposed, such as SVOR [1], RedSVM [15], GPOR [16], KDLOR [17], SVMOP [18], AL [19] etc. However, the characteristics of large scale, high dimension and high sparsity make traditional nonlinear model difficult to deal with. So we are considering some faster and more efficient algorithms for the large-scale and high dimensional problems. For OR problems, Support Vector Machines (SVMs) [20–22] have shown promising results. In addition, linear SVMs [23, 24] provide state-of-the-art prediction accuracy and is quite efficiently when handling problems with a large number of features in the field of text mining. Because feature vectors are already linearly separable and the structure of the linear model is relatively simple. Such phenomena supports the hypothesis that linear SVM is a better alternative method to solve large scale OR.

According to the taxonomy [25], OR methods mainly fall into three categories, naive approach, ordinal binary decomposition approaches and threshold models. Ordinal binary decomposition approaches and threshold models fail to consider the data distribution of different ranks properly and might lead to unreasonable results. To better utilize the distribution information, the novel model called Nonparallel Support Vector Ordinal Regression (NPSVOR) was proposed in [26], and the alternating direction method of multipliers(ADMM) has been designed for its nonlinear dual model. Numerical experiments have shown that NPSVOR is superior to other SVM-based methods. Therefore, it is necessary to further study linear NPSVOR so that it can be applied to text mining.

This paper has the contributions as follows:

- We have studied the large scale ordinal regression problems. Consider the linear NPSVOR, the coordinate descent (CD) with careful design is provided. Since optimizing fewer variables, the algorithm is faster than directly extending the DCD algorithm of LIBLINEAR to linear NPSVOR.
- Considering the order information existing in labels, we proposed a new prediction function, which is better than the prediction function on the minimal distance principle. Compared with the latter one, the proposed

³https://www.yelp.com/dataset_challenge

prediction function can effectively reduce the generation of ambiguous decision regions and can give the lower prediction error.

- Fifty text ordinal regression datasets are provided or collected for testing and comparing. Those datasets are collected from different areas including sentiment analysis, film reviews, product reviews and health consultation service.

The organization of this paper begins with a brief review of NPSVOR and the coordinate descent method (CD) in Section 2. In Section 3, we discuss the linear NPSVOR and propose an efficient implementation of coordinate descent methods. We conduct experiments in Section 4 on some collected large scale ordinal regression datasets. Section 4.4 concludes this work.

2 Related Works

2.1 Nonparallel Support Vector Ordinal Regression

Consider an OR problem with n training examples $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1,\dots,n}$, where each example \mathbf{x}_i drawn from a domain $\mathcal{X} \subseteq \mathbb{R}^m$ and each ordinal label y_i is an integer from a finite set of consecutive integers $\mathcal{Y} = \{1, 2, \dots, p\}$.

In [26], a SVM-based model, called Nonparallel Support Vector Ordinal Regression (NPSVOR), was proposed for OR problem and got a superior performance among the existing state-of-art SVM-based models. This method firstly constructs three index sets for each rank $k \in \{1, 2, \dots, p\}$,

$$\mathcal{L}_k = \{i | \forall i, y_i < k\}, \mathcal{I}_k = \{i | \forall i, y_i = k\}, \mathcal{R}_k = \{i | \forall i, y_i > k\}. \quad (1)$$

where y_i is the rank of the instance \mathbf{x}_i . Its purpose is to find a linear hyperplane $f_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} + b_k$ for each rank $k \in \mathcal{Y}$ based on the triple decomposition (1). The model is constructed as the following optimization problem:

$$\min_{\mathbf{w}_k, b_k, \xi_k^+, \xi_k^-, \eta_k} \frac{1}{2} \|\mathbf{w}_k\|_2^2 + C_1 \sum_{i \in \mathcal{I}_k} (\xi_{ki}^+ + \xi_{ki}^-) + C_2 \sum_{i \notin \mathcal{I}_k} \eta_{ki} \quad (2a)$$

$$\text{s.t.} \quad -\varepsilon - \xi_{ki}^- \leq \mathbf{w}_k^\top \mathbf{x}_i + b_k \leq \varepsilon + \xi_{ki}^+, i \in \mathcal{I}_k, \quad (2b)$$

$$\mathbf{w}_k^\top \mathbf{x}_i + b_k \leq -1 + \eta_{ki}, i \in \mathcal{L}_k, \quad (2c)$$

$$\mathbf{w}_k^\top \mathbf{x}_i + b_k \geq 1 - \eta_{ki}, i \in \mathcal{R}_k, \quad (2d)$$

$$\xi_{ki}^+ \xi_{ki}^- \geq 0, i \in \mathcal{I}_k, \quad (2e)$$

$$\eta_{ki} \geq 0, i \notin \mathcal{I}_k. \quad (2f)$$

where $C_1, C_2 > 0$ are model parameters.

Denote the optimal proximal hyperplane of rank k obtained by (2) as $f_k(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_k^* + b_k^* = 0$ for all $k = 1, 2, \dots, p$. In the model (2), the first term is a regularization item of \mathbf{w}_k , and both the second term and third term are Hinge losses. The second term requires learning the hyperplane $f_k(\mathbf{x})$ as close as possible to the k -th class samples $\{\mathbf{x}_i : i \in \mathcal{I}_k\}$. The third term requires that the samples with other ranks to be as far away as possible from the hyperplane $f_k(\mathbf{x})$. In order to utilize the order information contained among labels, the sample sets $\{\mathbf{x}_i : i \in \mathcal{R}_k\}$ and $\{\mathbf{x}_i : i \in \mathcal{L}_k\}$ are required to locate on both sides of the hyperplane. Figure 2 gives a geometrical illustration of the construction of proximal hyperplane in \mathbb{R}^2 .

Since NPSVOR is an extension model of the binary classification method Twin SVM [27] or NPSVM [28] in OR problems, the prediction rule takes the minimal distance principle and is defined as

$$r(\mathbf{x}) = \arg \min_{k \in \{1, \dots, p\}} |(\mathbf{w}_k^*)^\top \mathbf{x} + b_k^*. \quad (3)$$

The OR predictor assigns the new sample \mathbf{x} in the class which has the minimum absolute value of the learned function.

Wang et al. [26] established the kernel NPSVOR model by introducing the kernel trick on its dual model, and designed the alternating direction method of multipliers (ADMM). But this algorithm only adapts to small-scale training data, and even with SMO, it is still difficult to handle large-scale problem.

2.2 Coordinate descent method

Coordinate Descent Method (CD) is a common unconstrained optimization technique, which updates one variable at a time by minimizing a single-variable sub-problem. It is very popular recently for solving large-scale unconstrained

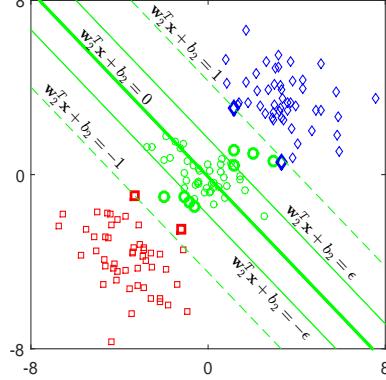


Figure 2: The construction of proximal hyperplane for rank 2. “ \square ”, “ \circ ” and “ \diamond ” stand for rank 1,2,3, respectively. Support vectors are marked in bold.

optimization problems. Hong et al. [29] provide a unified iteration complexity analysis for a family of general block coordinate descent methods and show that CD under block successive upper-bound minimization framework can achieve a global sublinear iteration complexity of $O(1/r)$, where r is the iteration index. CD has been exploited widely for linear SVM in large-scale scenarios. Hsieh et al. [23] proposed using CD for solving primal L2-SVM. Experiments show that their approach obtains an useful model more quickly. Hsieh et al. [23] proposed a dual coordinate descent method (DCD) for linear SVM which investigated CDs for the dual problem of both SVM with hinge loss (L1-SVM) and SVM with the square hinge loss (L2-SVM). They proved that an ε -optimal solution can be obtained in $O(\log(1/\varepsilon))$ iterations. The shrinking strategy is also applied to make the CD faster. Yuan et al. [30] proposed a coordinate descent method for L1-regularized problems by extending Chang et al. [31]’s approach for L2-regularized classifiers. Tseng and Yun [32] broadly discussed decomposition methods for L1-regularized problems. Ho and Lin [33] extended LIBLINEAR’s SVC solvers TRON and DCD to solve large-scale linear SVR problems. They are different because the former is to solve the primal problem, while the latter is for the dual problem. In this paper, we aim at applying DCD to NPSVOR. A coordinate descent method updates one component of α of its dual problem at a time by solving an one-variable sub-problem. It is competitive if one can exploit efficient ways to solve the sub-problem.

3 Linear Nonparallel Support Vector Ordinal Regression

In some applications, we include a bias term b_k for each hyperplane in NPSVOR problems. To have a simpler derivation without the bias term b_k , one often augments each instance with an additional dimension:

$$\mathbf{x}^\top \leftarrow [\mathbf{x}^\top, 1], \mathbf{w}_k^\top \leftarrow [\mathbf{w}_k^\top, b_k], k = 1, \dots, p.$$

The model is constructed as the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}_k} & \frac{1}{2} \mathbf{w}_k^\top \mathbf{w}_k + C_1 \sum_{i \in \mathcal{I}_k} \max(|\mathbf{w}_k^\top \mathbf{x}_i| - \varepsilon, 0) \\ & + C_2 \sum_{i \notin \mathcal{I}_k} \max(1 - \tilde{y}_{ki} \mathbf{w}_k^\top \mathbf{x}_i, 0) \end{aligned} \quad (4)$$

where $C_1, C_2 > 0$ are the model parameters, $k = 1, 2, \dots, p$, and $\tilde{y}_{ki} = 1$, if $i \in \mathcal{R}_k$, otherwise, $\tilde{y}_{ki} = -1$. It is obvious that the objective function is not differentiable.

The optimal proximal hyperplane of rank k obtained by (4) is denoted as $(\mathbf{w}_k^*)^\top \mathbf{x} = 0$. The prediction rule can be written as follows

$$r(\mathbf{x}) = \arg \min_{k \in \{1, \dots, p\}} |(\mathbf{w}_k^*)^\top \mathbf{x}|. \quad (5)$$

This prediction function assigns the label based on the minimal distance from \mathbf{x} to each nonparallel hyperplane. But it does not consider the order information exists in ranks during prediction so that it easily leads to ambiguity in decision region. We will have a deeper discussion on it and propose a new predictor in Section 3.2.

To emphasize that p proximal hyperplanes are computed independently and are not necessarily parallel with each other, model (4) is named as Nonparallel Support Vector Ordinal Regression (NPSVOR) in this paper. Obviously, when the number of ranks is two, NPSVOR degenerates to nonparallel support vector machine for binary classification in [28].

From [26], the dual problem of (4) has the following form

$$\begin{aligned} \min_{\alpha_k} f_k(\boldsymbol{\alpha}_k) &= \frac{1}{2} \mathbf{w}_k^\top \mathbf{w}_k + \varepsilon \sum_{i \in \mathcal{I}_k} (\alpha_k^+ + \alpha_k^-) - \sum_{i \notin \mathcal{I}_k} \beta_{ki} \\ \text{s.t. } &0 \leq \alpha_{ki}^+, \alpha_{ki}^- \leq C_1, i \in \mathcal{I}_k \\ &0 \leq \beta_{ki} \leq C_2, i \notin \mathcal{I}_k \end{aligned} \quad (6)$$

where $\boldsymbol{\alpha}_k = ((\boldsymbol{\alpha}_k^-)^\top, (\boldsymbol{\alpha}_k^+)^{\top}, \boldsymbol{\beta}_k^\top)^\top$.

The primal-dual relationship indicates that primal optimal solution \mathbf{w}_k and dual optimal solution $\boldsymbol{\alpha}_k^+$, $\boldsymbol{\alpha}_k^-$ and $\boldsymbol{\beta}_k$ satisfy

$$\mathbf{w}_k = - \sum_{i \in \mathcal{I}_k} (\alpha_{ki}^+ - \alpha_{ki}^-) \mathbf{x}_i + \sum_{i \notin \mathcal{I}_k} \tilde{y}_{ki} \beta_{ki} \mathbf{x}_i. \quad (7)$$

An important property of the dual problem (6) is that at the optimum condition,

$$(\alpha_{ki}^+)^* (\alpha_{ki}^-)^* = 0, \forall i \in \mathcal{I}_k. \quad (8)$$

The dual problem of subproblem of NPSVOR for rank k has $n + |\mathcal{I}_k|$ variables, while SVC has only n . With the increasing of the size of the training samples, the computational cost will increase significantly. If a dual-based solver is applied without a careful design, the cost may be significantly higher than that of SVC.

3.1 Dual Coordinate Descent Method for NPSVOR

In this section, we introduce DCD, a coordinate descent method for the dual form of NPSVOR. We first extend the setting of [23] to NPSVOR and then propose a better algorithm using properties of NPSVOR. Since each subproblem (6) have same form, we disregard the subscript for rank k for simplicity , such as. some symbols \mathbf{w}_k , $\boldsymbol{\alpha}_k^+$, $\boldsymbol{\alpha}_k^-$, $\boldsymbol{\beta}_k$, \mathcal{I}_k will be replaced by \mathbf{w} , $\boldsymbol{\alpha}^+$, $\boldsymbol{\alpha}^-$, $\boldsymbol{\beta}$, \mathcal{I} in following discussion.

3.1.1 A Direct Extension from the SVC's DCD solver to NPSVOR

Let $\boldsymbol{\alpha} = ((\boldsymbol{\alpha}^-)^\top, (\boldsymbol{\alpha}^+)^{\top}, \boldsymbol{\beta}^\top)^\top$. Suppose the size of index set \mathcal{I} be l , the upper bound of the i -th variable of $\boldsymbol{\alpha}$ be

$$U_i = \begin{cases} C_1, & 1 \leq i \leq 2l \\ C_2, & 2l < i \leq n+l \end{cases}$$

A coordinate descent method sequentially updates the i -th variable by solving the following subproblem.

$$\begin{aligned} \min_d f(\boldsymbol{\alpha} + d\mathbf{e}_i) \\ \text{s.t. } 0 \leq \alpha_i + d \leq U_i \end{aligned} \quad (9)$$

$\mathbf{e}_i \in \mathbb{R}^{n+l}$ is a vector with i -th element one and others zero.

We compute the

$$A_i = \mathbf{x}_i^\top \mathbf{x}_i, B_i = \mathbf{w}^\top \mathbf{x}_i, \quad (10)$$

which is constants obtained using of the previous iteration $\boldsymbol{\alpha}$.

The objective function of (9) is a simple quadratic function of d :

$$f(\boldsymbol{\alpha} + d\mathbf{e}_i) = \frac{1}{2} \bar{A}_i d^2 + G_i d + \text{constants}$$

where

$$\bar{A}_i = \begin{cases} A_i & \text{if } 1 \leq i \leq l \\ A_{i-l} & \text{if } l < i \leq n+l \end{cases} \quad (11)$$

and

$$G_i = \begin{cases} -B_i + \varepsilon & \text{if } 1 \leq i \leq l \\ B_{i-l} + \varepsilon & \text{if } l < i \leq 2l \\ B_{i-l} - 1 & \text{if } 2l < i \leq n+l \end{cases} \quad (12)$$

In fact, G_i is the i -th component of the gradient ∇f . It is obvious that the optimal value d of (9) can be solved in a closed form, so that α_i is updated by

$$\alpha_i \leftarrow \min(\max(\alpha_i - \frac{G_i}{\bar{A}_i}, 0), U_i) \quad (13)$$

If $\bar{\alpha}_i$ is the current value and α_i is the value after the updating by (13), then vector \mathbf{w} can be maintained by

$$\mathbf{w} \leftarrow \begin{cases} \mathbf{w} - (\alpha_i - \bar{\alpha}_i)\mathbf{x}_i & 1 \leq i \leq l \\ \mathbf{w} + (\alpha_i - \bar{\alpha}_i)\mathbf{x}_i & \text{otherwise} \end{cases} \quad (14)$$

Hsieh et al. [23] check the violation of the optimality condition of (17) for the stopping condition because $\boldsymbol{\alpha}$ is optimal if and only if $v_i = 0$, which is defined as

$$v_i = \begin{cases} \min(G_i, 0) & \text{if } \alpha_i = 0 \\ \max(G_i, 0) & \text{if } \alpha_i = U_i \\ G_i & \text{if } 0 < \alpha_i < U_i \end{cases} \quad (15)$$

If $v_i = 0$, then (13) and (15) imply that α_i needs not be updated. We show the overall procedure in Algorithm 1.

Algorithm 1 DCD-1: a dual coordinate descent method for (4)

```

1: Initial  $\boldsymbol{\alpha} = \mathbf{0} \in \Re^{n+l}$  and the corresponding  $\mathbf{w} = \mathbf{0}$ .
2: Compute  $A_i = \mathbf{x}_i^\top \mathbf{x}_i, i = 1, \dots, n$ .
3: while  $\boldsymbol{\alpha}$  is not optimal, do
4:   for  $i = 1, \dots, n+l$  do
5:     Obtain  $\bar{A}_i$  and  $G_i$  by (11) and (12).
6:     if  $v_i \neq 0$ , then
7:       Update  $\alpha_i$  by (13).
8:       Update  $\mathbf{w}$  by (14).
9:     end if
10:   end for
11: end while

```

Hsieh et al. [23] apply a shrinking strategy to make Dual Coordinate Descent Method (DCD) faster in LIBLINEAR's SVC solvers. By gradually removing some variables, smaller optimization problems are solved to save the training time. They remove the variables which are likely to be bounded at optimum. Although this DCD for SVC can be extended directly to solve the large-scale linear NPSVOR problem (6) as given in Algorithm 1, the procedure does not take NPSVOR's special structure into account. For example, the number of variables of its dual model is larger than the size of training samples n , which obviously affects the efficiency of the model when the sample size is too large. Note that α_i^+ and α_i^- in (6) (omit the index k) are very related. We can see that in the following situations some operations using DCD directly are redundant.

- The optimal solution $\boldsymbol{\alpha}$ of (6) satisfies $\alpha_i^+ \alpha_i^- = 0, \forall i \in \mathcal{I}$ (The the rank index k is omitted). Since α_i^+ and α_i^- cannot be nonzero at the same time⁴, one of α_i^+ and α_i^- is nonzero at optimum (without loss of generality, assume $\alpha_i^+ = 0$), then the another one (α_i^-) must be zero. Thus, optimization of the variable α_i^- is not necessary. We do not need to compute the corresponding $B_i = \mathbf{w}^\top \mathbf{x}_i$ and update \mathbf{w} . Therefore, some operations are wasted. Although shrinking strategy can partially solve this problem, alternatively we can explicitly use the property (8) in designing the CD algorithm.
- Some operations in calculating gradient are unnecessary because the stopping condition needs the largest component of the optimal condition violation (15) of all variables. Suppose $\alpha_i^- > 0$ and $\alpha_i^+ = 0$, if the optimality condition at α_i^+ is not satisfied yet, then we have

$$v_{i+l} = G_{i+l} = \tilde{y}_i \mathbf{w}^\top \mathbf{x}_i + \varepsilon < 0,$$

that is,

$$0 < -v_{i+l} < -\tilde{y}_i \mathbf{w}^\top \mathbf{x}_i + \varepsilon = G_i$$

And since $v_i = G_i$, we have $|v_i| > |v_{i+l}|$. This shows us a larger absolute violation of the optimality condition occurs at α_i^- . Thus, when $\alpha_i^- > 0$ and $\alpha_i^+ = 0$, checking the G_i and v_i of α_i^+ is not necessary.

In the following discussion, we propose a method to address these issues.

⁴From the KKT conditions, we have $(\mathbf{w}^\top \mathbf{x}_i + \varepsilon)\alpha_i^+ = 0$ and $(\mathbf{w}^\top \mathbf{x}_i - \varepsilon)\alpha_i^- = 0$. If α_i^+ and α_i^- are none-zero at optimum, there exists \mathbf{w} such that $\mathbf{w}^\top \mathbf{x}_i + \varepsilon = 0, \mathbf{w}^\top \mathbf{x}_i - \varepsilon = 0$, but it is impossible.

3.1.2 A New Coordinate Descent Method by Solving \mathbf{a}^+ and \mathbf{a}^- Together

It should be noted that an important property of the dual problem (6) is that at the optimum, $\alpha_i^+ \alpha_i^- = 0, i \in \mathcal{J}$, which, together with the condition $\alpha_i^+, \alpha_i^- \geq 0$, imply that at the optimum

$$\alpha_i^+ + \alpha_i^- = |\alpha_i^+ - \alpha_i^-|, i \in \mathcal{J}.$$

Denote $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\top$, where

$$\alpha_i = \begin{cases} \alpha_i^+ - \alpha_i^- & i \in \mathcal{J}, \\ \beta_i & i \notin \mathcal{J}. \end{cases}$$

Then \mathbf{w} defined by (7) becomes

$$\mathbf{w} = - \sum_{i \in \mathcal{J}} (\alpha_i^+ - \alpha_i^-) \mathbf{x}_i + \sum_{i \notin \mathcal{J}} \tilde{y}_i \beta_i \mathbf{x}_i = \sum_{i=1}^n \tilde{y}_i \alpha_i \mathbf{x}_i \quad (16)$$

The problem (6) can be transformed as:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) &= \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \sum_{i \in \mathcal{J}} \varepsilon |\alpha_i| - \sum_{i \notin \mathcal{J}} \alpha_i \\ \text{s.t. } &-C_1 \leq \alpha_i \leq C_1, i \in \mathcal{J} \\ &0 \leq \alpha_i \leq C_2, i \notin \mathcal{J} \end{aligned} \quad (17)$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n)^\top$, the \mathbf{w} get defined via $\boldsymbol{\alpha}$ as (16).

A coordinate descent method sequentially updates one variable when fixing other variables. Assume $\boldsymbol{\alpha}$ is the current iterate and its i -th component, denoted as a scalar variable s , is being updated.

(1) **For** $i \notin \mathcal{J}$, since

$$f(\boldsymbol{\alpha} + (s - \alpha_i) \mathbf{e}_i) = \frac{1}{2} A_i (s - \alpha_i)^2 + (B_i - 1)s + \text{const.}$$

where $A_i = \mathbf{x}_i^\top \mathbf{x}_i, B_i = \tilde{y}_i \mathbf{w}^\top \mathbf{x}_i$, which are constants obtained using $\boldsymbol{\alpha}$ of the previous iteration. So we just need to solve the following one-variable sub-problem:

$$\min_{s \in [0, C_2]} h(s) \equiv \frac{1}{2} A_i (s - \alpha_i)^2 + (B_i - 1)s, \quad (18)$$

If $A_i > 0$, obviously the solution is:

$$\alpha_i \leftarrow \min \left(\max \left(\alpha_i - h'(\alpha_i)/A_i, 0 \right), C_2 \right). \quad (19)$$

We thus need to calculate A_i and $h'(\alpha_i)$. First,

$$h'(\alpha_i) = B_i - 1. \quad (20)$$

One can easily see that (18) has an optimum at $s = \alpha_i$ (i.e., no need to update α_i) if and only if $v_i = 0$, where $v_i = h'_P(\alpha_i)$ means the projected gradient

$$h'_P(\alpha_i) = \begin{cases} h'(\alpha_i) & \text{if } 0 < \alpha_i < C_2, \\ \min(0, h'(\alpha_i)) & \text{if } \alpha_i = 0, \\ \max(0, h'(\alpha_i)) & \text{if } \alpha_i = C_2. \end{cases} \quad (21)$$

If $h'_P(\alpha_i) = 0$, then (19) and (21) imply that α_i needs not be updated. Otherwise, we must find the optimal solution of (18).

(2) **For** $i \in \mathcal{J}$, we solve the following one-variable sub-problem:

$$\min_{s \in [-C_1, C_1]} g(s) \equiv \frac{1}{2} A_i (s - \alpha_i)^2 + B_i s + \varepsilon |s|, \quad (22)$$

where $A_i = \mathbf{x}_i^\top \mathbf{x}_i, B_i = \tilde{y}_i \mathbf{w}^\top \mathbf{x}_i$, which are constants obtained using $\boldsymbol{\alpha}$ of the previous iteration.

Although the objective function of (22) is not differentiable, we can still easily compute a simple closed-form solution to this problem by using subdifferential calculus [34]. In detail, the objective function of the subproblem (22) can be represented as

$$g(s) = \varepsilon |s| + \frac{A_i}{2} \left(s - \alpha_i + \frac{B_i}{A_i} \right)^2 + \text{const.}$$

Thus, (22) can be reduced to “soft-thresholding” in signal processing and has a closed-form minimum. Let its derivatives at $s > 0$ and $s < 0$ be the following, respectively

$$\begin{aligned} g'_p(s) &= A_i(s - \alpha_i) + B_i + \varepsilon, \text{ if } s > 0 \\ g'_n(s) &= A_i(s - \alpha_i) + B_i - \varepsilon, \text{ if } s < 0 \end{aligned}$$

If $\mathbf{x}_i \neq \mathbf{0}$ we have $A_i > 0$, then both $g'_p(s)$ and $g'_n(s)$ are strictly monotonic with respect to the variable s , i.e.

$$g'_n(s) < g'_p(s), \forall s \in \mathfrak{R}.$$

So $g(s)$ is a strictly convex quadratic function and has a unique minimum.

Since $g(s)$ is pairwise quadratic, we can consider the following three cases:

- If $g'_p(0) < 0$, the minimum s^* of $g(s)$ will occur at range $s > 0$. Let $g'_p(s) = 0$, we have $s^* = \alpha_i - g'_p(\alpha_i)/A_i$;
- If $g'_n(0) > 0$, the minimum s^* of $g(s)$ will occur at range $s < 0$. Let $g'_n(s) = 0$, we have $s^* = \alpha_i - g'_n(\alpha_i)/A_i$;
- If $g'_n(0) < 0 < g'_p(0)$, then the optimal solution of $g(s)$ can obtain at $s = 0$.

It is easy to known that (22) has the following closed form solution.

$$\alpha_i \leftarrow \min(\max(\alpha_i + d_i, -C_1), C_1) \quad (23)$$

where

$$d_i = \begin{cases} -g'_p(\alpha_i)/A_i, & g'_p(\alpha_i) < A_i\alpha_i \\ -g'_n(\alpha_i)/A_i, & g'_n(\alpha_i) > A_i\alpha_i \\ -\alpha_i, & \text{otherwise} \end{cases} \quad (24)$$

In (24), we simplify the form of solution by using the property

$$g'_p(\alpha_i) = B_i + \varepsilon \text{ and } g'_n(\alpha_i) = B_i - \varepsilon \quad (25)$$

If $\bar{\alpha}_i$ is the current value and α_i is the value after the update, we can maintain \mathbf{w} by

$$\mathbf{w} \leftarrow \mathbf{w} + (\alpha_i - \bar{\alpha}_i)\tilde{y}_i\mathbf{x}_i. \quad (26)$$

For the stopping condition, as a beginning, we study how to measure the violation of the optimality condition of (17) during the optimization procedure. After considering all situations, we know that

$$\alpha_i \text{ is optimal for (22) if and only if } v_i = 0,$$

where

$$v_i = \begin{cases} g'_p(\alpha_i) & 0 < \alpha_i < C_1 \\ g'_n(\alpha_i) & -C_1 < \alpha_i < 0 \\ \min(0, g'_n(\alpha_i)) & \alpha_i = -C_1 \\ \max(0, g'_p(\alpha_i)) & \alpha_i = C_1 \\ \max(0, g'_n(\alpha_i)) - \min(0, g'_p(\alpha_i)) & \alpha_i = 0 \end{cases} \quad (27)$$

Stopping Condition and Shrinking Strategy Based on the above discussion, we can derive their stopping condition and shrinking scheme. We follow [30] to use a similar stopping condition.

$$\|\mathbf{v}^t\|_1 < \varepsilon_s \|\mathbf{v}^0\|_1. \quad (28)$$

where \mathbf{v}^0 and \mathbf{v}^t are the initial violation and the violation in the t -th iteration, respectively. Note that \mathbf{v}^t 's components are sequentially obtained via (27) in n coordinate descent steps of the t -th iteration.

For shrinking, we remove bounded variables (i.e. $\alpha_i = 0, C_2, \forall i \notin \mathcal{I}$ or $\alpha_i = 0, C_1$ or $-C_1, \forall i \in \mathcal{I}$) if they may not be changed at the final iterations. Following [30], we use a “tighter” form of the optimality condition to conjecture that a variable may have stuck at a bound. We shrink α_i if it satisfies one of the following conditions:

For $i \notin \mathcal{I}$,

$$\alpha_i = 0 \text{ and } v_i > M, \text{ or } \alpha_i = C_2 \text{ and } v_i < -M. \quad (29)$$

and for $i \in \mathcal{J}$,

$$\alpha_i = 0 \text{ and } g'_n(\alpha_i) < -M \text{ and } g'_p(\alpha_i) > M, \quad (30a)$$

$$\alpha_i = C_1 \text{ and } g'_p(\alpha_i) < -M, \quad (30b)$$

$$\alpha_i = -C_1 \text{ and } g'_n(\alpha_i) > M, \quad (30c)$$

where

$$M = \max_{\forall i} |v_i^{t-1}| \quad (31)$$

is the maximal violation of the previous iteration.

Algorithm 2 is the overall procedure to solve (17).

Algorithm 2 DCD-2: DCD with a stopping condition and a shrinking implementation for rank k

```

1: Initial  $\boldsymbol{\alpha} := \mathbf{0}$  and the corresponding  $\mathbf{w} := \mathbf{0}$ .
2: Compute  $A_i = \mathbf{x}_i^\top \mathbf{x}_i, i = 1, \dots, n$ .
3:  $T \leftarrow \{1, \dots, n\}, M \leftarrow \infty$ 
4: for  $t = 1, 2, \dots$  do
5:   Randomly permute  $T$ .
6:   for  $i \in T$  do
7:     Find  $\tilde{y}_i$  by  $\tilde{y}_i = \begin{cases} -1, & y_i \leq k \\ 1, & y_i > k \end{cases}$ 
8:      $B = \tilde{y}_i \mathbf{w}^\top \mathbf{x}_i$ 
9:     if  $y_i \neq k$ , then
10:     $G = B - 1$ 
11:     $d_i \leftarrow -G/A_i$  and find  $v_i^t$  by (21)
12:    if the condition in (29) is satisfied, then
13:       $T \leftarrow T \setminus \{i\}$ , Continue
14:    else
15:       $\bar{\alpha}_i \leftarrow \alpha_i, \alpha_i \leftarrow \min(\max(\alpha_i + d_i, 0), C_2)$ 
16:    end if
17:    else if  $y_i = k$ , then
18:       $G_p = B + \epsilon$  and  $G_n = B - \epsilon$ 
19:      Find  $d_i$  by (24) and find  $v_i^t$  by (27)
20:      if any condition in (30) is satisfied, then
21:         $T \leftarrow T \setminus \{i\}$ , Continue
22:      else
23:         $\bar{\alpha}_i \leftarrow \alpha_i, \alpha_i \leftarrow \min(\max(\alpha_i + d_i, -C_1), C_1)$ 
24:      end if
25:    end if
26:     $\mathbf{w} \leftarrow \mathbf{w} + (\alpha_i - \bar{\alpha}_i) \tilde{y}_i \mathbf{x}_i$ ,
27:    if  $\|\mathbf{v}^t\|_1 < \epsilon_s \|\mathbf{v}^0\|_1$  and  $|T| < n$  then
28:       $T \leftarrow \{1, \dots, n\}, M \leftarrow \infty$ 
29:    end if
30:     $M \leftarrow \max_{\forall i} |v_i^t|$ 
31:  end for
32: end for

```

3.2 Constructing The Predictors Based on The Order Binary Classifications

The NPSVOR model can be regarded as an extension of the TSVM model [27] which was constructed for binary classification problem, and many studies have shown that TSVM has better performance than the standard SVM on

many binary classification problems. Its prediction function is defined as (5) based on the minimum distance principle. For a given sample, it will be assigned to the rank where the nearest hyperplane is located. However, there are still several problems with this prediction criterion for ordinal regression:

1. When these hyperplanes are not parallel, it is easy to produce ambiguous regions when predicting. At this situation, using this prediction function based on the minimum distance principle is unreasonable. It may assign the sample with a wrong label which have larger error cost.
2. In prediction, the prediction function just takes the ordinal regression as a general multiple-class classification, and doesn't consider the order information exiting in labels.

Lin and Li [15] proposed a framework to deal with the ordinal regression, that is, the ordinal regression is decomposed into a series of ordered binary classifications, and then constructed a discriminant function for each of them. Here, each discriminant function can be seen as a preference function of binary judgment problem "Is it better than the rank k ?" and then based on these binary judgment to build rank prediction. Inspired by this idea, using the approximated hyperplanes $f_k(\mathbf{x}) = \mathbf{w}_k^\top \mathbf{x} (k = 1, \dots, p)$ of p classes, we construct the following discriminant functions of $p - 1$ binary classifiers

$$g_k(\mathbf{x}) = f_k(\mathbf{x}) + f_{k+1}(\mathbf{x}) = (\mathbf{w}_k + \mathbf{w}_{k+1})^\top \mathbf{x}, \quad (32)$$

where $k = 1, \dots, p - 1$, and proposed the following *new hyperplane ranker*,

$$r_{\text{new}} : r(\mathbf{x}) = 1 + \sum_{i=1}^{p-1} I(g_i(\mathbf{x}) > 0) \quad (33)$$

where $I(\cdot)$ is the indicator function. One of the advantages of this prediction function (33) is that it can reduce the ambiguous region. Since the prediction is based on the ordered binary decomposition, the ordered structure information among labels is considered and can make the predicted label closer to the true label.

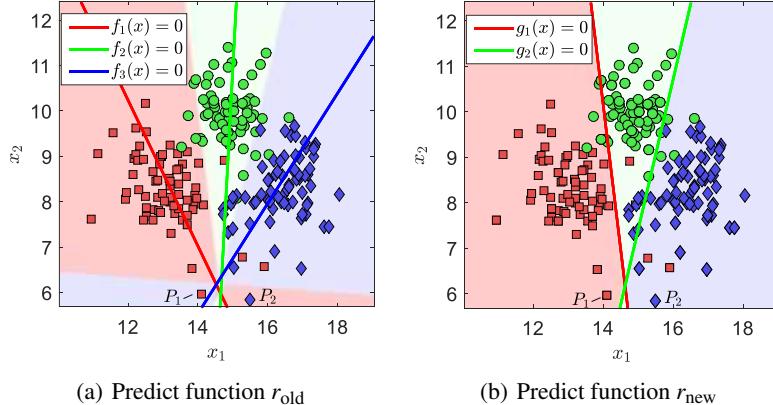


Figure 3: Demo shows the difference of NPSVOR's two predict functions r_{new} and r_{old} . The points with square, circle and diamond shape stand for rank 1, rank 2 and rank 3, respectively. The decision regions is filled with different colors (rank 1(red),rank 2(green) and rank 3(blue)).

See Figure 3, the old predict function (5) obviously produces a large ambiguous region, but it is eliminated by the new proposed predict function (33). Furthermore, two points P_1 and P_2 belonged to class 1 and class 3 are predicted incorrectly to rank 3 and rank 1 by the old predict function, which bring a large relative error. However, they are predicted rightly by the new prediction method.

4 Numerical Experiments

In this section, we evaluate the proposed algorithm described in Sections 3 and compare it with other state-of-the-art SVM-based ordinal regression models. Numerical experiments are carried out to test the performance of NPSVOR. All the implementations are in C++ based on LIBLINEAR and the experiments are conducted on a 64-bit machine with Intel Xeon 2.0GHz CPU (E5504), 4MB cache, and 8GB main memory. The programs used for our experiment is available at <https://github.com/huadong2014/LinearNPSVOR/>. Two evaluation criteria are utilized to

evaluate the prediction error between the predicted ordinal scales $\{\hat{y}_1, \dots, \hat{y}_n\}$ and the true targets $\{y_1, \dots, y_n\}$: Mean Absolute Error (MAE), which is computed as

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|. \quad (34)$$

and Mean Squared Error (MSE), which is computed as

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2. \quad (35)$$

which is widely used to evaluate the divergence between predicted sentiment labels and the ground truth labels [35, 36].

4.1 Datasets and experimental settings

We consider the following datasets in our experiments:

- **TripAdvisor**⁵, is a dataset about hotel reviews originally used in [37]. Each review is scored on a scale ranging from one star to five stars.
- **YelpReviews** [38], is obtained from the Yelp Dataset Challenge in 2015⁶. One task of this Challenge is to predict the full number of stars the user has given.
- **Treebank**⁷: Stanford Sentiment Treebank, is a classification task consisting of sentences extracted from movie reviews with human annotations of their sentiment. Train/dev/test splits are provided and each sentence is rated with a fine-grained label among five sentiment-level scores(*very negative, negative, neutral, positive, very positive*).
- **MovieReview** [39], a collection of movie reviews whose numerical rating converted from a four-star system.⁸.
- **LargeMovie**⁹, a movie review dataset for eight-class sentiment classification which contains substantially more data than the previous benchmark datasets. It provides a set of 25,000 highly polar movie reviews for training, and 25,000 reviews for testing. We combine the training dataset and testing dataset together in our experiment.
- **DoctorQuality**, a Chinese E-health dataset, we collected it from a doctor consulting APP which provides healthcare Q&A service under the mobile platform. Patients can give the description about their diseases and this App will assign a doctor to answer and chat with the patients. This process proceeds in a text dialogue. Each patient can assess the doctor after their consultation. The assessment is given at five levels: *Strong bad, Very bad, bad, satisfied and Very satisfied*. In order to avoid the influence of extreme imbalance data, about 20k instances were extracted for each category. Jieba¹⁰ were used for word segmentation.
- **Eight Amazon product datasets**, collected from two data resources: four datasets (**AmazonMp3, Video, Mobilephone, Camera**¹¹) are originally used in [37], and another four datasets ((**Electronics, HealthCare, AppsAndroid, HomeKitchen**)) [40, 41] are downloaded from Amazon product data¹². Each review consists of a review raw text with a overall rating score(1-5 stars). We choose the different types of product review datasets with different scales for our experiments. Since their rating distributions are very imbalanced, we down sample a balanced dataset from each category.

For all these English datasets, the following procedures are executed when preprocessing: stemming, stop word removal and omitting the words that occur less than three times, the term document frequency is larger than 50% or is shorter than 2 in length. TF-IDF is used to extract text features from unigrams and bigrams. To examine the testing performance, we use a stratified selection to split each data set to 70% training and 30% testing. The number of instances, features, nonzero elements in training data, and our proposed measures tested on twenty-one datasets are reported in Table 1. To estimate the test performances MAE and MSE, a stratified selection is taken to split each data set into one fifth for testing and the rest for training.

⁵The dataset is available at <http://www.cs.virginia.edu/~hw5x/dataset.html>

⁶https://www.yelp.com/dataset_challenge

⁷<http://nlp.stanford.edu/sentiment/>

⁸scale dataset v1.0: <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

⁹<http://ai.stanford.edu/~amaas/data/sentiment/>

¹⁰Jieba is an open-source Chinese word segmentation tool which is available at <https://github.com/fxsjy/jieba>

¹¹<http://sifaka.cs.uiuc.edu/~wang296/Data/index.html>

¹²Amazon product reviews datasets: <http://jmcauley.ucsd.edu/data/amazon/>

Table 1: Data statistics. n and m denote the numbers of instances and features in a data set, respectively. Nonzeros indicates the number of non-zero entries.

Datasets	Instances(n)	Features(m)	Nonzeros	Classes	Distribution
AmazonMp3	10391	65487	1004435	5	≈ 2078
Video	22281	119793	1754092	5	≈ 4456
Tablets	35166	201061	3095663	5	≈ 7033
Mobilephone	69891	265432	5041894	5	≈ 13978
Cameras	138011	654268	14308676	5	≈ 27602
TripAdvisor	65326	404778	8687561	5	≈ 13065
DoctorQuality	119879	325448	7265259	5	≈ 23975
Treebank	11856	8569	98883	5	≈ 2371
MovieReview	5007	55020	961379	4	≈ 1251
YelpReview	1121671	3138663	102232013	5	≈ 224334
LargeMovie	50000	309362	6588192	8	≈ 6250
Electronics	409041	1422181	37303259	5	≈ 81808
HealthCare	82251	283542	5201794	5	≈ 16450
AppsAndroid	220566	253932	6602522	5	≈ 44113
HomeKitchen	120856	427558	8473465	5	≈ 24171

4.2 The Performance of NPSVOR

The parameter C in NPSVOR is chosen in the range $\{2^{-5}, 2^{-4}, \dots, 2^5\}$ by five-fold cross validation (CV) on the training set. The MAE was selected for parameter tuning. After obtaining the optimal parameter C of each dataset, we conduct training set with this C and then obtain MAE/MSE on corresponding test sets.

The trained model by using the best parameter C of each data set running on the whole training set under the best C is then applied to obtain MAE/MSE on its corresponding test set. ϵ in NPSVOR is fixed at 0.1. Since the results for all datasets are similar, we only show the results of the top eight datasets in Table 1 in this section.

4.2.1 The Performance of NPSVOR on Two Prediction Functions

NPSVOR is viewed as an extension of TSVM for ordinal regression. [26] defined the prediction criterion as shown in (5) which is based on the distance from the sample to the corresponding hyperplanes, i.e. the rank of nearest category is assigned to the label. However, this prediction criterion is originally proposed for standard (multi-)classification problems. But for OR problems, the order information contained in labels has not been considered. Inspired by [15], we propose a new prediction criterion as (33). The p hyperplanes are learned for each class, and the label of a new instance is predicted by an ensemble of these ordered binary predictors. In order to verify the effectiveness of the new prediction function, we compare NPSVOR under the two prediction criteria on the above datasets. The experimental results are shown in Table 2.

From the Table 2, we can see that the performance of linear NPSVOR under the new predictor (33) is obviously better than that using old one (5) on MAE and MSE.

4.2.2 Effects of Different Parameters C_1 and C_2

In previous experiments, we set the parameters C_1 and C_2 to an equal value for the NPSVOR. Now, we try to evaluate the performance when parameters C_1 and C_2 are set to different values. We still carry out our experiments on the above datasets. ϵ is fixed at 0.1. Grid search for C_1 and C_2 are tuned in the range $\log_2 C_1, \log_2 C_2 \in [-8, -7, \dots, 5]$ and 5-fold cross validation are conducted. The results of MSE and MAE are given in Table 4. Compared to the results obtained from setting C_1 and C_2 equally, tuning these two parameters respectively makes some improvement, but this improvement is not obvious. Therefore, $C_1 = C_2$ is suggested for NPSVOR, so that it can reduce the time of tuning model parameters.

Table 2: Comparison of NPSVOR with the new prediction function (r_{new}) and the old prediction function (r_{old}). Best results are boldfaced.

Datasets	MAE		MSE	
	r_{old}	r_{new}	r_{old}	r_{new}
AmazonMp3	0.511	0.508	0.720	0.707
Video	0.401	0.390	0.543	0.541
Tablets	0.302	0.299	0.412	0.406
Mobilephone	0.410	0.404	0.548	0.554
Cameras	0.249	0.247	0.328	0.322
TripAdvisor	0.366	0.364	0.452	0.447
DoctorQuality	0.731	0.727	1.304	1.292
Treebank	0.830	0.827	1.242	1.230
MovieReview	0.424	0.424	0.452	0.452
YelpReview	0.382	0.379	0.475	0.466
LargeMovie	1.046	1.022	2.274	2.163
Electronics	0.551	0.548	0.775	0.767
HealthCare	0.617	0.612	0.954	0.941
AppsAndroid	0.598	0.594	0.879	0.868
HomeKitchen	0.543	0.538	0.766	0.780

4.2.3 Comparison of Two DCD Algorithms

Our first experiment is to compare two DCD implementations (Algorithms 4 and 2, called DCD-1 and DCD-2) so that only the better one is used for subsequent analysis. For this comparison, we consider NPSVOR with $C_1 = C_2 = 1$ and $\varepsilon = 0.1$. Because the results for all data sets are similar, we only present the results for rank 3 of top four datasets in Figure 4. The x -axis is the training time, and the y -axis is the relative difference to the dual optimal function value.

$$\frac{f(\boldsymbol{\alpha}_k) - f(\boldsymbol{\alpha}^*)}{|f(\boldsymbol{\alpha}^*)|}. \quad (36)$$

where $\boldsymbol{\alpha}^*$ is the optimum solution. We run optimization algorithms long enough to get an approximate $f(\boldsymbol{\alpha}^*)$. Results

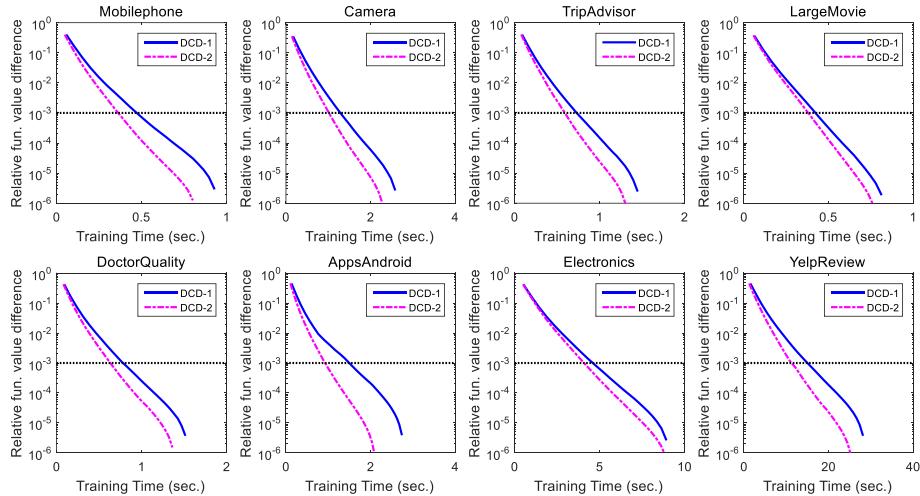


Figure 4: A comparison between two DCD algorithms (DCD-1 and DCD-2). We only present the training time and relative difference to the dual optimal function values for rank 3. NPSVOR with $C = 1$ and $\varepsilon = 0.1$ is used. Both x -axis and y -axis are in log scale.

show that DCD-2 is significantly faster than DCD-1. We notice that the running time in Figure 4 is log-scaled. This

observation is consistent with our discussion in Section 3.1 that using DCD-1 directly suffers from some redundant operations.

4.2.4 Role of The Parameter ε

If $\varepsilon=0$, linear NPSVOR can be reduced to

$$\frac{1}{2} \|\mathbf{w}_k\|_2^2 + C_1 \sum_{i \in \mathcal{I}_k} |\mathbf{w}_k^\top \mathbf{x}_i| + C_2 \sum_{i \notin \mathcal{I}_k} \max\{0, 1 - \tilde{y}_{ki} \mathbf{w}_k^\top \mathbf{x}_i\}. \quad (37)$$

The model becomes more concise. We are interested in the necessity of ε -insensitive loss function. DCD implementations can be applied to the situation of $\varepsilon = 0$, so we conduct a comparison using DCD in Figure 5. In order to verify the necessity of the parameter ε , we show the following four measures: MAE, MSE, cross validation time and the number of support vectors (SVs) with respect to ε in $[0, 0.5]$. We take four datasets (Mobilephone, Cammeras, TripAdvisor and Large Movie) to implement. During the experiment, 5-fold cross validation and grid search are used to get the optimal values of two penalty parameters C_1, C_2 from $[2^{-5}, \dots, 2^5]$ (set equal in experiments). We take $\varepsilon = 0$ as the baseline and just show their changes of the relative values of dual objective function:

$$\begin{aligned} & \text{MAE}(\varepsilon)/\text{MAE}(\varepsilon = 0), \text{MSE}(\varepsilon)/\text{MSE}(\varepsilon = 0), \\ & \text{Time}(\varepsilon)/\text{Time}(\varepsilon = 0), \text{nSVs}(\varepsilon)/\text{nSVs}(\varepsilon = 0) \end{aligned}$$

As a reference, we draw a horizontal dotted line to indicate the relative difference 10^{-3} . For the comparison between NPSVOR with and without ε , Figure 5 indicates that NPSVOR with ε can give a better MSE, faster running time and a more sparse solution. In fact, from the construction of NPSVOR, the ε insensitive loss helps to bring sparsity to the model's solution. The efficiency of DCD algorithm actually depends on the sparsity of the solution, the larger the value of ε is, the sparser the solution is, and it runs faster. From Figure 5, the relative values of running time on four datasets are decreasing with respect to ε changing from 0 to 0.5. Considering the performance of MAE on the four datasets, three of them decrease first and then increase, getting the minimum value near $\varepsilon = 0.1$. But for the LargeMovie, the ε in a proper interval can improve the value of MAE significantly, and the optimal value shows up at 0.2. Therefore, for these data sets, ε -insensitive loss function is useful and with ε , time for parameter selection can be reduced.

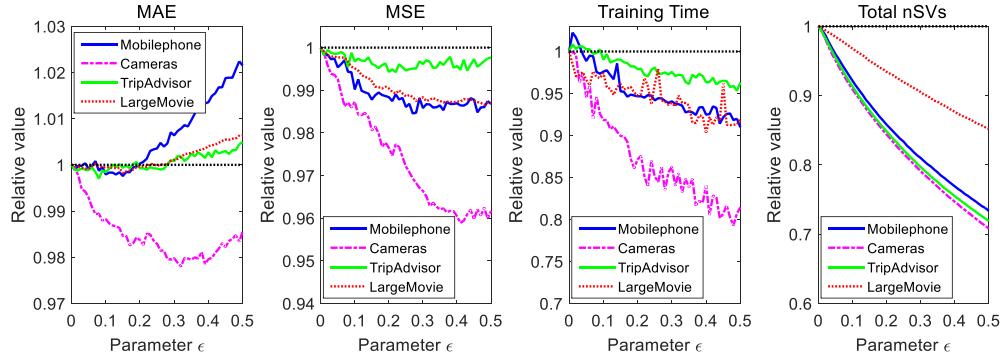


Figure 5: A comparison between NPSVOR with/without using ε . Show the MAE, MSE, Training time and the total number of support vectors(nSVs) of the model with ε relative to it fix $\varepsilon = 0$ on four datasets (Mobilephone, Cameras, TripAdvisor and LargeMovie)

4.3 Comparison of NPSVOR with Other Methods

[42] is an outstanding review which contains comparisons among the existing state-of-the-art ordinal regression methods, like naive approach, threshold approach and ordered decomposition. The experiments showed that the SVM-based methods had a promising and better performance. In this experiment, four state-of-the-art SVM-based methods are selected for comparison.

We compare the generalization performance of the linear NPSVOR with the following four state-of-art SVM-based methods:

1. SVC [23], a naive approach which treats the OR problem as a general multi classification problem. [23] provides a DCD algorithm for solving linear SVC, and takes two important accelerated strategies, the shrinking

and random permutation. The algorithm can be founded in LIBLINEAR software package, using one vs. all strategy to deal with multi-classification problems.

2. **SVR** [33]: The considered class of support vector regression was provided by the software implementation LIBLINEAR 3.21. The ordinal targets were treated as continuous values in standard SVR, and the predictions for test cases were rounded to the nearest ordinal scale. The insensitive zone parameter ϵ of SVR was fixed at 0.1.
3. **SVOR** (Support Vector Ordinal Regression with Implicit Constraints of the ordered thresholds [1]), which targets on finding the fixed max margin between each two nested classes. This method belongs to the threshold approach, where a real value mapping and several ordered thresholds were learned. Until now, there are still no large-scale algorithm for this method and just exists a nonlinear version.
4. **RedSVM** (Reduced to SVM) [15], a threshold approach to finding $(p - 1)$ thresholds that divide the real-valued line into p consecutive intervals which corresponds to the p ranks. [15] proposed a reduction framework which reduces the ordinal regression to binary classification.

In order to analyze the performance of these methods in different scale datasets and their differences between the linear and nonlinear cases in dealing with high dimensional sparse problems, two experiments are carried out as following:

4.3.1 Comparison of Linear/Nonlinear Models on Small Datasets

Some models still have no implementation for large-scale cases, such as SVOR [1] and RedSVM [15], which only have their nonlinear versions (kernelized dual models solved by SMO), we need to check whether those models with linear cases could give a competitive performance compared to their nonlinear cases for some applications, and could also enjoy faster training.

For all the nonlinear models, we consider RBF kernel $\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp(-\gamma\|\mathbf{x}_1 - \mathbf{x}_2\|^2)$, where γ is a user-specified parameter. Because the training of nonlinear SVM-based models is time consuming, we only use a subset of each dataset for training. The number of each subset is sampled less than 10000 from each dataset with a certain proportion as shown in the first column of Table 3. All subsets are divided into two parts, the 70% of data for training and the rest for testing. We conduct five-fold cross validation (CV) to find the best $C \in \{2^{-5}, \dots, 2^5\}$ and $\gamma \in \{2^{-8}, 2^{-7}, \dots, 2^0\}$. We denote the NPSVOR model with linear kernel as Lin-NPSVOR and denote it with the nonlinear kernel (RBF) as NPSVOR. The MAE/MSE are reported in Table 3. The average ranking of each method on MAE/MSE are also presented in the last row of the table.

Table 3: Test MAE and MSE results on the testing set for each small dataset and methods, including the results of these five nonlinear svm-based models. Boldface the best results.

Datasets	MAE						MSE					
	SVC	SVR	SVOR	REDSVM	NPSVOR	Lin-NPSVOR	SVC	SVR	SVOR	REDSVM	NPSVOR	Lin-NPSVOR
AmazonMp3(10%)	0.803	0.800	0.761	0.761	0.745	0.748	1.501	1.125	1.127	1.127	1.073	1.083
Video(10%)	0.679	0.737	0.674	0.674	0.651	0.647	1.115	0.993	0.896	0.895	0.875	0.872
Tablets(10%)	0.582	0.666	0.580	0.579	0.563	0.564	0.916	0.833	0.746	0.743	0.727	0.727
Mobilephone(10%)	0.620	0.697	0.624	0.623	0.607	0.607	1.009	0.919	0.827	0.829	0.809	0.812
Cameras(1%)	0.752	0.815	0.728	0.729	0.738	0.739	1.251	1.094	0.955	0.956	0.982	0.988
TripAdvisor(10%)	0.515	0.620	0.563	0.563	0.518	0.521	0.757	0.755	0.698	0.699	0.640	0.639
DoctorQuality(10%)	1.240	1.086	1.082	1.080	1.059	1.061	2.701	1.780	1.779	1.777	1.765	1.774
Treebank	1.009	0.933	0.924	0.976	0.928	0.930	1.858	1.421	1.380	1.740	1.441	1.462
MovieReview	0.616	0.569	0.560	0.558	0.562	0.560	0.758	0.627	0.618	0.616	0.629	0.627
YelpReview(0.1%)	0.747	0.946	0.892	0.870	0.791	0.792	1.381	1.404	1.290	1.250	1.063	1.069
LargeMovie(10%)	1.360	1.374	1.286	1.286	1.198	1.198	4.462	3.051	2.752	2.752	2.618	2.616
Electronics(1%)	0.899	0.892	0.840	0.841	0.816	0.819	1.779	1.305	1.256	1.257	1.187	1.198
HealthCare(10%)	0.866	0.866	0.820	0.820	0.801	0.801	1.735	1.274	1.285	1.285	1.233	1.260
AppsAndroid(1%)	0.909	0.891	0.824	0.824	0.832	0.835	1.758	1.286	1.224	1.224	1.212	1.224
HomeKitchen(1%)	0.901	1.022	0.970	0.971	0.914	0.900	1.736	1.557	1.465	1.466	1.342	1.328
Avg. Ranking	4.600	5.467	3.200	3.267	1.933	2.267	5.933	4.467	3.067	3.267	1.800	2.067

4.3.2 Comparison of Linear Methods on Large Datasets

Since no large-scale algorithm has been proposed for linear RedSVM, we modify the DCD algorithm which is originally designed for linear SVM to solve this model (RedSVM can be reduced to a binary classification based on SVC by extending the samples [15]). The parameter C is chosen in range $\{2^{-5}, 2^{-4}, \dots, 2^5\}$ by five-fold cross validation (CV)

on the training set. Using models trained under the best C , we conduct prediction on the testing set to obtain the testing MAE and MSE. We use stopping tolerance 0.1 for SVC, RedSVM and NPSVOR methods although their stopping conditions are slightly different, and use the default stopping tolerance 0.01 for SVR in LIBLINEAR. ϵ in NPSVOR and SVR is fixed to the value of 0.1. MAE was used for tuning the parameter(s).

Our code supports a bias term in a common way by appending one more feature to each data instance. We apply it to all the data sets, just compare the MAE and MSE values with the bias term. With the stopping tolerance $\epsilon = 0.1$, the results in Table 4 show that the values obtained with bias term are similar for almost all the data sets. The average ranks of each methods on MAE and MSE are presented in the last row of Table 4. We also provide the training time of those models in Table 5 which is running on training set under the best parameter C tuned on each dataset.

Table 4: The MAE/MSE on 15 datasets of four methods in their linear cases. Boldface the best results.

Datasets	MAE				MSE			
	SVC	SVR	RedSVM	NPSVOR	SVC	SVR	RedSVM	NPSVOR
AmazonMp3	0.595	0.561	0.556	0.508	1.095	0.757	0.755	0.707
Video	0.436	0.445	0.464	0.390	0.756	0.591	0.620	0.541
Tablets	0.325	0.344	0.363	0.299	0.551	0.447	0.472	0.406
Mobilephone	0.442	0.463	0.464	0.404	0.746	0.607	0.608	0.554
Cameras	0.269	0.296	0.290	0.247	0.435	0.380	0.378	0.322
TripAdvisor	0.405	0.443	0.429	0.364	0.625	0.533	0.521	0.447
DoctorQuality	0.793	0.860	0.803	0.727	1.763	1.348	1.374	1.292
Treebank	0.937	0.823	0.803	0.827	1.701	1.175	1.159	1.230
MovieReview	0.474	0.449	0.429	0.424	0.565	0.465	0.454	0.452
YelpReview	0.462	0.469	0.462	0.379	0.559	0.580	0.559	0.466
LargeMovie	1.213	1.213	1.130	1.022	3.651	2.531	2.368	2.163
Electronics	0.611	0.619	0.632	0.548	1.112	0.860	0.866	0.767
HealthCare	0.645	0.680	0.695	0.612	1.333	1.032	1.058	0.941
AppsAndroid	0.656	0.667	0.665	0.594	1.220	0.929	0.936	0.868
HomeKitchen	0.613	0.599	0.633	0.538	1.126	0.831	0.866	0.780
Avg. Ranking	2.533	3.267	3.000	1.133	3.867	2.467	2.467	1.133

Table 5: Training time of five methods under the best parameter(s).

Datasets	SVC	SVR	RedSVM	NPSVOR
AmazonMp3	0.167	0.043	0.205	0.128
Video	0.544	0.085	0.372	0.270
Tablets	1.307	0.162	0.664	0.496
Mobilephone	2.305	0.333	1.544	1.103
Cameras	11.854	1.193	5.348	3.352
TripAdvisor	4.488	0.511	1.892	1.257
DoctorQuality	3.135	0.289	1.612	2.286
Treebank	0.012	0.006	0.010	0.061
MovieReview	0.077	0.025	0.065	0.074
YelpReview	57.501	16.044	28.862	32.389
LargeMovie	5.815	0.283	0.952	1.362
Electronics	34.508	4.746	10.462	11.319
HealthCare	2.840	0.470	1.603	0.945
AppsAndroid	6.013	0.493	1.757	1.998
HomeKitchen	5.446	0.833	2.205	2.422

From the Table 4 and Table 5, we can observe the following results:

1. The NPSVOR almost achieves the best overall results compared to the SVC, SVR and RedSVM on the MAE and MSE. It shows that the prediction result of NPSVOR is closer to the true label and this method reduces the possibility of a larger prediction deviation.
2. The performance of linear RedSVM is inferior to the RedSVM of the non-linear case [1]. However, from the experimental results, its performance is rather commonplace and even inferior to the standard SVC on some data sets. The underlying assumption of threshold methods is that all classes are well ordered in a unique direction and separable by parallel hyperspaces in the primal feature space. This assumption could not be always satisfied without using nonlinear feature mapping.
3. Both NPSVOR and SVC train an independent optimization model for each category, but with the same termination accuracy, NPSVOR has a faster training speed than SVC. Because they use different shrinking strategies and termination conditions in their algorithm design. In these methods, SVR achieves the fastest training speed because it only needs to solve an optimization model of the same size as the SVC subproblem.

Ordinal regression is different from the normal multiclass classification, it needs the predicted label as close as possible to the actual label. To show the difference between the prediction of the four models, we visualize the confusion matrix of the prediction results. Figure 6 shows the confusion matrixes of the four methods on the datasets in Table 1, where the rows refer to the real ratings and the columns refer to the assigned ranks. We only show two datasets *Mobilephone* and *Camera* for discussion here; other datasets exhibit a similar results.

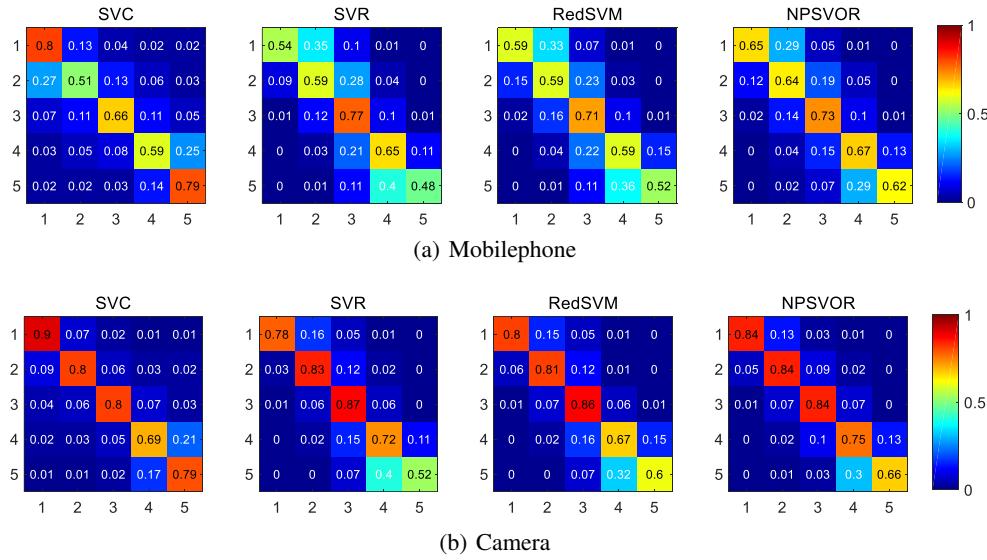


Figure 6: Confusion matrices of four methods under two datasets (MobilePhone and Camera)

For the dataset “Mobilephone”, SVC assigns the samples with rank 1 wrongly to rank 5 with an error rate of 2%, while NPSVOR does not assign any samples of rank 1 to rank 5. Since the order information existing among labels is considered in modeling, the performance of SVR, RedSVM and NPSVOR are in line with expectations, i.e. the predicted label is closer to the true label than SVC. Compared with SVC, although both SVR and RedSVM improve the MAE/MSE with a certain extent, they get a lower prediction accuracy. NPSVOR achieves a relatively higher prediction accuracy. From the diagonal of each matrix, the prediction accuracy of NPSVOR for each category is significantly higher than SVC on intermediate ranks (rank 2,3,4), and SVC only gets higher accuracy at both end sides (rank 1 and rank 5).

4.4 Sensitivity of Parameter C

To observe the sensitivity of the parameter C in models, we compare the four methods (SVC, SVR, RedSVM and NPSVOR) on the prediction performance and the training time. The value of C is changed in the range $\{2^{-5}, 2^{-4}, \dots, 2^5\}$. For each value C , we train the models on each training set and then apply the trained models to the testing set. The parameter settings are the same as Section 4.3.2. Figure 7 show how the MAE change with respect to C , respectively. It can be observed that MAE of NPSVOR changes stably with C . When the value of C is larger than 0.1, the performance

of NPSVOR is obviously better than other three methods. Since the figure under MSE is similar with Figure 7, we don't discuss and show it again.

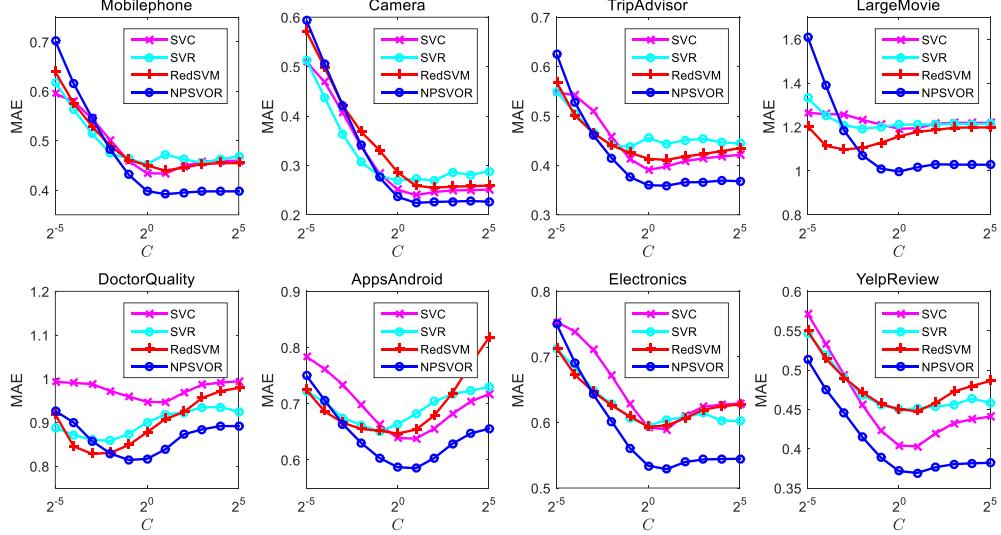


Figure 7: MAE changes with the parameter C on eight datasets

Conclusion and further works

We have studied the NPSVOR in linear case for training large-scale ordinal regression problems. A new DCD method (DCD-2) having the same efficiency with linear SVC is proposed to solve a reformulation of the dual problem of NPSVOR. It utilizes the structural relationship of the solution of the dual model and then reformulates the dual model with only n variables. Empirical comparisons show that the training time of DCD-2 reduces obviously when compared to DCD-1. The proposed method can train linear NPSVOR efficiently and achieve comparable performances to other state-of-the-art methods. To further utilize the ordered information contained in labels, a new predict function r_{new} is defined, which is obtained by $p - 1$ ordered binary discrimination functions constructed by p learned hyperplanes. Experiments show that r_{new} is better than the OR predictor r_{old} based on the minimum distance principle.

In addition, although we only consider the hinge loss in our implementation for the experiments because of the lack of space, our method is applicable to the square hinge loss (L2 loss), logistic loss etc with a minor change. Note that each hyperplane learned in NPSVOR is obtained independently with each other, without considering their correlation. Considering OR as a structured and interdependent output problem is our future work.

Acknowledgments

This work is partially supported in part by the National Science Foundation of China under Grant No.11671379, No.11331012, No.71331005, No.91546201.

References

- [1] Wei Chu and S Sathiya Keerthi. Support vector ordinal regression. *Neural computation*, 19(3):792–815, 2007.
- [2] Maria J. Cardoso Jaime S. Cardosoa, Joaquim F. Pinto da Costab. Modelling ordinal relations with svms: An application to objective aesthetic evaluation of breast cancer conservative treatment. *Neural Networks*, 18(5-6):808-817, 2005.
- [3] Francisco Fernandez-Navarro, Pilar Campoy-Munoz, Monica-de la Paz-Marín, Casar Hervás-Martínez, and Xin Yao. Addressing the EU sovereign ratings using an ordinal regression approach. *Cybernetics, IEEE Transactions on*, 43(6):2228–2240, 2013.

- [4] C Hervícs-Martínez L Carro-Calvo J Sícnchez-Monedero PA Gutiérrez, S Salcedo-Sanz. Ordinal and nominal classification of wind speed from synoptic pressure patterns. *Engineering Applications of Artificial Intelligence*, 26(3):1008–1015, 2013.
- [5] M. Dyson J. W. Yoon, S. J. Roberts and J. Q. Gan. Bayesian inference for an adaptive ordered probit model: An application to brain computer interfacing. *Neural Networks*, 24(7):726–734, 2011.
- [6] Tie-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.
- [7] Jaime Carbonell Shinjae Yoo, Yiming Yang. Modeling personalized email prioritization: classification-based and regression-based approaches. *Office of Scientific and Technical Information Technical Reports*, 21(1):729–738, 2013.
- [8] A Doucet P Martin. Dating color images with ordinal classification. In *International Conference on Multimedia Retrieval*, number 447–450, 2014.
- [9] Rupesh K. Gopal and Saroj K. Meher. Customer churn time prediction in mobile telecommunication industry using ordinal regression. In *Pacific-asia Conference on Advances in Knowledge Discovery and Data Mining*, pages 884–889, 2008.
- [10] H.Yan. Cost-sensitive ordinal regression for fully automatic facial beauty assessment. *Neurocomputing*, 129:334–342, 2014.
- [11] Vladimir Pavlovic Ognjen Rudovic and Maja Pantic. Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 2634–2641, 2012.
- [12] M Pantic O Rudovic, V Pavlovic. Context-sensitive dynamic ordinal regression for intensity estimation of facial action units. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 37(5):944–958, 2015.
- [13] Chu-Song Chen Kuang-Yu Chang and Yi-Ping Hung. Ordinal hyperplanes ranker with cost sensitivities for age estimation. *IEEE Conference on Computer Vision and Pattern Recognition*, 42:585–592, 6 2011.
- [14] S Baccianella, A Esuli, and F Sebastiani. Feature selection for ordinal text classification. *Neural Computation*, 26(3):557–591, 2014.
- [15] Hsuan-Tien Lin and Ling Li. Reduction from cost-sensitive ordinal ranking to weighted binary classification. *Neural Computation*, 24(5):1329–1367, 2012.
- [16] Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. In *Journal of Machine Learning Research*, pages 1019–1041, 2005.
- [17] Bing-Yu Sun, Jiuyong Li, Desheng Dash Wu, Xiao-Ming Zhang, and Wen-Bo Li. Kernel discriminant learning for ordinal regression. *Knowledge and Data Engineering, IEEE Transactions on*, 22(6):906–910, 2010.
- [18] Willem Waegeman and Luc Boullart. An ensemble of weighted support vector machines for ordinal regression. *International Journal of Computer Systems Science and Engineering*, 3(1):47–51, 2009.
- [19] Rizal Fathony, Mohammad Ali Bashiri, and Brian Ziebart. Adversarial surrogate losses for ordinal regression. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 563–573. Curran Associates, Inc., 2017.
- [20] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–180, 1971.
- [21] T. Graepel R. Herbrich and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.
- [22] Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Support vector learning for ordinal regression. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN 99)*, volume 1, pages 97–102, 1999.
- [23] Cho Jui Hsieh, Kai Wei Chang, Chih Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *ICML*, pages 1369–1398, 2008.
- [24] Thorsten Joachims. Training linear svms in linear time. In *Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pages 217–226, 2006.
- [25] F Fernandez-Navarro F Fernandez-Navarro C Hervas-Martinez M Perez-Ortiz, M Perez-Ortiz. Ordinal regression methods: survey and experimental study. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 28(1):127–146, 1 2016.
- [26] Huadong Wang, Yong Shi, Lingfeng Niu, and Yingjie Tian. Nonparallel support vector ordinal regression. *IEEE Transactions on Cybernetics*, 2016.

- [27] Jayadeva, R. Khemchandani, and Suresh Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 29(5):905–910, 2007.
- [28] Yingjie Tian, Zhiqian Qi, Xuchan Ju, Yong Shi, and Xiaohui Liu. Nonparallel support vector machines for pattern classification. *Cybernetics, IEEE Transactions on*, 44(7):1067–1079, 2014.
- [29] Mingyi Hong, Xiangfeng Wang, Meisam Razaviyayn, and Zhi Quan Luo. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming*, 163(1-2):85–114, 2017.
- [30] Guo Xun Yuan, Kai Wei Chang, Cho Jui Hsieh, and Chih Jen Lin. A comparison of optimization methods and software for large-scale l1-regularized linear classification. *Journal of Machine Learning Research*, 11(2):3183–3234, 2010.
- [31] Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale l2-loss linear support vector machines. *Journal of Machine Learning Research*, 9(Jul):1369–1398, 2008.
- [32] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1):387–423, 2009.
- [33] Chia-Hua Ho and Chih-Jen Lin. Large-scale linear support vector regression. *The Journal of Machine Learning Research*, 13(1):3323–3348, 2012.
- [34] Ralph Tyrell Rockafellar. *Convex analysis*. Princeton university press, 2015.
- [35] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Conference on Empirical Methods in Natural Language Processing*, 2015.
- [36] Qiming Diao, Minghui Qiu, Chao Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 193–202, 2014.
- [37] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 783–792, 2010.
- [38] Xiang Zhang, Junbo Zhao, and Yann Lecun. Character-level convolutional networks for text classification. 2016.
- [39] Bo Pang and Lillian Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Meeting on Association for Computational Linguistics*, pages 115–124, 2005.
- [40] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.
- [41] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2015.
- [42] M Pérez-Ortiz, Manuel Cruz-Ramírez, MD Ayllón-Terán, N Heaton, R Ciria, and César Hervás-Martínez. An organ allocation system for liver transplantation based on ordinal regression. *Applied Soft Computing*, 14:88–98, 2014.