Machine Learning (2024) 113:2575–2595
https://doi.org/10.1007/s10994-024-06518-x

Tackle balancing constraints in semi-supervised ordinal regression

Chenkang Zhang[1] · Heng Huang[2] · Bin Gu[3,4]

Received: 2 April 2022 / Revised: 30 January 2024 / Accepted: 30 January 2024 /
Published online: 4 March 2024

Abstract

Semi-supervised ordinal regression (S^{2}OR) has been recognized as a valuable technique to improve the performance of the ordinal regression (OR) model by leveraging available unlabeled samples. The balancing constraint is a useful approach for semi-supervised algorithms, as it can prevent the trivial solution of classifying a large number of unlabeled examples into a few classes. However, rapid training of the S^{2}OR model with balancing constraints is still an open problem due to the difficulty in formulating and solving the corresponding optimization objective. To tackle this issue, we propose a novel form of balancing constraints and extend the traditional convex–concave procedure (CCCP) approach to solve our objective function. Additionally, we transform the convex inner loop (CIL) problem generated by the CCCP approach into a quadratic problem that resembles support vector machine, where multiple equality constraints are treated as virtual samples. As a result, we can utilize the existing fast solver to efficiently solve the CIL problem. Experimental results conducted on several benchmark and real-world datasets not only validate the effectiveness of our proposed algorithm but also demonstrate its superior performance compared to other supervised and semi-supervised algorithms

Keywords Semi-supervised learning · Ordinal regression · Balancing constraint

Editor: Joao Gama.

✉ Bin Gu
jsgubin@gmail.com

Chenkang Zhang
zhangchenkang@cmss.chinamobile.com

Heng Huang
heng@umd.edu

[1] China Mobile (Suzhou) Software Technology Company Limited, Suzhou, People's Republic of China

[2] Department of Computer Science, University of Maryland, College Park, MD, USA

[3] School of Artificial Intelligence, Jilin University, Changchun, People's Republic of China

[4] Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence, Masdar City, UAE

1 Introduction

Ordinal regression (OR) has been a subject of research for the past two decades and is commonly formulated as a multi-class problem with ordinal constraints (Berg et al., 2021; Buri & Hothorn, 2020; Garg & Manwani, 2020; Gu et al., 2023; Li et al., 2020; Pang et al., 2020). This learning task has been widely applied in various real-world scenarios such as information retrieval (Herbrich, 1999), collaborative filtering (Shashua & Levin, 2003), social sciences (Fullerton & Xu, 2012), and medical analysis (Cardoso et al., 2005). However, most of the existing ordinal regression models can only deal with labeled data, such as SVOR-IMC and SVOR-EXC (Chu & Keerthi, 2007). As a result, significant effort is required to obtain a sufficient number of labeled samples. To tackle this challenging problem, it has become necessary to incorporate unlabeled samples, which are often readily available at a low cost, into the training process (Chapelle et al., 2009). This has led to an increasing amount of attention being given to semi-supervised ordinal regression (S^{2}OR) (Ganjdanesh et al., 2020; Gu et al., 2022; Liu et al., 2011; Seah et al., 2012; Srijith et al., 2013).

In semi-supervised problems, it is crucial to prevent the trivial solution of classifying a large number of unlabeled examples into a few classes (Chapelle et al., 2008; Xu et al., 2005; Zhu, 2005). To address this issue, the balancing constraint (Collobert et al., 2006) has been proposed as an effective solution, and several semi-supervised binary classification algorithms (Chapelle & Zien, 2005; Collobert et al., 2006; Joachims, 1999) with balancing constraints have been developed over the last two decades. Joachims (1999) enforced balancing constraints by swapping the pseudo-labels of unlabeled samples, assuming that these pseudo-labels match the class distribution of the labeled samples beforehand. Meanwhile, Chapelle and Zien (2005) directly treated balancing constraints as additional constraints, utilizing the gradient descent method to optimize the objective function. Additionally, Collobert et al. (2006) viewed balancing constraints as virtual samples, training the model with these virtual samples and other regular samples. Among these methods, the virtual samples approach is particularly efficient since it can leverage existing state-of-the-art solvers for the objective function with balancing constraints.

Efficiently training S^{2}OR problems with balancing constraints remains a significant challenge due to the complexity of formulating and solving related problems. While ManifoldOR (Liu et al., 2011) and SSORERM (Tsuchiya et al., 2019) are both S^{2}OR algorithms, they do not consider the influence of balancing constraints. To date, TOR (Seah et al., 2012) and SSGPOR (Srijith et al., 2013) are the only two S^{2}OR algorithms using balancing constraints, and they achieve this by swapping the pseudo-labels of unlabeled samples. However, this approach may require a large number of iterations to make all pseudo-labels reach the optimal position. Therefore, the TOR and SSGPOR algorithms are not yet optimal S^{2}OR algorithms using balancing constraints. Table 1 summarizes representative OR algorithms, which highlights the need for a fast S^{2}OR algorithm using balancing constraints, as this remains an open question.

In this paper, we propose a new algorithm, called BC-S^{2}OR, to address the challenging problem of balancing constraints in S^{2}OR via virtual samples. Specifically, we introduce a novel form of balancing constraints for S^{2}OR to prevent most of the unlabeled samples from being classified into a few classes. Then, we extend the traditional convex-concave procedure (CCCP) framework to solve this complex optimization problem. We transform the convex inner loop (CIL) problem with multiple equality constraints into a quadratic problem like support vector machine (SVM), where the multiple equality constraints are

Table 1 Several representative OR algorithms

	Algorithm	Balancing constraints (BC)	Time complexity[1]
Supervised	SVOR-IMC (Chu & Keerthi, 2007)	No	$O(N^a)$
	SVOR-EXC (Chu & Keerthi, 2007)	No	$O(N^a)$
Semi-supervised	SSORERM (Tsuchiya et al., 2019)	No	$O(tN^3)$
	ManifoldOR (Liu et al., 2011)	No	$O(tN^3)$
	SSGPOR (Srijith et al., 2013)	Swap labels of unlabeled data	$O(TN^3)$
	TOR (Seah et al., 2012)	Swap labels of unlabeled data	$O(TN^a)$
	BC-S^{2}OR	Treat BC as virtual samples	$O(tN^a)$

[1] N denotes the size of problem, $1 < a < 2.3$, and both of T and t denote number of iterations, but T is much more than t

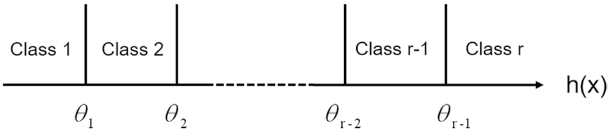


Fig. 1 Threshold model of OR problem

treated as virtual samples. This allows us to use existing solvers (Chu & Keerthi, 2007) to efficiently solve the CIL problems. Numerical experiments conducted on several benchmark and real-world datasets confirm the effectiveness of our proposed algorithm, which outperforms other supervised and semi-supervised OR algorithms.

2 Preliminaries

2.1 Ordinal regression

Ordinal regression (OR) is a significant supervised learning problem that involves learning a ranking or ordering of instances. It combines the properties of classification and metric regression. The goal of ordinal regression is to categorize data points into a set of finite ordered categories.

Let $S = \{(x_i, y_i)|i = 1, 2, \dots, n\}$ denote an OR dataset, where $x_i \in \mathbb{R}^d$ is a d-dimensional vector and $y_i \in \{1, 2, \dots, r\}$ is the corresponding ordinal class label. Based on the threshold model (Crammer & Singer, 2002) illustrated in Fig. 1, the OR problem with r classes has $r - 1$ ordered thresholds: $\theta_1 \le \theta_2 \le \cdots \le \theta_{r-1}$. In this threshold model, a sample x is classified as class k if its predictive output $h(x) = \langle w, \phi(x) \rangle$[1] falls within the range of $\theta_{k-1} < h(x) \le \theta_k$.

On the basis of this threshold model, Li and Lin (2007) proposed a new formulation of the OR problem. Specifically, they reduced the OR problem with r classes to $r - 1$ binary classification sub-problems and defined the training set of each sub-problem as:

[1] $\phi(\cdot)$ is transformation function from an input space to a high-dimensional reproducing kernel Hilbert space.

$$\begin{aligned} x_i^k &= x_i, \\ y_i^k &= 1 - 2I[y_i \leq k], \end{aligned} \tag{1}$$

where $I[a]$ denotes an indicator function that returns 1 if a is true and returns 0 if a is false, $k \in \{1, 2, \dots, r-1\}$ and $y_i^k \in \{-1, +1\}$. These $r-1$ sub-problems are aimed to define $r-1$ parallel decision boundaries for ordinal scales and the predictive ordinal class label $f(x_i)$ is defined as:

$$\begin{aligned} f(x_i) &= 1 + \sum_{k=1}^{r-1} I[g(x_i^k) > 0], \\ g(x_i^k) &= h(x_i) - \theta_k, \end{aligned} \tag{2}$$

where $h(x) = \langle w, \phi(x) \rangle$ is the predictive output, θ_k means the k-th threshold and $g(x_i^k)$ denotes the binary classifier.

2.2 Semi-supervised ordinal regression

In many cases, ordinal regression models suffer from poor performance due to a limited number of ordinal samples. To overcome this challenge, it becomes necessary to incorporate unlabeled samples into the training process. As a result, there has been a growing interest in semi-supervised ordinal regression (S^{2}OR).

For a particular labeled sample (x_i, y_i), the extended binary classification loss ζ_i^k for a particular threshold θ_k can be derived as

$$\zeta_i^k = \max\left\{0, 1 - y_i^k(w^T\phi(x_i) - \theta_k)\right\}. \tag{3}$$

Consequently, the ordinal regression loss ζ_i of the labeled sample (x_i, y_i) superimposing the $r-1$ parts easily becomes

$$\begin{aligned} \zeta_i = \sum_{k=1}^{r-1} \zeta_i^k &= \sum_{k=1}^{r-1} \max\left\{0, 1 - y_i^k(w^T\phi(x_i) - \theta_k)\right\} \\ &= \sum_{k=1}^{r-1} H_1(y_i^k g(x_i^k)) \end{aligned} \tag{4}$$

where $H_s(t) = \max\{0, s - t\}$.

Given that n labeled samples and u unlabeled samples are available in S^{2}OR, the objective function can be derived as:

$$\min_{\bar{w} \overset{\text{def}}{=} (w,\theta)} J(\bar{w}) = \frac{1}{2}\|w\|^2 + C \sum_{k=1}^{r-1} \sum_{i=1}^{n} H_1\left(y_i^k g\left(x_i^k\right)\right) \tag{5}$$

$$\begin{aligned} &+ C^* \sum_{k=1}^{r-1} \sum_{i=n+1}^{n+u} H_1(|g(x_i^k)|) \\ s.t. \quad &\theta_1 \leq \theta_2 \leq \cdots \leq \theta_{r-1}, \end{aligned} \tag{6}$$

where $\bar{w} = (w, \theta)$ is the parameter vector, C means the penalty coefficient of labeled samples and C^* means the penalty coefficient of unlabeled samples.

The threshold order defined in Eq. (6) plays a crucial role in the OR problem and (Chu & Keerthi, 2007) proposed two approaches to ensure this order. The first approach involves calculating the loss contribution of all samples for each threshold, which automatically satisfies the ordered relations between the thresholds at the optimal solution. However, in Eq. (5), the loss term of unlabeled samples causes the objective function to lose the property of convexity. Consequently, the threshold order is not guaranteed automatically, as shown in Lemma 1. The detailed proof of Lemma 1 can be found in Appendix A.

Lemma 1 *For the S^{2}OR problem based on the threshold model, if the constraint of the threshold order (i.e., $\theta_1 \le \theta_2 \le \cdots \le \theta_{r-1}$) is not explicitly enforced, the threshold order cannot be guaranteed automatically.*

Building upon the previous discussion, we adopt the second approach proposed by Chu and Keerthi (2007) to ensure the threshold order. This approach involves explicitly incorporating the constraint of the threshold order into the objective function. Although this approach may make the S^{2}OR problem slightly more complicated, it can perfectly guarantee the threshold order.

2.3 Concave–convex procedure

The concave-convex procedure (CCCP) is a highly effective majorization-minimization algorithm that is capable of solving the non-convex program formulated as the difference of convex functions (DC) through a sequence of convex programs.

We provide a specific example of the CCCP approach. Firstly, we define the DC program as:

$$\begin{aligned} \min_{w} \ & \&o(w) - v(w) \\ s.t. \quad & a_i \le 0, i \in [A], \\ & b_j = 0, j \in [B], \end{aligned} \tag{7}$$

where o, v and a_i are real-valued convex functions, b_j is an affine function, A and B respectively represent the number of inequality constraints and equality constraints. The CCCP approach is an iterative procedure that solves the following sequence of convex programs:

$$\begin{aligned} w^{t+1} = \arg\min_{w} \ & \&o(w) - w \bigtriangledown v(w^t) \\ s.t. \quad & a_i \le 0, i \in [A], \\ & b_j = 0, j \in [B], \end{aligned} \tag{8}$$

where t means the iteration number. In our paper, we refer to these convex programs as the convex inner loop (CIL) problems. As demonstrated in Eq. (8), the CCCP approach involves linearizing the concave portion (i.e., $-v(w)$) to achieve a sequence of convex programs (i.e., $o(w) - w \bigtriangledown v(w^t)$).

3 Proposed algorithm

In this section, we first introduce our local balancing constraints. Then, we propose our BC-S^{2}OR algorithm to tackle the local balancing constraints via virtual samples. Finally, we analyze the time complexity of our algorithm.

3.1 Local balancing constraints

In semi-supervised binary classification problems, a binary classifier may classify all unlabeled examples into the same class due to the large margin criterion. To address this issue, Joachims (1999) proposed a balancing constraint that ensures the proportion of different classes assigned to the unlabeled samples is the same as that found in the labeled samples. Building on this idea, Collobert et al. (2006) and Chapelle and Zien (2005) introduced a similar but slightly relaxed balancing constraint:

$$\frac{1}{u}\sum_{i=n+1}^{n+u} f(x_i) = \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{9}$$

In the S^{2}OR problem based on the threshold model, multiple binary classifiers are trained to predict the ordinal class label. However, the large margin criterion may negatively impact each extended binary classifier, resulting in poor performance of the overall S^{2}OR problem. To address this issue, we introduce multiple similar balancing constraints as the ones mentioned above to restrict each binary classifier separately. The balancing constraint formulation for the S^{2}OR problem is as follows:

$$\frac{1}{u}\sum_{i=n+1}^{n+u} g(x_i^k) = \frac{1}{n}\sum_{i=1}^{n} y_i^k, \tag{10}$$

where there are $r-1$ balancing constraints.

However, it should not be overlooked that the original balancing constraint defined in Eq. (10) is one slightly relaxed constraint and cannot perfectly guarantee that the proportion of positive and negative labels assigned to the unlabeled samples matches that found in the labeled samples in binary classification problems. The superposition of multiple errors of original balancing constraints can lead to the instability of the entire S^{2}OR problem, resulting in most unlabeled samples being classified into a few categories.

To alleviate this situation, we propose a modification to the original balancing constraint. Specifically, we reduce the weights of samples that are far away from decision boundaries and only maintain the balancing constraints of samples near the decision boundaries. This modification ensures that the proportion of each class of unlabeled samples is consistent with that of labeled samples as much as possible. We introduce a novel formulation called the local balancing constraint in our S^{2}OR, which is expressed as follows:

$$\frac{1}{Z^k}\sum_{i=n+1}^{n+u} \gamma_i^k g(x_i^k) = \frac{1}{E^k}\sum_{i=1}^{n} \gamma_i^k y_i^k, \tag{11}$$

where there are $r-1$ local balancing constraints and γ_i^k, Z^k and E^k can be defined as:

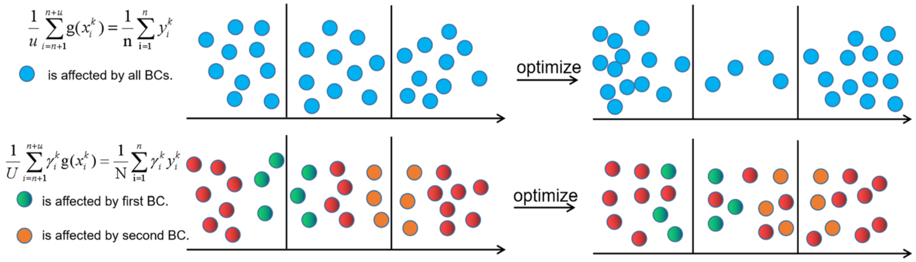

$\frac{1}{u}\sum_{i=n+1}^{n+u} \mathrm{g}(x_i^k) = \frac{1}{\mathrm{n}}\sum_{\mathrm{i=1}}^{n} y_i^k$
● is affected by all BCs.
optimize
$\frac{1}{U}\sum_{i=n+1}^{n+u} \gamma_i^k \mathrm{g}(x_i^k) = \frac{1}{\mathrm{N}}\sum_{\mathrm{i=1}}^{n} \gamma_i^k y_i^k$
● is affected by first BC.
● is affected by second BC.
optimize

Fig. 2 Contrast between original balancing constraints and our local balancing constraints

$$
\gamma_i^k = \begin{cases} 1 & \text{if } |g(x_i^k)| < c \\ 0 & \text{otherwise} \end{cases} \tag{12}
$$

$$
E^k = \sum_{i=1}^{n} I\left[\gamma_i^k \neq 0\right], Z^k = \sum_{i=n+1}^{n+u} I\left[\gamma_i^k \neq 0\right], \tag{13}
$$

where c is the constraint parameter. In practical applications, to avoid the high complexity of Eq. (11), it is common to use γ_i^k as a constant coefficient, which is calculated in advance.

The comparison between the original balancing constraints and our local balancing constraints is presented in Fig. 2 to illustrate the difference. Specifically, we use a three-class S^{2}OR problem as an example. The original balancing constraints defined in Eq. (10) are influenced by unlabeled samples far away from the decision boundaries, which may cause an incorrect reflection of the proportion of positive and negative labels of unlabeled samples in the binary classification sub-problem. Consequently, the superposition of multiple errors of original balancing constraints can result in a serious imbalance between the proportions of unlabeled samples in each class in the whole S^{2}OR problem. This imbalance can lead to the classification of only a few unlabeled samples into the second class, as shown in the first picture in Fig. 2. In contrast, our proposed local balancing constraints can ignore samples far away from the decision boundaries and better ensure the proportion of unlabeled samples in each class, as shown in the second picture in Fig. 2.

3.2 BC-S^{2}OR algorithm

In this subsection, we discuss the transformation of our non-convex objective function into a formulation of the difference of convex functions (DC) and how we utilize the CCCP approach to solve it (Allahzadeh & Daneshifar, 2021; Oliveira & Valle, 2020; Rastgar et al., 2020; Zhai et al., 2020, 2023). We specifically focus on transferring the local balancing constraints to virtual samples and then using existing SVOR solvers (Chu & Keerthi, 2007) to efficiently solve the CIL problem.

3.2.1 DC formulation

The treatment of loss terms for unlabeled samples poses a challenging problem in semi-supervised learning, particularly in the context of S^{2}OR. Given the large number of loss terms for each unlabeled sample at all thresholds, addressing this issue is a thorny task. To overcome this challenge, we propose to convert the unlabeled samples into artificially labeled ones. This conversion enables us to transform the non-convex objective function into a DC formulation, which is expressed as the difference between two convex functions.

Firstly, we duplicate unlabeled samples and introduce that

$$\begin{aligned} y_i &= r, \quad \forall i \in \{n+1, \dots, n+u\}, \\ y_i &= 1, \quad \forall i \in \{n+u+1, \dots, n+2u\}, \\ x_i &= x_{i-u}, \quad \forall i \in \{n+u+1, \dots, n+2u\}. \end{aligned} \tag{14}$$

Note that when $y_i = r$, we have that $\forall k \in \{1, \dots, r-1\}, y_i^k = +1$ according to Eq. (1), and when $y_i = 1$, we have that $\forall k \in \{1, \dots, r-1\}, y_i^k = -1$.

Then, the original S^{2}OR problem (i.e., Eq. (5)) can be equivalently rewritten as Eq. (15). The proof can be found in Appendix B.

$$\min_{\bar{w} \overset{\text{def}}{=} (w,\theta)} J(\bar{w}) = o(\bar{w}) - v(\bar{w}) \tag{15}$$

where o and v are convex functions defined as follows:

$$\begin{aligned} o(\bar{w}) &= \frac{1}{2}\|w\|^2 + C \sum_{k=1}^{r-1} \sum_{i=1}^{n} H_1\left(y_i^k g\left(x_i^k\right)\right) \\ &\quad + C^* \sum_{k=1}^{r-1} \sum_{i=n+1}^{n+2u} H_1\left(y_i^k g\left(x_i^k\right)\right), \\ v(\bar{w}) &= C^* \sum_{k=1}^{r-1} \sum_{i=n+1}^{n+2u} H_0\left(y_i^k g\left(x_i^k\right)\right). \end{aligned} \tag{16}$$

3.2.2 CCCP for S^{2}OR

We utilize the CCCP approach to solve the DC formulation (i.e., Eq. (15)). As shown in Eq. (8), the main process of the CCCP approach is to iteratively solve a sequence of CIL problems, which is generated by linearizing the concave part of the original DC formulation. Here, we first calculate the derivative of the concave part with respect to $\bar{w}$

$$-\bar{w} \cdot \nabla v(\bar{w}) = \sum_{k=1}^{r-1} \sum_{i=n+1}^{n+2u} \beta_i^k y_i^k g\left(x_i^k\right) \tag{17}$$

$$\text{where} \quad \beta_i^k = \begin{cases} C^* & \text{if } y_i^k g\left(x_i^k\right) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Then, according Eq. (8), we obtain the following CIL problem:

$$
\begin{aligned}
\min_{\bar{w} \overset{\text{def}}{=} (w,\theta)} \quad & \frac{1}{2}\|w\|^2 + C\sum_{k=1}^{r-1}\sum_{i=1}^{n}\xi_i^k + C^*\sum_{k=1}^{r-1}\sum_{i=n+1}^{n+2u}\xi_i^k \\
& + \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+2u}\beta_i^k y_i^k g\left(x_i^k\right) \\
s.t. \quad & \frac{1}{Z^k}\sum_{i=n+1}^{n+u}\gamma_i^k g(x_i^k) = \frac{1}{E^k}\sum_{i=1}^{n}\gamma_i^k y_i^k, \\
& y_i^k g(x_i^k) \geq 1 - \xi_i^k, \xi_i^k \geq 0, \quad \forall i \in \{1, \dots, n+2u\}, \\
& \theta_1 \leq \theta_2 \leq \cdots \leq \theta_{r-1}.
\end{aligned}
\tag{19}
$$

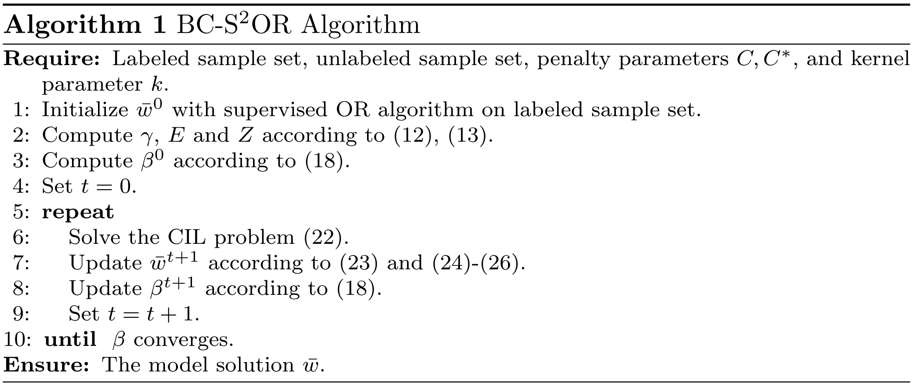
To solve the aforementioned CIL problem, we introduce Lagrangian variables (Bertsekas, 2014) and calculate the partial derivative with respect to primal variables. In order to simplify the dual problem, we transform our local balancing constraints to virtual samples. It should be noted that the number of virtual samples is the same as the number of local balancing constraints, both of which are $r-1$. The virtual samples are defined as follows:

$$
\phi(x_0^k) = \frac{1}{Z^k}\sum_{i=n+1}^{n+u}\gamma_i^k \phi\left(x_i^k\right), \;\; y_0^k = 1,
\tag{20}
$$

where $\beta_0^k = 0$. The column in kernel matrix corresponding to the example x_0^k is computed as follow:

$$
\left\langle \phi\left(x_0^k\right), \phi\left(x_i^{k'}\right)\right\rangle = \frac{1}{Z^k}\sum_{j=n+1}^{n+u}\gamma_j^k\left\langle \phi\left(x_j^k\right), \phi\left(x_i^{k'}\right)\right\rangle.
\tag{21}
$$

Algorithm 1 BC-S^{2}OR algorithm



Algorithm 1 BC-S^{2}OR Algorithm

```
Require: Labeled sample set, unlabeled sample set, penalty parameters $C, C^*$, and kernel parameter $k$.
Initialize $\bar{w}^0$ with supervised OR algorithm on labeled sample set.
Compute $\gamma$, $E$ and $Z$ according to (12), (13).
Compute $\beta^0$ according to (18).
Set $t = 0$.
repeat
    Solve the CIL problem (22).
    Update $\bar{w}^{t+1}$ according to (23) and (24)-(26).
    Update $\beta^{t+1}$ according to (18).
    Set $t = t + 1$.
until $\beta$ converges.
Ensure: The model solution $\bar{w}$.
```

Here, we directly show the final duality formulation of the CIL problem:

$$
\begin{aligned}
&\arg\max_{\bar{\alpha}} \sum_{k=1}^{r-1}\sum_{i=0}^{n+2u} \varsigma_i^k \bar{\alpha}_i^k \\
&-\frac{1}{2}\sum_{k=1}^{r-1}\sum_{k'=1}^{r-1}\sum_{i=0}^{n+2u}\sum_{j=0}^{n+2u} \bar{\alpha}_i^k \bar{\alpha}_j^{k'} K\left(x_i^k, x_j^{k'}\right) \\
&s.t.\ \underline{C}_i^k \le \bar{\alpha}_i^k \le \overline{C}_i^k, \sum_{i=0}^{n+2u} \bar{\alpha}_i^k - s^k + s^{k+1} = 0, s^k \ge 0.
\end{aligned}
\tag{22}
$$

where $\varsigma_0^k = \frac{1}{E^k}\sum_{i=1}^n \gamma_i^k y_i^k$ and $\varsigma_i^k = y_i^k$ when $i \neq 0$, $s^k = 0$ when $k = 0$ or $k = r$, $K(x_i^k, x_j^{k'}) = \langle \phi(x_i^k), \phi(x_j^{k'}) \rangle$, $\bar{\alpha}_i^k = (\alpha_i^k - \beta_i^k) y_i^k$, and $\underline{C}_i^k$ and $\overline{C}_i^k$ are defined as:

$$
\underline{C}_i^k = \begin{cases} \min\left\{y_i^k C, 0\right\} & \text{if } 1 \le i \le n \\ \min\left\{y_i^k\left(C^* - \beta_i^k\right), -y_i^k \beta_i^k\right\} & \text{if } i \ge n+1 \end{cases} ll
$$

$$
\overline{C}_i^k = \begin{cases} \max\left\{y_i^k C, 0\right\} & \text{if } 1 \le i \le n \\ \max\left\{y_i^k\left(C^* - \beta_i^k\right), -y_i^k \beta_i^k\right\} & \text{if } i \ge n+1 \end{cases}
$$

Equation (22) is close to the SVOR optimization problem and thus can be optimized by the standard SVOR solver (Chu & Keerthi, 2005, 2007).

When using the Lagrange multiplier method, w is formed as:

$$
w = \sum_{k=1}^{r-1}\sum_{i=0}^{n+2u} \bar{\alpha}_i^k \phi(x_i^k). \tag{23}
$$

Also, θ can be obtained by the following Karush–Kuhn–Tucker (KKT) conditions (Haeser & Ramos, 2020; Su & Luu, 2020; Van Su & Hien, 2021; Zemkoho & Zhou, 2021):

If $i = 0$ and $\alpha_i^k \neq 0$, we have

$$
\frac{1}{Z^k}\sum_{i=n+1}^{n+u} \gamma_i^k g\left(x_i^k\right) = \frac{1}{E^k}\sum_{i=1}^{n} \gamma_i^k y_i^k, \tag{24}
$$

If $i \in \{1, \dots, n\}$ and $0 < \alpha_i^k < C$, we have

$$
y_i^k g\left(x_i^k\right) = 1, \tag{25}
$$

If $i \in \{n+1, \dots, n+u\}$ and $0 < \alpha_i^k < C^*$, we have

$$
y_i^k g\left(x_i^k\right) = 1. \tag{26}
$$

Finally, we summarize our BC-S^{2}OR algorithm in Algorithm 1 where C means the penalty coefficient of labeled samples, C^* means the penalty coefficient of unlabeled samples and k is the Gaussian kernel parameter. Moreover, the illustration of our BC-S^{2}OR algorithm is provided in Fig. 3.

3.3 Time complexity

In this subsection, we analyze the time complexity of our BC-S^{2}OR algorithm. In each iteration, our algorithm mainly consists of three parts: solving the CIL problem, updating

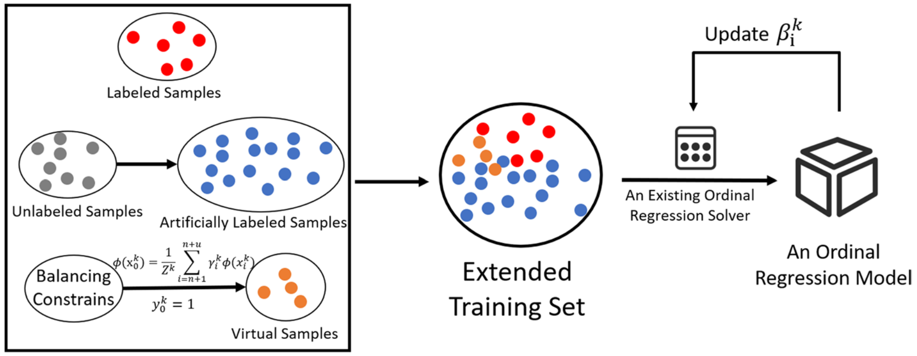

Labeled Samples
Unlabeled Samples
Artificially Labeled Samples
Balancing Constrains
$\phi(x_0^k) = \frac{1}{Z^k}\sum_{i=n+1}^{n+u} \gamma_i^k \phi(x_i^k)$
$y_0^k = 1$
Virtual Samples
Extended Training Set
Update β_i^k
An Existing Ordinal Regression Solver
An Ordinal Regression Model

Fig. 3 Illustration of BC-S^{2}OR

the $\bar{w}$ and updating the β. To solve the CIL problem, we need to solve a SVOR problem with $N = (r-1)(n+2u)$ samples. The time complexity of solving a SVOR problem using a properly modified state-of-the-art solver (Chu & Keerthi, 2007) is $O(N^a)$, where $1 < a < 2.3$. Moreover, the updates of $\bar{w}$ and β have linear time complexity. Therefore, each iteration of our BC-S^{2}OR algorithm requires $O(N^a)$ computations.

4 Experiments

In this section, we present experimental results to demonstrate the superiority of our BC-S^{2}OR algorithm.

4.1 Compared algorithms

We compared our algorithm with existing state-of-the-art algorithms including supervised and semi-supervised algorithms.

SVOR-EXC/SVOR-IMC[2] Support vector approaches for ordinal regression, which optimize multiple thresholds to define parallel decision boundaries for ordinal scales, including two methods of explicit and implicit constraints on thresholds (Chu & Keerthi, 2007).
TOR A transductive ordinal regression method working by a label swapping scheme that facilitates a strictly monotonic decrease in the objective function value (Seah et al., 2012).
ManifoldOR A semi-supervised ordinal regression method projecting the original data to the one-dimensional ranking axis under the manifold learning framework (Liu et al., 2011).
TSVM-CCCP[3] A transductive classification method using convex-concave procedure framework to iteratively optimize non-convex cost functions (Collobert et al., 2006).

[2] SVOR is available at http://www.gatsby.ucl.ac.uk/~chuwei/svor.htm.
[3] TSVM-CCCP is available at https://github.com/fabiansinz/UniverSVM.

Table 2 Datasets

Type	Dataset	Sample size	Features	Class
Benchmark Datasets	Abalone	4177	8	3
	Bank	8192	32	5
	California	20,640	8	5
	Census	22,784	16	5
Real-world Datasets	Software	5269	1000	5
	Luxury	9207	1000	5
	Industrial	15,976	1000	5
	Pantry	25,628	1000	5

TSVM-LDS[4] A transductive classification method trying to place decision boundaries in regions with low density (Chapelle & Zien, 2005).

Specially, we utilized the TSVM-CCCP algorithm and the TSVM-LDS algorithm to handle multi-class data by the one-versus-rest approach.

4.2 Implementations

We implemented our BC-S^{2}OR algorithm by building upon the SVOR-IMC code. Specifically, we introduced an outer loop based on the CCCP approach, where each iteration solves an SVM-like sub-problem. To solve the related dual problem, we modified the sequential minimal optimization (SMO) method, which has been widely used in SVMs (Chen et al., 2006; Nakanishi et al., 2020; Platt, 1998; Sornalakshmi et al., 2020). Additionally, we incorporated the shrinking technique (Chang & Lin, 2011) and a warm-start strategy to accelerate the procedure.

We utilized the open-source codes available for the TSVM-CCCP algorithm, the TSVM-LDS algorithm, the SVOR-EXC algorithm, and the SVOR-IMC algorithm. These codes were provided by their respective authors. Additionally, we implemented the TOR algorithm and the ManifoldOR algorithm by following the algorithm description provided by the authors.

4.3 Datasets

To evaluate the performance of our proposed BC-S^{2}OR algorithm along with other existing state-of-the-art methods, we conducted experiments on a collection of benchmark and real-world datasets, as presented in Table 2. For the benchmark datasets,[5] we discretized the continuous target into ordinal scale using the approach of the equal-frequency bin (Sulaiman & Bakar, 2017). For the real-world datasets,[6] we first processed the text data using the TF-IDF technique (Aizawa, 2003; Onan, 2020; Wang et al., 2020) and then

[4] TSVM-LDS is available at https://github.com/paperimpl/LDS.

[5] Benchmark datasets are available at https://www.dcc.fc.up.pt/~ltorgo/Regression.

[6] Real-world datasets are available at https://nijianmo.github.io/amazon/ index.html.

reduced the dimensionality to 1000 using the PCA approach (Abdi & Williams, 2010; Chen et al., 2020). It is worth noting that we scaled all features to the range $[-1, 1]$ for all datasets. To perform the experiments in a semi-supervised setting, we randomly split the labeled data into different sizes of 100, 200, 300, 400, 500, and 600, and the remaining data formed the set of unlabeled data.

4.4 Experimental setup

All the experiments were conducted on a PC with 48 2.2GHz cores and 80GB RAM, and all the results were the average of 10 trials. The penalty coefficient C of labeled samples was fixed at 10. For the kernel-based methods, we used the Gaussian kernel $k(x, x') = \exp(\kappa||x - x'||^2)$ and fixed κ at 10^{-1}. For all compared algorithms, we followed the hyperparameter settings of their original papers. In particular, the initial penalty coefficient of unlabeled samples was set at 10^{-5} for the TOR algorithm. For the ManifoldOR algorithm, the nearest neighbor number is set at 50 and the value of regularization parameter is 0.5. For the TSVM-CCCP algorithm, we used the symmetric hinge loss and turned the penalty coefficient of unlabeled samples via 5-fold cross validation. For the TSVM-LDS algorithm, the exponent parameter was turned via 5-fold cross validation. And for our BC-S^{2}OR algorithm, we used 5-fold cross validation to optimize the the penalty coefficient C^* of unlabeled samples.

To compare the generalization performance of the algorithms, we employed three widely used measure criteria. The mean zero–one error was used to determine the classification error of the samples, which is defined as

$$\frac{1}{u}\sum_{i=n+1}^{n+u} I[f(x_i) \neq y_i].$$

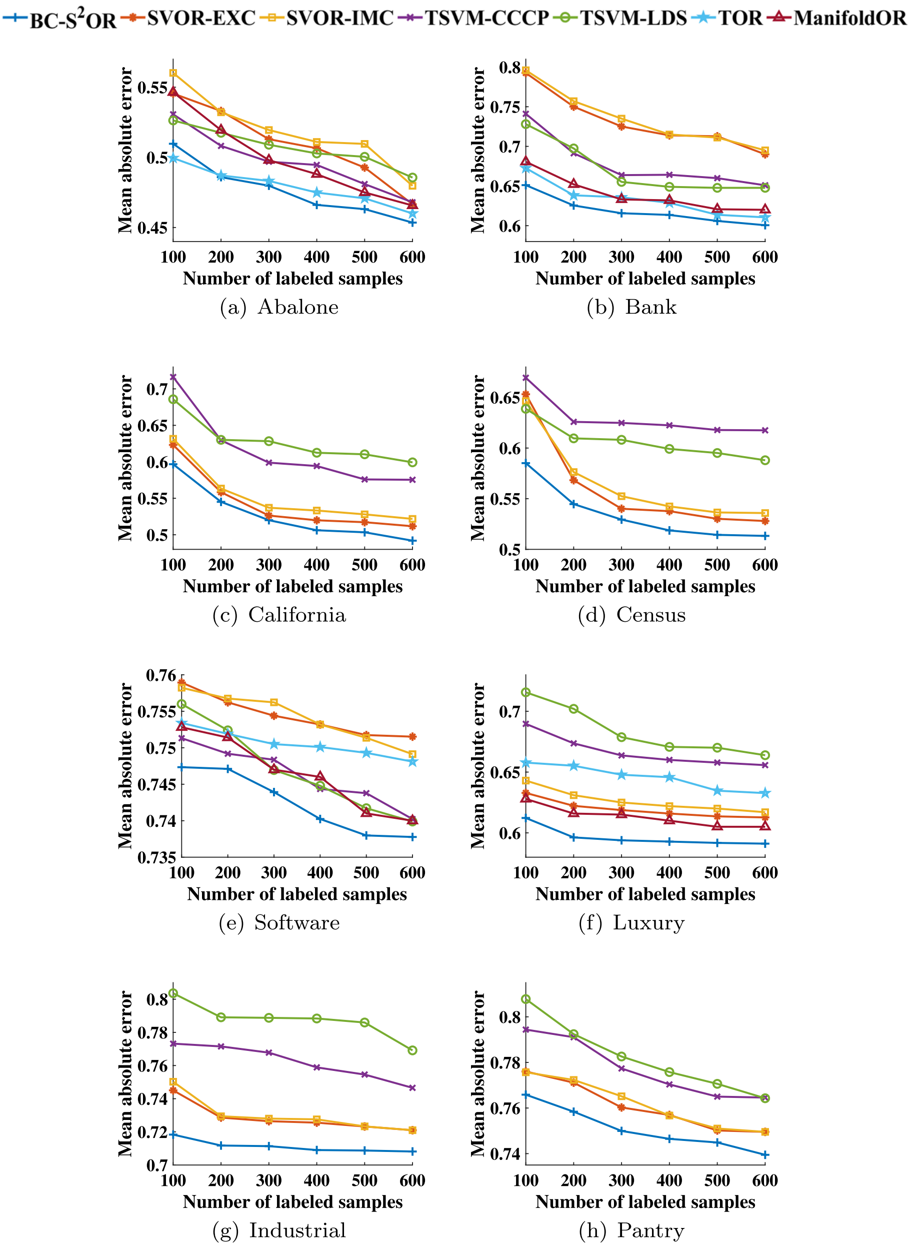
We also used the mean absolute error to measure the deviation between the predicted and true class labels of the samples, which is defined as

$$\frac{1}{u}\sum_{i=n+1}^{n+u} |f(x_i) - y_i|.$$

Furthermore, training time is an important criterion for evaluating an algorithm's efficiency. Consuming less time to meet the requirements denotes more efficient performance. To ensure fairness, we only compare the training time among the S^{2}OR algorithms.

4.5 Results and discussion

Figures 4 and 5 present the results of the mean zero–one error and the mean absolute error on different datasets. We would like to note that due to the excessive training time consumed by the TOR algorithm and the ManifoldOR algorithm, some of the experimental results are missing. Through careful analysis, we find that when the numbers of ordinal classes and samples are relatively small, the performance of the semi-supervised multi-classification algorithms (i.e., the TSVM-LDS algorithm and TSVM-CCCP algorithm) is better than that of the supervised OR algorithms (i.e., the SVOR-EXC algorithm and SVOR-IMC algorithm). This is because the semi-supervised algorithms can utilize unlabeled samples to improve performance. However, as the numbers of ordinal



(a) Abalone

(b) Bank

(c) California

(d) Census

(e) Software

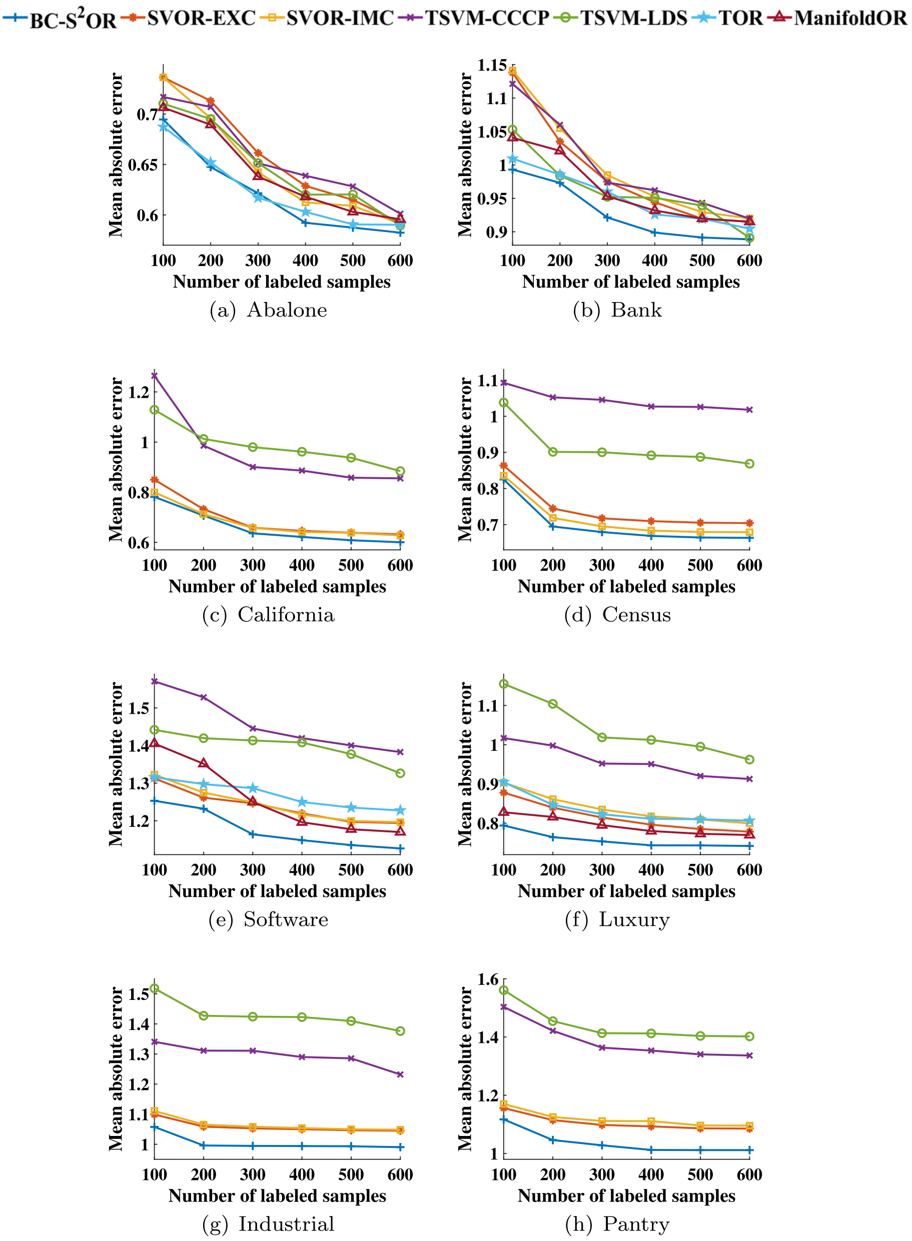
(f) Luxury

(g) Industrial

(h) Pantry

Fig. 4 Mean zero–one errors of different algorithms

classes and samples increase, the OR algorithms start to exhibit better performance than the semi-supervised multi-classification algorithms because they make good use of the ordinal information. Importantly, our proposed BC-S^{2}OR algorithm outperforms the above-mentioned algorithms, which is obviously due to its ability to leverage both the unlabeled samples and the ordinal information. Furthermore, our BC-S^{2}OR algorithm



(a) Abalone

(b) Bank

(c) California

(d) Census

(e) Software

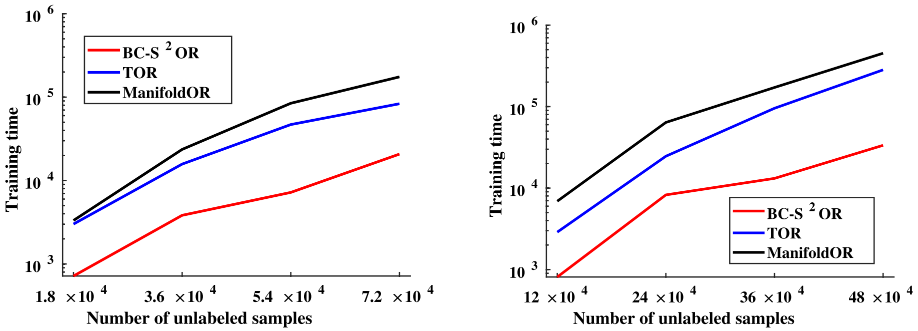
(f) Luxury

(g) Industrial

(h) Pantry

Fig. 5 Mean absolute errors of different algorithms

also outperforms other S^{2}OR methods (i.e., the TOR algorithm and ManifoldOR algorithm). Figure 6 compares the average training time of the S^{2}OR algorithms. The experimental results indicate that our BC-S^{2}OR algorithm performs better than the other S^{2}OR algorithms in terms of training time.



(a) svhn (b) covtype

Fig. 6 Training time (s) of S^{2}OR algorithms on large-scale datasets with fixed 1000 labeled samples

5 Conclusion

In this paper, we present a novel algorithm, named BC-S^{2}OR, that addresses the challenging issue of the balancing constraints in S^{2}OR using virtual samples. Firstly, we introduce a new type of the balancing constraints for S^{2}OR that prevents the majority of unlabeled samples from being classified into only a few classes. Then, to solve our complex optimization problem, we extend the traditional convex-concave procedure (CCCP) approach. We convert the convex inner loop (CIL) problem, which includes multiple equality constraints, into a quadratic problem similar to support vector machine (SVM). In this quadratic problem, the multiple equality constraints are considered as virtual samples. This enables us to use existing solvers (Chu & Keerthi, 2007) to efficiently solve the CIL problems. Numerical experiments carried out on various benchmark and real-world datasets confirm the superiority of our proposed algorithm, which outperforms other supervised and semi-supervised algorithms.

Appendix A: Proof of Lemma 1

Firstly, we define some data subsets where $j \in \{1, \dots, r\}$:

$$
\begin{aligned}
I_j^{low}(\theta) &\overset{\text{def}}{=} \{i \in \{1, \dots, n\} : y_i = j, w^T\phi(x_i) - \theta \geq -1\},\\
I_j^{up}(\theta) &\overset{\text{def}}{=} \{i \in \{1, \dots, n\} : y_i = j, w^T\phi(x_i) - \theta \leq 1\},\\
I^{low}(\theta) &\overset{\text{def}}{=} \{i \in \{n+1, \dots, n+u\} : -1 \leq w^T\phi(x_i) - \theta \leq 0\},\\
I^{up}(\theta) &\overset{\text{def}}{=} \{i \in \{n+1, \dots, n+u\} : 0 < w^T\phi(x_i) - \theta \leq 1\}.
\end{aligned}
$$

It is easy to see that θ_k is optimal if it minimizes the function:

$$
\begin{aligned}
e_k(\theta) = C \sum_{j=1}^{k} \sum_{i \in I_j^{low}(\theta)} (w^T \phi(x_i) - \theta + 1) \\
+ C \sum_{j=k+1}^{r} \sum_{i \in I_j^{up}(\theta)} (-w^T \phi(x_i) + \theta + 1) \\
+ C^* \sum_{i \in I^{low}(\theta)} (w^T \phi(x_i) - \theta + 1) \\
+ C^* \sum_{i \in I^{up}(\theta)} (-w^T \phi(x_i) + \theta + 1).
\end{aligned}
\tag{27}
$$

Then, we obtain the derivative of $e_k(\theta)$ with respect to θ:

$$
\begin{aligned}
\bigtriangledown e_k(\theta) = -C \sum_{j=1}^{k} |I_j^{low}(\theta)| + C \sum_{j=k+1}^{r} |I_j^{up}(\theta)| \\
- C^* |I^{low}(\theta)| + C^* |I^{up}(\theta)| \; := l_k(\theta).
\end{aligned}
\tag{28}
$$

And through simple calculation, we have:

$$
l_{k+1}(\theta) - l_k(\theta) = -C|I_{k+1}^{low}(\theta)| - C|I_{k+1}^{up}(\theta)| < 0. \tag{29}
$$

In order to prove that the order of θ is no longer guaranteed in S^{2}OR problem, we construct a counterexample. Firstly, we find that the derivative $l_k(\theta)$ (28) is not a monotone function. What's more, when θ tends to be positive infinity, we have $l_k(\theta) > 0$ and when θ tends to be negative infinity, we have $l_k(\theta) < 0$. In this case, we assume that $l_k(\theta)$ and $l_{k+1}(\theta)$ both own three zero points, and the first derivative of these points is not zero. Next, we set the zero points of $l_k(\theta)$ from left to right as a_1, a_2 and a_3, and the zero points of $l_{k+1}(\theta)$ as b_1, b_2 and b_3. According to (29), we assume one situation that $a_1 < b_1 < b_2 < a_2 < a_3 < b_3$. Under such condition, $e_k(\theta)$ and $e_{k+1}(\theta)$ both own two local minimum points a_1, a_3 and b_1, b_3. Finally, we assume that point a_3 and point b_1 are global optimal solutions of $e_k(\theta)$ and $e_{k+1}(\theta)$. Then we have $\theta_k = a_3 > \theta_{k+1} = b_1$, and this situation conflicts with the order $\theta_k \leq \theta_{k+1}$ we asked for. Therefore, we conclude that the order of θ is no longer guaranteed in S^{2}OR problem.

Appendix B: Proof of the DC formulation

In order to handle the non-convex objective function (5) better, we rewrite the loss of unlabeled samples:

$$
H_1(|z|) = R_0(z) + R_0(-z) + \text{const}, \tag{30}
$$

where $R_s(t) = \min\{1 - s, \max\{0, 1 - t\}\} = H_1(t) - H_s(t)$ represents a ramp loss. Further more, we rewrite the loss term of unlabeled samples in the objective function (5) with the help of the ramp loss:

$$\begin{aligned}\sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} & H_1\left(\left|g\left(x_i^k\right)\right|\right)\\ &= \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} R_0\left(g\left(x_i^k\right)\right) + \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} R_0\left(-g\left(x_i^k\right)\right) + \text{const}\\ &= \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} H_1\left(g\left(x_i^k\right)\right) + \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} H_1\left(-g\left(x_i^k\right)\right)\\ &\quad - \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} H_0\left(g\left(x_i^k\right)\right) - \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} H_0\left(-g\left(x_i^k\right)\right) + \text{const.}\end{aligned} \tag{31}$$

Then, according to the artificial labeled samples (14), the expression in (31) is in the new formulation:

$$\begin{aligned}\sum_{k=1}^{r-1}\sum_{i=n+1}^{n+u} H_1\left(\left|g\left(x_i^k\right)\right|\right) = & \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+2u} H_1\left(y_i^k g\left(x_i^k\right)\right)\\ & - \sum_{k=1}^{r-1}\sum_{i=n+1}^{n+2u} H_0\left(y_i^k g\left(x_i^k\right)\right) + \text{const},\end{aligned} \tag{32}$$

where the constant does not affect the optimization problem obviously, and we can just ignore it.

Finally, original objective function is in the formulation of

$$\begin{aligned}J(\bar{w}) = & \underbrace{\frac{1}{2}\|w\|^2 + C\sum\nolimits_{k=1}^{r-1}\sum\nolimits_{i=1}^{n} H_1\left(y_i^k g\left(x_i^k\right)\right) + C^*\sum\nolimits_{k=1}^{r-1}\sum\nolimits_{i=n+1}^{n+2u} H_1\left(y_i^k g\left(x_i^k\right)\right)}_{o(\bar{w})}\\ & \underbrace{-\, C^*\sum_{k=1}^{r-1}\sum_{i=n+1}^{n+2u} H_0\left(y_i^k g\left(x_i^k\right)\right)}_{v(\bar{w})}.\end{aligned} \tag{33}$$

Author contributions Bin Gu contributed to the conception and the design of the method. Chenkang Zhang contributed to running the experiments and writing paper. Heng Huang contributed to providing feedback and guidance. All authors contributed on the results analysis and the manuscript writing and revision.

Funding Bin Gu was partially supported by the National Natural Science Foundation of China (No. 61573191).

Data availability Benchmark datasets are available at https://www.dcc.fc.up.pt/~ltorgo/Regression and real-world datasets are available at https://nijianmo.github.io/amazon/ index.html.

Code availability The code will be publicly available once the work is published upon agreement of different sides.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethics approval Not applicable.

References

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics, 2*(4), 433–459.

Aizawa, A. (2003). An information-theoretic perspective of tf-idf measures. *Information Processing & Management, 39*(1), 45–65.

Allahzadeh, S., & Daneshifar, E. (2021). Simultaneous wireless information and power transfer optimization via alternating convex-concave procedure with imperfect channel state information. *Signal Processing, 182*, 107953.

Berg, A., Oskarsson, M., & O'Connor, M. (2021) Deep ordinal regression with label diversity. In: *2020 25th international conference on pattern recognition (ICPR)* (pp. 2740–2747). IEEE.

Bertsekas, D. P. (2014). *Constrained optimization and Lagrange multiplier methods*. Academic Press.

Buri, M., & Hothorn, T. (2020). Model-based random forests for ordinal regression. *The International Journal of Biostatistics*. https://doi.org/10.1515/ijb-2019-0063

Cardoso, J. S., da Costa, J. F. P., & Cardoso, M. J. (2005). Modelling ordinal relations with SVMs: An application to objective aesthetic evaluation of breast cancer conservative treatment. *Neural Networks, 18*(5–6), 808–817.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 1–27.

Chapelle, O., Scholkopf, B., & Zien, A. (2009). Semi-supervised learning. *IEEE Transactions on Neural Networks, 20*(3), 542–542.

Chapelle, O., Sindhwani, V., & Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research, 9*, 203–233.

Chapelle, O., & Zien, A. (2005). Semi-supervised classification by low density separation. In *AISTATS* (Vol. 2005, pp. 57–64). Citeseer.

Chen, P. H., Fan, R. E., & Lin, C. J. (2006). A study on SMO-type decomposition methods for support vector machines. *IEEE Transactions on Neural Networks, 17*(4), 893–908.

Chen, Y., Tao, J., Zhang, Q., Yang, K., Chen, X., Xiong, J., Xia, R., & Xie, J. (2020). Saliency detection via the improved hierarchical principal component analysis method. *Wireless Communications and Mobile Computing*. https://doi.org/10.1155/2020/8822777

Chu, W., & Keerthi, S. S. (2005). New approaches to support vector ordinal regression. In *Proceedings of the 22nd international conference on machine learning* (pp. 145–152).

Chu, W., & Keerthi, S. S. (2007). Support vector ordinal regression. *Neural Computation, 19*(3), 792–815.

Collobert, R., Sinz, F., Weston, J., & Bottou, L. (2006). Large scale transductive SVMs. *Journal of Machine Learning Research, 7*, 1687–1712.

Crammer, K., & Singer, Y. (2002). Pranking with ranking. In *Advances in neural information processing systems* (pp. 641–647).

Fullerton, A. S., & Xu, J. (2012). The proportional odds with partial proportionality constraints model for ordinal response variables. *Social Science Research, 41*(1), 182–198.

Ganjdanesh, A., Ghasedi, K., Zhan, L., Cai, W., & Huang, H. (2020). Predicting potential propensity of adolescents to drugs via new semi-supervised deep ordinal regression model. In *International conference on medical image computing and computer-assisted intervention* (pp. 635–645). Springer.

Garg, B., & Manwani, N. (2020). Robust deep ordinal regression under label noise. In: *Asian conference on machine learning* (pp. 782–796). PMLR.

Gu, B., Zhang, C., Huo, Z., & Huang, H. (2023). A new large-scale learning algorithm for generalized additive models. *Machine Learning, 112*, 3077–3104.

Gu, B., Zhang, C., Xiong, H., & Huang, H. (2022). Balanced self-paced learning for AUC maximization. In *Proceedings of the AAAI conference on artificial intelligence* (vol. 36, pp. 6765–6773).

Haeser, G., & Ramos, A. (2020). Constraint qualifications for Karush–Kuhn–Tucker conditions in multiobjective optimization. *Journal of Optimization Theory and Applications, 187*(2), 469–487.

Herbrich, R. (1999). Support vector learning for ordinal regression. In: *Proceedings of the 9th international conference on neural networks* (pp. 97–102).

Joachims, T. (1999) Transductive inference for text classification using support vector machines. In *ICML* (vol. 99, pp. 200–209).

Li, L., & Lin, H. T. (2007). Ordinal regression by extended binary classification. In *Advances in neural information processing systems* (pp. 865–872).

Li, X., Wang, M., & Fang, Y. (2020). Height estimation from single aerial images using a deep ordinal regression network. *IEEE Geoscience and Remote Sensing Letters, 19*, 1–5.

Liu, Y., Liu, Y., Zhong, S., & Chan, K. C. (2011). Semi-supervised manifold ordinal regression for image ranking. In *Proceedings of the 19th ACM international conference on multimedia* (pp. 1393–1396).

Nakanishi, K. M., Fujii, K., & Todo, S. (2020). Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research, 2*(4), 043158.

Oliveira, A. L., & Valle, M. E. (2020). Linear dilation-erosion perceptron trained using a convex-concave procedure. In *SoCPaR* (pp. 245–255).

Onan, A. (2020). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurrency and Computation: Practice and Experience, 33*, e5909.

Pang, G., Yan, C., Shen, C., Hengel, A. v. d., & Bai, X. (2020). Self-trained deep ordinal regression for end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12173–12182).

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Microsoft Research Technical Report 98.

Rastgar, F., Singh, A. K., Masnavi, H., Kruusamae, K., & Aabloo, A. (2020). A novel trajectory optimization for affine systems: Beyond convex–concave procedure. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1308–1315). IEEE.

Seah, C. W., Tsang, I. W., & Ong, Y. S. (2012). Transductive ordinal regression. *IEEE Transactions on Neural Networks and Learning Systems, 23*(7), 1074–1086.

Shashua, A., & Levin, A. (2003). Ranking with large margin principle: Two approaches. In *Advances in neural information processing systems* (pp. 961–968).

Sornalakshmi, M., Balamurali, S., Venkatesulu, M., Krishnan, M. N., Ramasamy, L. K., Kadry, S., Manogaran, G., Hsu, C. H., & Muthu, B. A. (2020). Hybrid method for mining rules based on enhanced apriori algorithm with sequential minimal optimization in healthcare industry. *Neural Computing and Applications, 34*, 10597–10610.

Srijith, P., Shevade, S., & Sundararajan, S. (2013). Semi-supervised Gaussian process ordinal regression. In *Joint European conference on machine learning and knowledge discovery in databases* (pp. 144–159). Springer.

Su, T. V., & Luu, D. V. (2020). Higher-order Karush–Kuhn–Tucker optimality conditions for Borwein properly efficient solutions of multiobjective semi-infinite programming. *Optimization, 71*, 1749–1775.

Sulaiman, N. S., & Bakar, R. A. (2017). Rough set discretization: Equal frequency binning, entropy/mdl and semi Naives algorithms of intrusion detection system. *Journal of Intelligent Computing, 8*(3), 91.

Tsuchiya, T., Charoenphakdee, N., Sato, I., & Sugiyama, M. (2019). Semi-supervised ordinal regression based on empirical risk minimization. arXiv preprint arXiv:1901.11351.

Van Su, T., & Hien, N. D. (2021). Strong Karush–Kuhn–Tucker optimality conditions for weak efficiency in constrained multiobjective programming problems in terms of mordukhovich subdifferentials. *Optimization Letters, 15*(4), 1175–1194.

Wang, T., Lu, K., Chow, K. P., & Zhu, Q. (2020). COVID-19 sensing: Negative sentiment analysis on social media in China via BERT model. *IEEE Access, 8*, 138162–138169.

Xu, L., Neufeld, J., Larson, B., & Schuurmans, D. (2005). Maximum margin clustering. In *Advances in neural information processing systems* (pp. 1537–1544).

Zemkoho, A. B., & Zhou, S. (2021). Theoretical and numerical comparison of the Karush–Kuhn–Tucker and value function reformulations in bilevel optimization. *Computational Optimization and Applications, 78*(2), 625–674.

Zhai, Z., Gu, B., Deng, C., & Huang, H. (2023). Global model selection via solution paths for robust support vector machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2023.3346765

Zhai, Z., Gu, B., Li, X., & Huang, H. (2020). Safe sample screening for robust support vector machine. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, pp. 6981–6988).

Zhu, X. J. (2005). *Semi-supervised learning literature survey*. Technical report, Department of Computer Sciences, University of Wisconsin-Madison.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.