Research Article

Perceptron Ranking Using Interval Labels with Ramp Loss for Online Ordinal Regression

Cuiqing Zhang,¹ Maojun Zhang ,^{1,2} Xijun Liang ,³ Zhonghang Xia,⁴ and Jiangxia Nan^{1,2}

¹School of Mathematics and Computer Science, Guilin University of Electronic Technology, Guilin 541004, China

²School of Business, Suzhou University of Science and Technology, Suzhou 215009, China

³College of Science, China University of Petroleum, Qingdao 266580, China

⁴School of Engineering and Applied Science, Western Kentucky University, Bowling Green 42101, KY, USA

Correspondence should be addressed to Maojun Zhang; 2721@mail.usts.edu.cn and Xijun Liang; liangxijunsd@163.com

Received 12 September 2020; Revised 9 October 2020; Accepted 5 November 2020; Published 2 December 2020

Academic Editor: Fazal Mahomed

Copyright © 2020 Cuiqing Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to its wide applications and learning efficiency, online ordinal regression using perceptron algorithms with interval labels (PRIL) has been increasingly applied to solve ordinal ranking problems. However, it is still a challenge for the PRIL method to handle noise labels, in which case the ranking results may change dramatically. To tackle this problem, in this paper, we propose noise-resilient online learning algorithms using ramp loss function, called PRIL-RAMP, and its nonlinear variant K-PRIL-RAMP, to improve the performance of PRIL method for noisy data streams. The proposed algorithms iteratively optimize the decision function under the framework of online gradient descent (OGD), and we justify the algorithms by showing the order preservation of thresholds. It is validated in the experiments that both approaches are more robust and efficient to noise labels than state-of-the-art online ordinal regression algorithms on real-world datasets.

1. Introduction

Ordinal regression, also called ranking learning, plays a central role in the learning task where the labels of data samples need to be ordered. It has been routinely used in social ranking tasks, e.g., collaborative filtering [1], ecology [2], and detecting the severity of Alzheimer disease [3], to name a few. In contrast to other types of regression analysis, ordinal regression describes the relationship between variables where the order matters. For example, the label of age category can be sorted as (0–9, 10–19, ..., 90–99) and the levels of bond credits can be sorted as “B” < “A” < “AA” < “AAA” [4]. In general, ordinal regression requires a linear (nonlinear) function and a set of $K - 1$ thresholds, where each threshold corresponds to a class. However, ordinal regression also distinguishes multiclassification in the sense that the output labels have a natural order.

A seminal work of ordinal regression can be traced back to the proportional odds model (POM) proposed by McCullagh

[5], in which a general class of regression models for ordinal data was studied. Later, large margin formulations for ordinal regression were proposed in [6, 7]. Both approaches outperformed existing benchmarks when they were applied to ranking and multiclass classification. Furthermore, to accelerate the training, Gu et al. [8] proposed asynchronous parallel coordinate descent algorithms. Besides, Berg et al. [9] showed that training a classifier can improve neural network accuracy and proposed that using several discrete data representations simultaneously can improve neural network learning compared to a single representation. In many situations, interval labels have to be used instead of the exact label. As an example of predicting product ratings, we get an entire range of scores (e.g., 1–3 and 4–7) from different customers. In [10], a large margin batch algorithm was proposed by using interval labels. The approaches discussed so far are batch algorithms. Recently, in order to reduce computational time, Crammer et al. [11] proposed an online learning algorithm, passive aggressive (PA) algorithm, for ordinal regression, which chooses an appropriate step size to

find the new parameters in every trial. Later, Manwani and Chandra [12] employed the PA algorithm to solve problems with interval labels. Perceptron is another principled method of learning classifiers in online fashion. The authors in [13, 14] proposed an online algorithm (with similar principles to the classic perceptron used for 2-class separation) for finding the set of parallel hyperplanes which would comply with the separation rule. In [15], perceptron-based approach was proposed for ordinal regression using interval labels.

Although existing works have made huge progress in reducing memory requirement and computational time, training models on the dataset with poor quality, e.g., having a lot of noise, remains challenging. Existing scene classification algorithms predominantly focus on static data and are designed to learn discriminant information from clean data. They, however, suffer from two major shortcomings, i.e., the noisy label may negatively affect the learning procedure, and learning from scratch may lead to a huge computational burden. The authors in [16] propose a noise-resilient online classification algorithm, which is scalable and robust to noisy labels and applied it to peptide identification [17]. To reduce the negative influence of outliers, the authors in [18] propose a more robust algorithm termed as ramp loss for twin K -class support vector classification (Ramp-TKSVC) where the ramp loss function was used to substitute the Hinge loss function.

In this work, we aim to study a noise-resilient online learning algorithm to promote the online ordinary regression problem for noisy data streams. Particularly, we propose perceptron ranking algorithms using interval labels with ramp loss (PRIL-RAMP) and its kernel variant (K-PRIL-RAMP) for ordinal regression in an online learning manner. Our key contributions are as follows:

- (1) Propose linear and nonlinear noise-resilient online algorithms, i.e., PRIL-RAMP and K-PRIL-RAMP, and design a procedure to update the model parameters. The optimal parameters are obtained by online gradient descent (OGD) procedure.
- (2) It is theoretically proved that the proposed the PRIL-RAMP algorithm keeps the thresholds sequentially.
- (3) Practical experimental studies on various datasets demonstrate the effectiveness of the proposed algorithms by comparing them with state-of-the-art algorithms.

The paper is organized as follows. In Section 2, we discuss a generic framework of ordinal regression using interval labels. Furthermore, we introduce the proposed PRIL-RAMP and K-PRIL-RAMP algorithms and discuss the order preservation of thresholds of the proposed algorithms. In Section 3, we present experiments and illustrate the comparison results between PRIL-RAMP and the state-of-the-art algorithms. We conclude the whole paper in Section 4.

2. Method

2.1. Learning to Rank in Online Ordinary Regression. Ordinary regression has been successfully applied to a great number of real-world problems. Typical ordinary regression

trains models in a batch manner, i.e., feeding all data into training model at once [6–8]. The batch training manner demands huge computation and memory for large-scale problems, and it is also not adaptable to streaming data. Comparing with the traditional batch learning framework, online learning (shown in Figure 1) learns samples in a streaming fashion so that it can be scaled and respond in real-time. We first introduce the online ordinary regression of the perceptron algorithm and its variant, and then present noise-resilient online learning algorithms PRIL-RAMP and K-PRIL-RAMP in the next section.

Let $\mathcal{X} \subset \mathbb{R}^d$ be the instance space, $\mathcal{Y} = \{1, \dots, K\}$ the label space, and $(x^1, y^1), \dots, (x^T, y^T)$ a sequence of instance-rank pairs, where $x^t \in \mathbb{R}^d$ and $y^t \in \mathcal{Y}$ are its corresponding rank, $t = 1, \dots, T$. We define $\mathcal{Y} = \{1, 2, \dots, K\}$ with “ $<$ ” as the order relation. Every instance $x \in \mathcal{X}$ is assigned an interval label $[y_l, y_r] \in \mathcal{Y} \times \mathcal{Y}$. Particularly, it becomes the exact label scenario when $y_l = y_r$. Let $S = \{(x^1, y_l^1, y_r^1), \dots, (x^T, y_l^T, y_r^T)\}$ be the training set. In ordinary regression, our objective is to learn model $b: \mathcal{X} \rightarrow \mathcal{Y}$, which also defined as the upper interval endpoint:

$$b(x) = \min_{i \in [K]} \{i: f(x) - \theta_i < 0\}, \quad (1)$$

where function $f: \mathcal{X} \rightarrow \mathbb{R}$ is defined as $f(x) = w \cdot x$, and thresholds are $\theta_1 \leq \dots \leq \theta_{K-1}$, $\theta_K = \infty$, $[K] = \{1, \dots, K\}$.

The lower interval endpoint can be defined similarly. Let $L_I^{\text{IMC}}(f(x), \theta, y_l, y_r)$ denote the implicit constraints for ordering of thresholds in θ_i , and I stands for the interval [8]. Then, we have the loss function defined as follows [13]:

$$\begin{aligned} L_I^{\text{IMC}}(f(x), \theta, y_l, y_r) &= \sum_{i=1}^{y_l-1} l_i + \sum_{i=y_r}^{K-1} l_i \\ &= \sum_{i=1}^{y_l-1} [-f(x) + \theta_i]_+ + \sum_{i=y_r}^{K-1} [f(x) - \theta_i]_+. \end{aligned} \quad (2)$$

As the loss function $[-f(x) + \theta_i]_+ = \max(0, -f(x) + \theta_i)$ is convex for each $i \in \{1, \dots, y_l - 1\}$, the sum of them is convex. Similarly, the sum of $[f(x) - \theta_i]_+$, $i \in \{y_r, \dots, K - 1\}$ is also convex. Note that the loss tends to ∞ if there is a large amount of noise. Loss function L_I^{IMC} is primarily designed to learn information from clean data as the overall performance of the learning model degrades with interference of label noise.

2.2. Perceptron Ranking Using Interval Label Algorithms with Ramp Loss. We start the presentation of noise-resilient online learning algorithm with the PRIL algorithm and its variants.

2.2.1. PRIL Algorithm. Perceptron ranking using interval labels (PRILs) was first introduced in [13] for ordinal regression in online fashion. Let $\mathbf{x}^t$ be the observed example, $\mathbf{w}^t \in \mathbb{R}^d$, and $\theta^t \in \mathbb{R}^{K-1}$, which are the parameters of the

FIGURE 1: The framework of online ordinary regression. At a round t , the learner is given a question, $x^t \in \mathcal{X}$, and is required to provide an answer to this question, $f(x^t)$. After predicting an answer, it receives the correct rank y^t and updates its ranking rule by modifying w , so that it enjoys good properties of scalability and real-time.

online ordinary at time t . Let $\bar{I} = \{1, \dots, y_l - 1\} \cup \{y_r, \dots, K - 1\}$. Then, we define

$$z_i = \mathbb{I}_{i \in \{1, \dots, y_l - 1\}} - \mathbb{I}_{i \in \{y_r, \dots, K - 1\}}, \quad \forall i \in \bar{I}, \quad (3)$$

where $\mathbb{I}_A$ is an indicator function which takes value 1 if A is true. Thus $z_i = 1, i \in 1, \dots, y_l$ and $z_i = -1, i \in y_r, \dots, K - 1$. Let $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$. For given sample-interval pair $\{\mathbf{x}, (y_l, y_r)\}$, the loss $L_I^{\text{IMC}}(\mathbf{w} \cdot \mathbf{x}, \theta, y_l, y_r) = 0$ only when

$$\begin{aligned} f(\mathbf{x}) - \theta_i &\geq 0, \quad \forall i \in \{1, \dots, y_l - 1\}, \\ f(\mathbf{x}) - \theta_i &\leq 0, \quad \forall i \in \{y_r, \dots, K\}. \end{aligned} \quad (4)$$

Therefore, $L_I^{\text{IMC}}(\mathbf{w} \cdot \mathbf{x}, y_l, y_r, \theta) = 0$ requires that $z_i(\mathbf{w} \cdot \mathbf{x} - \theta_i) \geq 0, \forall i \in \bar{I}$. The perceptron ranking loss function in linear situation can be rewritten as

$$L_I^{\text{IMC}}(\mathbf{w} \cdot \mathbf{x}, y_l, y_r, \theta) = \sum_{i \in \bar{I}} \max(0, -z_i(\mathbf{w} \cdot \mathbf{x} - \theta_i)). \quad (5)$$

2.2.2. PRIL-RAMP and K-PRIL-RAMP Algorithms. Now, we present the PRIL-RAMP procedure to minimize the estimated risk using the OGD structure [14] of ramp loss in online ordinary regression.

(1) *Ramp Loss.* To offset the influence of noise data, we apply ramp loss as a surrogate of the perceptron loss function. The surrogate of perceptron loss function is ramp loss is defined as

$$\begin{aligned} \ell_s(f(\mathbf{x}), y_l, y_r, \theta) &= \text{ramp}_s(z_i(f(\mathbf{x}) - \theta_i)) = \min(s, \max(0, -z_i(f(\mathbf{x}) - \theta_i))), \quad i \in \bar{I} \\ &= \begin{cases} \min(s, \max(0, -\langle \mathbf{w}, \mathbf{x} \rangle + \theta_i)), & i \in \{1, \dots, y_l^t - 1\}, \\ \min(s, \max(0, \langle \mathbf{w}, \mathbf{x} \rangle - \theta_i)), & i \in \{y_r^t, \dots, K - 1\}. \end{cases} \end{aligned} \quad (6)$$

Notice that the ramp loss is a noise-resilient function, which makes the contribution of noise loss zero when $z_i(f(\mathbf{x}) - \theta_i)$ tends to $\pm\infty$. In classification methodologies, robustness to noise is always an important issue. The effect of noise samples can be significantly large since the penalty given to the outliers by the perceptron loss is quite huge, as any convex loss is unbounded. On the other hand, the ramp loss has an upper bound, and hence, it can control the effect of noisy samples and remove the effect of noise. Plots of perceptron loss and ramp loss in Figure 2 show the robustness (noise-resilient) of the ramp loss.

In addition, the ramp loss is a difference of convex (DC) function and can be represented as the difference of two convex functions:

$$\text{ramp}_s(z_i(f(\mathbf{x}) - \theta_i)) = l_0(z_i(f(\mathbf{x}) - \theta_i)) - l_s(z_i(f(\mathbf{x}) - \theta_i)), \quad (7)$$

where l_0 is perceptron loss function, as illustrated in Figure 3.

(2) *PRIL-RAMP Algorithm.* In the setting of ordinal regression, instead of considering all the categories to

contribute errors to threshold, we allow the ramp loss function to update the thresholds. Ordinal inequalities on thresholds are satisfied automatically at the optimal solution.

Figure 4 uses an example to illustrate the update rule of PRIL-RAMP algorithm. In this example, we set $\theta = \{1, \dots, 10\}$. Note that $\theta_{10} = \infty$ is omitted from all the plots in Figure 4. The correct rank of the instance is $[y_l, y_r] = [9, 10]$, and thus, the value of $\mathbf{w} \cdot \mathbf{x}^t$ should fall in the last interval, between θ_9 and θ_{10} . However, as you can see in Figure 4, the value of $\mathbf{w} \cdot \mathbf{x}^t$ falls below interval $[6, 7]$, and the predicted rank is $\widehat{y}_y^t = 6$. Threshold values of θ_7, θ_8 , and θ_9 are errors since the values of θ_7, θ_8 , and θ_9 are higher than $\mathbf{w} \cdot \mathbf{x}^t$. To mend the mistake, the algorithm decreases the values of θ_7, θ_8 by a unit and replace them with θ_{7-1} and θ_{8-1} . It also modifies $\mathbf{w}$ to be $\mathbf{w} + 2\mathbf{x}^t$ since $\sum_{i: y_i(\mathbf{w} \cdot \mathbf{x}^t - \theta_i) \leq 0} y_i = 2$. Thus, the inner-product $\mathbf{w} \cdot \mathbf{x}^t$ increases by $2\|\mathbf{x}^t\|^2$. This update is illustrated at the middle plot of Figure 4. The updated prediction rule is sketched on the right-hand side. Note that after the update, predicted rank of $\mathbf{x}^t$ is $y = 8$ which is closer to the true rank $y = 9$. We allow a small error of one unit. Note that the data in the interval label $[1, 2]$ (in red) are noise. By replacing perceptron loss with ramp loss,

FIGURE 2: Perceptron loss is shown in I, and II is the ramp loss.

FIGURE 3: Ramp loss is a DC function.

FIGURE 4: An illustration of PRIL-RAMP algorithm framework for online ordinary regression.

PRIL-RAMP guarantees that the parameters of this data with scoring $z_i(f(\mathbf{x}) - \theta_i) < -8$ are not updated, which significantly controls the effect of noisy data.

Next, we describe the procedure in PRIL-RAMP to update the parameters for predicting the label:

$$\begin{aligned}
 & (w^{t+1}, \theta^{t+1}) \\
 &= \arg \min_{\mathbf{w}, \boldsymbol{\theta}} \sum_{i=1}^{y_l^t-1} \text{ramp}(f(\mathbf{x}), y_l, y_r, \boldsymbol{\theta}) + \sum_{i=y_r^t}^{K-1} \text{ramp}(f(\mathbf{x}), y_l, y_r, \boldsymbol{\theta}) \\
 &= \arg \min_{\mathbf{w}, \boldsymbol{\theta}} \underbrace{\sum_{i=1}^{y_l^t-1} l_0(z_i(\langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i)) + \sum_{i=y_r^t}^{K-1} l_0(z_i(\langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i))}_{\text{convex}} \\
 &\quad - \underbrace{\left[\sum_{i=1}^{y_l^t-1} l_s(z_i(\langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i)) + \sum_{i=y_r^t}^{K-1} l_s(z_i(\langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i)) \right]}_{\text{concave}} \\
 &\triangleq \arg \min_{\mathbf{w}, \boldsymbol{\theta}} F_1(\mathbf{w}, \boldsymbol{\theta}) - F_2(\mathbf{w}, \boldsymbol{\theta}),
 \end{aligned} \tag{8}$$

where

$$\begin{aligned} F_1(\mathbf{w}, \boldsymbol{\theta}) &= \sum_{i=1}^{y_l^t-1} l_0(-\langle \mathbf{w}, \mathbf{x}^t \rangle + \theta_i) + \sum_{i=y_r^t}^{K-1} l_0(\langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i) = \sum_{i=1}^{y_l^t-1} \max(0, -\langle \mathbf{w}, \mathbf{x}^t \rangle + \theta_i) + \sum_{i=y_r^t}^{K-1} \max(0, \langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i), \\ F_2(\mathbf{w}, \boldsymbol{\theta}) &= \sum_{i=1}^{y_l^t-1} l_s(-\langle \mathbf{w}, \mathbf{x}^t \rangle + \theta_i) + \sum_{i=y_r^t}^{K-1} l_s(\langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i) = \sum_{i=1}^{y_l^t-1} \max(0, s + \langle \mathbf{w}, \mathbf{x}^t \rangle - \theta_i) + \sum_{i=y_r^t}^{K-1} \max(0, s - \langle \mathbf{w}, \mathbf{x}^t \rangle + \theta_i). \end{aligned} \quad (9)$$

Note that $F_1(\mathbf{w}, \boldsymbol{\theta})$ is a convex function while $-F_2(\mathbf{w}, \boldsymbol{\theta})$ is a concave function. Let $\boldsymbol{\beta} = (\mathbf{w}, \boldsymbol{\theta})$, then we have

$$\boldsymbol{\beta}^{t+1} \triangleq \arg \min_{\boldsymbol{\beta}} F_1(\boldsymbol{\beta}) - F_2(\boldsymbol{\beta}). \quad (10)$$

The concave-convex procedure (CCCP) [19] can be used to obtain the optimal solution to the DC function. However, as is mentioned above, this batch learning manner cannot meet the real-time requirement when dealing with streaming data. In this work, we use the OGD method to find a near-optimal solution. It is a tradeoff between the accuracy and scalability. Particularly, we have

$$\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t + \eta \nabla F_1(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\boldsymbol{\beta}^t} + \eta \nabla F_2(\boldsymbol{\beta})|_{\boldsymbol{\beta}=\boldsymbol{\beta}^t}. \quad (11)$$

To estimate $\mathbf{w}$ and $\boldsymbol{\theta}$, we initialize $\mathbf{w}^0 = 0$ and $\boldsymbol{\theta}^0 = 0$. Let $\mathbf{w}^t$ and $\boldsymbol{\theta}^t$ be the estimates of the parameters in the beginning of trial t , $\mathbf{x}^t$ be the example observed, and $[y_l^t, y_r^t]$ be its label interval. $\mathbf{w}^{t+1}$ and $\boldsymbol{\theta}^{t+1}$ are estimated as follows:

$$\begin{aligned} \mathbf{w}^{t+1} &= \mathbf{w}^t - \eta \nabla_{\mathbf{w}} L(\mathbf{w} \cdot \mathbf{x}^t, y_l^t, y_r^t, \boldsymbol{\theta})|_{\mathbf{w}^t, \boldsymbol{\theta}^t} \\ &= \mathbf{w}^t + \eta (\nabla_{\mathbf{w}} F_1(\boldsymbol{\beta})|_{\mathbf{w}=\mathbf{w}^t} - \nabla_{\mathbf{w}} F_2(\boldsymbol{\beta})|_{\mathbf{w}=\mathbf{w}^t}). \end{aligned} \quad (12)$$

Let $s = -1$, then we have

$$\begin{aligned} \mathbf{w}^{t+1} &= \mathbf{w}^t + \eta \sum_{i \in \bar{I}^t} z_i^t \mathbf{x}^t \mathbb{I}_{\{-1 < z_i^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_i^t) < 0\}}, \\ \theta_i^{t+1} &= \theta_i^t - \eta \frac{\partial L(\mathbf{w}^t \cdot \mathbf{x}^t, y_l^t, y_r^t, \boldsymbol{\theta})}{\partial \theta_i} \Big|_{\mathbf{w}^t, \boldsymbol{\theta}^t} = \begin{cases} \theta_i^t - \eta z_i^t \mathbb{I}_{\{-1 < z_i^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_i^t) < 0\}}, & t \in \bar{I}^t, \\ \theta_i^t, & t \notin \bar{I}^t. \end{cases} \end{aligned} \quad (13)$$

Note that sets $\bar{I}$ are known at every trial t . The complete description of the PRIL-RAMP algorithm is as given in Algorithm 1.

(3) *K-PRIL-RAMP Algorithm*. For nonlinear ordinary regression, we choose the suitable kernel function $\kappa: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ as proposed by Manwani [15]. That is,

$$f^{t+1}(\mathbf{x}) = \sum_{s=1}^t \sum_{i \in \bar{I}^s} \tau_i^s \kappa(\mathbf{x}^s, \mathbf{x}), \quad (14)$$

where $s \in [t]$:

$$h(\mathbf{x}) = \min_{i \in [K]} \left\{ i: \sum_{s=1}^T \left(\sum_{i \in \bar{I}^s} \tau_i^s \kappa(\mathbf{x}^s, \mathbf{x}) \right) - \theta_i^{T+1} \right\}. \quad (15)$$

The Kernel PRIL algorithm with ramp loss (K-PRIL-RAMP) for PRIL-RAMP is discussed in Algorithm 2.

2.3. Theoretical Analysis of PRIL-Ramp Algorithm.

Considering the property of PRIL algorithms, we now show that the PRIL-RAMP algorithm inherently maintains the ordering of thresholds at each iteration.

Theorem 1. *Order preservation of thresholds in the PRIL-RAMP algorithm: let $\mathbf{w}^t$ and $\theta_1^t \leq \dots \leq \theta_{K-1}^t$ be the thresholds at trial t . Let $\mathbf{x}^t$ be the instance at trial t and $[y_l^t, y_r^t]$ be its corresponding interval label. Let $\theta_1^{t+1}, \dots, \theta_{K-1}^{t+1}$ be the updated thresholds using PRIL-RAMP. Then, $\theta_1^{t+1} \leq \dots \leq \theta_{K-1}^{t+1}$.*

Proof. We need to analyse four different cases as follows:

- (1) We know that $\theta_k^{t+1} = \theta_k^t, k = y_l^t, \dots, y_r^t - 1$. PRIL-RAMP does not update thresholds in this range. Thus, $\theta_1^{t+1} \leq \dots \leq \theta_{K-1}^{t+1}$.
- (2) $k = y_l^t - 1$: in this case, θ_{k+1}^{t+1} is in the update rule. Thus, using the fact that $\theta_{k+1}^{t+1} - \theta_k^t \geq 0$, we get

$$\theta_{k+1}^{t+1} - \theta_k^{t+1} = \theta_{k+1}^t - \theta_k^t + z_k^t \mathbb{I}_{\{-1 < z_k^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 0\}} = \theta_{k+1}^t - \theta_k^t + \mathbb{I}_{\{-1 < z_k^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 0\}} \geq 0. \quad (16)$$

Input: Training set S, η
Initialize: Set $t = 1, \mathbf{w}^1 = 0$ and $\theta_1^1 = \dots = \theta_{K-1}^1 = 0, \theta_K^1 = \infty$
Output: Predict $\hat{y}_t (t = 1, \dots, T)$
(1) **for** $i \leftarrow t = 1, \dots, T$ **do**
(2) Receive instance $\mathbf{X}_t$ from S
(3) Predict: $\hat{y}_t$
(4) observe: $\mathbf{y}_t^t \mathbf{y}_r^t$
(5) Set $z_i^t = +1, i = 1, \dots, y_l^t$
(6) Set $z_i^t = -1, i = y_r^t, \dots, K - 1$
(7) Initialize $\tau_i^t = 0, i \in [K]$
(8) **for** $i \in \bar{I}^t$ **do**
(9) **if** $-1 < z_i^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_i^t) < 0$ **then**
(10) $\tau_i^t = z_i^t$
(11) **end if**
(12) **end for**
(13) $\mathbf{w}^{t+1} = \mathbf{w}^t + \sum_{i=1}^K \tau_i^t \mathbf{x}^t$
(14) $\theta_i^{t+1} = \theta_i^t - \tau_i^t, i = 1, \dots, K$
(15) **end for**
(16) $h(\mathbf{x}) = \min_{i \in [K]} \{i: \mathbf{w}^{T+1} \cdot \mathbf{x} - \theta_i^{T+1} < 0\}$

ALGORITHM 1: PRIL-RAMP algorithm (perceptron ranking using interval label with ramp loss).

Input: Training set S, η
Initialize: Set $t = 0, \mathbf{w}_1 = 0$ and $\theta_1^0 = \dots = \theta_{K-1}^0 = 0, \theta_K^0 = \infty; \tau_i^0 = 0, i \in [K]; f^0(\cdot) = 0$
Output Predict $\hat{y}_t (t = 1, \dots, T)$
for $i \leftarrow t = 1, \dots, T$ **do**
(2) Receive instance $\mathbf{X}_t$ from S
(3) Predict: $\hat{y}_t$
(4) observe: $\mathbf{y}_t^t \mathbf{y}_r^t$
(5) Set $z_i^t = +1, i = 1, \dots, y_l^t$
(6) Set $z_i^t = -1, i = y_r^t, \dots, K - 1$
(7) Initialize $\tau_i^t = 0, i \in [K]$
(8) **for** $i \in \bar{I}^t$ **do**
(9) **if** $-1 < z_i^t (f^t(\mathbf{x}^t) - \theta_i^t) < 0$ **then**
(10) $\tau_i^t = z_i^t$
(11) **end if**
(12) **end for**
(13) $f^{t+1}(\cdot) = f^t(\cdot) + \sum_{i \in \bar{I}^t} \tau_i^t \kappa(\mathbf{x}^t, \cdot)$
(14) $\theta_i^{t+1} = \theta_i^t - \tau_i^t, i = 1, \dots, K$
(15) **end for**
(16) $h(\mathbf{x}) = \min_{i \in [K]} \{i: f^{T+1}(\mathbf{x}) - \theta_i^{T+1} < 0\}$

ALGORITHM 2: K-PRIL-RAMP algorithm (kernel perceptron ranking using interval label with ramp loss).

(3) $k \in \{1, \dots, y_l^t - 2\}$: we see that

$$\begin{aligned}
\theta_{k+1}^{t+1} - \theta_k^{t+1} &= \theta_{k+1}^t - \theta_k^t - z_{k+1}^t \mathbb{I}_{\{-1 < z_{k+1}^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 0\}} + z_k^t \mathbb{I}_{\{-1 < z_k^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 0\}} \\
&= \theta_{k+1}^t - \theta_k^t - \mathbb{I}_{\{-1 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 0\}} + \mathbb{I}_{\{-1 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 0\}}.
\end{aligned} \tag{17}$$

Thus, there can be two cases only.

(a) $\theta_{k+1}^t = \theta_k^t$: we simply get $\theta_{k+1}^{t+1} = \theta_k^{t+1}$.

- (b) $\theta_{k+1}^t > \theta_k^t$: since $\theta_i^t \in \mathbb{Z}$, $\forall k, \forall t$, we get $\theta_{k+1}^t \geq \theta_k^t + 1$.
That means

$$\theta_{k+1}^{t+1} - \theta_k^{t+1} \geq 0. \quad (19)$$

$$\theta_{k+1}^{t+1} - \theta_k^{t+1} \geq 1 - \mathbb{I}_{\{-1 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 0\}} + \mathbb{I}_{\{-1 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 0\}}. \quad (18)$$

However, $-\mathbb{I}_{\{-1 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 0\}} + \mathbb{I}_{\{-1 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 0\}} \in \{-1, 0\}$.
Thus,

(4) $k \in \{y_r, \dots, K-1\}$: we see that

$$\begin{aligned} \theta_{k+1}^{t+1} - \theta_k^{t+1} &= \theta_{k+1}^t - \theta_k^t - z_{k+1}^t \mathbb{I}_{\{-1 < z_{k+1}^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 0\}} + z_k^t \mathbb{I}_{\{-1 < z_k^t (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 0\}} \\ &= \theta_{k+1}^t - \theta_k^t + \mathbb{I}_{\{0 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 1\}} - \mathbb{I}_{\{0 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 1\}}. \end{aligned} \quad (20)$$

- (a) $\theta_{k+1}^t = \theta_k^t$: we simply get $\theta_{k+1}^{t+1} = \theta_k^{t+1}$.
(b) $\theta_{k+1}^t > \theta_k^t$: since $\theta_i^t \in \mathbb{Z}$, $\forall k, \forall t$, we get $\theta_{k+1}^t \geq \theta_k^t + 1$.
That means

$$\theta_{k+1}^{t+1} - \theta_k^{t+1} \geq 1 - \mathbb{I}_{\{0 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 1\}} + \mathbb{I}_{\{0 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 1\}}. \quad (21)$$

However, $-\mathbb{I}_{\{0 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_{k+1}^t) < 1\}} + \mathbb{I}_{\{0 < (\mathbf{w}^t \cdot \mathbf{x}^t - \theta_k^t) < 1\}} \in \{-1, 0\}$.
Thus,

$$\theta_{k+1}^{t+1} - \theta_k^{t+1} \geq 0. \quad (22)$$

This completes proof.

Similar demonstrations can be given for the K-PRIL-RAMP algorithm, and hence, we skip its proof. Theorem 1 shows that the K-PRIL-RAMP algorithm can keep the thresholds in sequential. $\square$

3. Experiments

We evaluate performance of the PRIL-RAMP algorithm by comparing it with other benchmark methods on datasets with various ratios of noise. All experimental studies are performed in MATLAB R2016a environment on a PC with 2.5 GHz Intel Core i5 processors and 8 GB RODRAM running under the Windows 10 operating system.

3.1. Dataset. All datasets are obtained from UCI (<http://archive.ics.uci.edu/ml/>) machine learning repository [20] and LIBSVM website, and data features are normalized to zero mean and unit variance coordinatewise. More details about the datasets are provided as follows:

Abalone: the dataset contains 4177 instances with 8 attributes, related to the physical measurement of Abalone found in Australia. A typical task based on the dataset is to predict the age of the Abalone using the “Rings” attribute which varies from 1–29. We divide the target variable 1–29 into 4 intervals as 1–7, 8–9, 10–12, and 13–29 [12].

Parkinsons-updrs: the dataset consists of a range of biomedical measurements by voice with early-stage Parkinson’s disease. There are 5847 instances with 21 features. The target variable of this dataset is “total UPDRS” (Clinician’s total UPDRS score) for the instance which varies from 7 to 54.992. There are 5 groups of target variable 7–54.992: $\theta = (17, 17, 27, 37, 54.992)$ [12].

Real estate valuation: the real estate valuation is collected from Sindian District, New Taipei City, Taiwan. This dataset has 414 instances with 7 attributes. We focus on predicting the house price of unit area which ranges from 0 to 200. The created 4 intervals are 0–20, 21–40, 41–60, and 61–200.

Table 1 shows the characteristics of the 5 datasets including the number of patterns, attributes, and classes, and also the number of patterns per class. To show the statistical properties of different datasets, we drew a box plot as shown in Figure 5. The x -axis denotes the dimensions of examples, and the y -axis corresponds to the outliers. It indicates that there exists noise in the datasets itself, especially in Parkinsons-updrs dataset.

3.2. Generating Noise. Noises are generated by choosing $m\%$ instances from the dataset randomly. Firstly, for covariance variables x , the error term follows a mixture normal distribution $0.8N(0, 1) + 0.2N(0, 10^2)$ [21]. The visualization of the noise datasets is proved in Figure 6. The red points denote the original examples, and the blue circles are corresponding to the outliers. Then, the label is generated in the target variables. For each of these examples, we randomly assign one of the following intervals, i.e., $[y-1, y]$, $[y, y+1]$, $[y-2, y-1]$, and $[y+1, y+2]$, where y is the actual label. Finally, we consider $m = 25$ and 50 .

3.3. Kernel Functions. We use the following kernel functions for different datasets [15] for Kernel PRIL.

$$\text{Abalone: } \kappa(x_1, x_2) = (x_1 \cdot x_2 + 1)^3$$

TABLE 1: Ordinal regression datasets.

Dataset	#Pat	#Attr	#Classes	#Class distribution
Abalone	4177	7	4	(839, 1257, 1388, 693)
Parkinsons-updrs	5847	21	5	(10, 798, 1885, 1443, 1749)
Real estate valuation	414	6	4	(36, 188, 171, 19)

FIGURE 5: Box plots of three benchmark datasets. (a) Abalone. (b) Parkinsons-updrs. (c) Real estate valuation.

Parkinsons-updrs: $\kappa(x_1, x_2) = (x_1 \cdot x_2)^2$

Real Estate Valuation: $\kappa(x_1, x_2) = (x_1 \cdot x_2)^4$

3.4. Result Comparison. We compare noise tolerance performance of PRIL-RAMP and K-PRIL-RAMP algorithms with basic PRIL and Kernel PRIL proposed in [15], respectively, where PRIL is perceptron-based approach for ordinal regression using interval labels in online. The difference between PRIL and Kernel PRIL is that the PRIL is linear while Kernel PRIL uses nonlinear kernel function in the experiments.

3.4.1. Accuracy. We use average accuracy as the metrics to evaluate the performance of all algorithms in the experimental studies. Accuracy, the short form of the average accuracy, is defined as $1 - \text{MZE}$ (mean zero-one error) $= 1 - (1/T) \sum_{t=1}^T [\hat{y}^t \neq y^t]$ [22], where y^t is the true label and $\hat{y}^t$ is the predicted label. We run all algorithms 10 times and average the instantaneous accuracy across the 10 runs. The values of accuracy output by four algorithms running on different noisy data are reported in Figures 7–9. In the figures, the x -axis represents the number of examples, and the y -axis represents the accuracy on test examples. We select appropriate ramp loss parameter s for PRIL-RAMP and K-PRIL-RAMP and set $\eta = 1$.

The results of experimental studies are summarized as follows:

- (1) We see that for all the datasets, accuracy will gradually increase and stabilize as T increases. The accuracy in all algorithms decreases as the level of

noise increases. The interference of label noise in the datasets does degrade the accuracy of prediction.

- (2) For different datasets, linear function algorithms (PRIL and PRIL-RAMP) and nonlinear function kernel algorithms (Kernel PRIL and K-PRIL-RAMP) perform not the same. That is determined by each dataset. Nevertheless, kernel algorithms need to solve a more complex model for ranking problems.
- (3) In general, PRIL-RAMP outperforms PRIL. Particularly, K-PRIL-RAMP shows consistently higher test accuracy than Kernel PRIL over all datasets with different noise levels. Also, noise-resilient algorithms (PRIL-RAMP and K-PRIL-RAMP) and noise-sensitive algorithms (PRIL and Kernel PRIL) work equally well in terms of the accuracy in the environment without noise added. Moreover, the gap of accuracy between noise-tolerances (PRIL-RAMP and K-PRIL-RAMP) and noise-sensitive (PRIL and Kernel PRIL) algorithms increases along with the level of noise.
- (4) On Abalone and Parkinson datasets, linear algorithms perform better than kernel methods when noise is added. The prediction accuracies of the PRIL and Kernel PRIL algorithms are relatively low and unstable in datasets with 25% and 50% noise, and this result shows that the two algorithms are very sensitive to noise. Interestingly, the noise-resilient PRIL-RAMP and K-PRIL-RAMP algorithms show better noise-tolerant performance. On the real estate valuation dataset, the accuracies of four algorithms are close when the dataset has 0% noise. However,

FIGURE 6: The noise data of three datasets. (a) Abalone. (b) Parkinsons-updrs. (c) Real estate valuation.

FIGURE 7: The accuracy comparison result of Abalone in different noise data. (a) Abalone. (b) Abalone-25% noise data. (c) Abalone-50% noise data.

FIGURE 8: The accuracy comparison result of Parkinsons-updrs in different noise data. (a) Parkinsons-updrs. (b) Parkinsons-updrs, 25% noise data. (c) Parkinsons-updrs, 50% noise data.

FIGURE 9: The accuracy comparison result of real estate valuation in different noise data. (a) Real estate valuation. (b) Real estate valuation, 25% noise data. (c) Real estate valuation, 50% noise data.

TABLE 2: Sensitivity estimation of PRIL, PA-I, and proposed PRIL-RAMP algorithm of three datasets under 0% noise data.

Dataset	Rounds	Method	MAE	RMSE	Spearman's correlation coefficient	Discard sample	Discard rate	Time (s)
Abalone	$T=1400$	PA-I	2.512 ± 0.014	2.728 ± 0.029	0.734 ± 0.051	0 ± 0	0.00	0.589
		PRIL	0.597 ± 0.143	0.087 ± 0.078	0.709 ± 0.006	0 ± 0	0.00	0.190
		PRIL-RAMP	0.766 ± 0.076	1.055 ± 0.098	0.415 ± 0.061	42 ± 11	3.00	0.018
	$T=2800$	PA-I	2.56 ± 0.035	2.661 ± 0.036	0.734 ± 0.051	0 ± 0	0.00	1.075
		PRIL	0.600 ± 0.357	0.871 ± 0.098	0.671 ± 0.020	0 ± 0	0.00	0.388
		PRIL-RAMP	0.732 ± 0.143	1.057 ± 0.096	0.538 ± 0.071	97 ± 6	3.46	0.041
	$T=4100$	PA-I	2.462 ± 0.088	2.654 ± 0.074	0.708 ± 0.044	0 ± 0	0.00	1.471
		PRIL	0.605 ± 0.084	0.872 ± 0.091	0.671 ± 0.020	0 ± 0	0.00	0.582
		PRIL-RAMP	0.774 ± 0.064	1.059 ± 0.014	0.426 ± 0.094	124 ± 12	3.02	0.062
Parkinsons-uprds	$T=2000$	PA-I	0.792 ± 0.000	1.070 ± 0.047	0.436 ± 0.066	0 ± 0	0.00	0.449
		PRIL	0.741 ± 0.020	1.079 ± 0.081	0.541 ± 0.079	0 ± 0	0.00	0.279
		PRIL-RAMP	0.682 ± 0.010	1.017 ± 0.034	0.601 ± 0.028	178 ± 21	8.90	0.302
	$T=4000$	PA-I	0.756 ± 0.075	1.044 ± 0.039	0.480 ± 0.017	0 ± 0	0.00	0.928
		PRIL	0.648 ± 0.075	0.979 ± 0.158	0.601 ± 0.009	0 ± 0	0.00	0.558
		PRIL-RAMP	0.640 ± 0.075	0.966 ± 0.825	0.639 ± 0.071	402 ± 8	10.05	0.621
	$T=5800$	PA-I	0.750 ± 0.036	1.040 ± 0.038	0.492 ± 0.025	0 ± 0	0.00	1.286
		PRIL	0.625 ± 0.057	0.946 ± 0.055	0.623 ± 0.032	0 ± 0	0.00	0.842
		PRIL-RAMP	0.612 ± 0.080	0.937 ± 0.063	0.664 ± 0.059	745 ± 23	12.8	0.911
Real estate valuation	$T=140$	PA-I	0.771 ± 0.014	1.076 ± 0.017	0.259 ± 0.009	0 ± 0	0.00	1.895
		PRIL	0.638 ± 0.035	0.928 ± 0.025	0.364 ± 0.080	0 ± 0	0.00	0.078
		PRIL-RAMP	0.764 ± 0.086	1.045 ± 0.039	0.278 ± 0.033	10 ± 1	7.14	0.108
	$T=280$	PA-I	0.700 ± 0.018	0.967 ± 0.003	0.431 ± 0.007	0 ± 0	0.00	3.728
		PRIL	0.638 ± 0.035	0.928 ± 0.025	0.364 ± 0.080	0 ± 0	0.00	0.156
		PRIL-RAMP	0.653 ± 0.057	0.939 ± 0.022	0.331 ± 0.065	34 ± 3	12.14	0.235
	$T=400$	PA-I	0.777 ± 0.024	1.033 ± 0.042	0.470 ± 0.056	0 ± 0	0.00	5.954
		PRIL	0.638 ± 0.035	0.928 ± 0.025	0.364 ± 0.080	0 ± 0	0.00	0.238
		PRIL-RAMP	0.638 ± 0.083	0.923 ± 0.007	0.307 ± 0.012	55 ± 5	13.75	0.318

when noise is added, PRIL-RAMP and K-PRIL-RAMP outperform Kernel PRIL across all levels of noise. In other words, PRIL-RAMP and K-PRIL-RAMP are more robust to noise.

Overall, the noise-insensitive PRIL-RAMP and K-PRIL-RAMP algorithms significantly outperform basic PRIL and Kernel PRIL, which indicates that ramp loss-based algorithms can effectively handle noisy data.

The major difference between linear algorithms (PRIL and PRIL-RAMP) and nonlinear (Kernel PRIL and K-PRIL-RAMP) is that PRIL and PRIL-RAMP use a constant step size whereas nonlinear algorithms choose a more complex step size to find the new $\mathbf{w}$ and θ in every trial. In PRIL-RAMP and K-PRIL-RAMP, the step size is determined by solving an optimization problem of OGD whose ramp loss is a noise-resilient function, which makes the contributes of noise loss become 0 when there is a lot of noise, whereas PRIL and Kernel PRIL take all samples to ensure that the loss on the current example becomes 0. Thus, we see that the proposed PRIL-RAMP and K-PRIL-RAMP perform better

or comparable to other methodologies and are noise resistance.

3.4.2. Noise Statistical Study. We investigate the proposed noise-resilient online ordinal regression algorithm in the case of covariance variable noisy data by considering the order. Specifically, we attempt to apply the statistical measures to answer the question of how effective the proposed PRIL-RAMP method is in handling data with noise input. In this subsection, we have reported a number of simulation studies on finite-sample performance evaluation ($t = (1/3)T, t = (2/3)T, t = T$). Considering order and statistical measure, we estimate the influence relation of prediction MAE (mean square error) [22], RMSE (root mean square error), Spearman's correlation coefficient, discarded sample, discarded rate, and run time between PRIL, PRIL-RAMP, and PA-I [10] for different noisy data. $MAE = (1/T) \sum_{t=1}^T |\mathcal{O}(\hat{y}^t) - \mathcal{O}(y^t)|$ and $RMSE = \sqrt{(1/T) \sum_{t=1}^T (\mathcal{O}(\hat{y}^t) - \mathcal{O}(y^t))^2}$, where $\mathcal{O}(\hat{y}^t)$ is predicted

TABLE 3: Sensitivity estimation of PRIL, PA-I, and proposed PRIL-RAMP algorithm of three datasets under 25% noise data.

Dataset	Rounds	Method	MAE	RMSE	Spearman's correlation coefficient	Discard sample	Discard rate	Time (s)
Abalone	T = 1400	PA-I-25%	1.297 ± 0.143	1.639 ± 0.686	0.129 ± 0.030	0 ± 0	0.00	0.035
		PRIL-25%	0.849 ± 0.286	1.247 ± 0.056	0.262 ± 0.053	0 ± 0	0.00	0.176
		PRIL-RAMP-25%	0.877 ± 0.143	1.273 ± 0.035	0.152 ± 0.012	301 ± 11	21.5	0.225
	T = 2800	PA-I-25%	1.319 ± 0.086	1.660 ± 0.068	0.194 ± 0.043	0 ± 0	0.00	0.069
		PRIL-25%	0.788 ± 0.057	1.161 ± 0.089	0.278 ± 0.031	0 ± 0	0.00	0.368
		PRIL-RAMP-25%	0.788 ± 0.092	1.144 ± 0.048	0.202 ± 0.075	645 ± 6	23.04	0.432
	T = 4100	PA-I-25%	1.298 ± 0.011	1.632 ± 0.004	0.230 ± 0.158	0 ± 0	0.00	0.105
		PRIL-25%	0.758 ± 0.021	1.122 ± 0.096	0.287 ± 0.031	0 ± 0	0.00	0.552
		PRIL-RAMP-25%	0.752 ± 0.064	1.082 ± 0.067	0.235 ± 0.012	1314 ± 34	32.04	0.623
Parkinsons-uprds	T = 2000	PA-I-25%	1.028 ± 0.000	1.350 ± 0.185	0.199 ± 0.060	0 ± 0	0.00	0.082
		PRIL-25%	0.964 ± 0.005	1.283 ± 0.016	0.148 ± 0.097	0 ± 0	0.00	0.297
		PRIL-RAMP-25%	0.829 ± 0.005	1.173 ± 0.669	0.407 ± 0.035	425 ± 21	21.25	0.382
	T = 4000	PA-I-25%	0.935 ± 0.075	1.235 ± 0.013	0.225 ± 0.095	0 ± 0	0.00	0.296
		PRIL-25%	0.907 ± 0.000	1.212 ± 0.023	0.206 ± 0.075	0 ± 0	0.00	0.548
		PRIL-RAMP-25%	0.709 ± 0.075	1.041 ± 0.033	0.503 ± 0.018	768 ± 8	19.20	0.786
	T = 5800	PA-I-25%	0.912 ± 0.066	1.206 ± 0.082	0.235 ± 0.010	0 ± 0	0.00	0.385
		PRIL-25%	0.903 ± 0.027	1.209 ± 0.039	0.210 ± 0.083	0 ± 0	0.00	0.854
		PRIL-RAMP-25%	0.690 ± 0.078	1.012 ± 0.069	0.520 ± 0.031	1456 ± 23	25.10	1.127
Real estate valuation	T = 140	PA-I-25%	0.721 ± 0.029	0.721 ± 0.021	0.368 ± 0.023	0 ± 0	0.00	0.421
		PRIL-25%	0.521 ± 0.045	0.854 ± 0.074	0.354 ± 0.041	0 ± 0	0.00	0.162
		PRIL-RAMP-25%	1.028 ± 0.057	1.357 ± 0.051	0.003 ± 0.001	32 ± 1	22.85	0.208
	T = 280	PA-I-25%	0.628 ± 0.071	0.910 ± 0.029	0.470 ± 0.091	0 ± 0	0.00	0.923
		PRIL-25%	0.521 ± 0.045	0.854 ± 0.041	0.354 ± 0.013	0 ± 0	0.00	0.348
		PRIL-RAMP-25%	0.746 ± 0.029	1.087 ± 0.246	0.111 ± 0.033	56 ± 3	20.00	0.378
	T = 400	PA-I-25%	0.644 ± 0.068	0.907 ± 0.032	0.498 ± 0.005	0 ± 0	0.00	1.263
		PRIL-25%	0.521 ± 0.084	0.854 ± 0.041	0.354 ± 0.013	0 ± 0	0.00	0.531
		PRIL-RAMP-25%	0.677 ± 0.184	0.998 ± 0.076	0.101 ± 0.096	121 ± 5	30.25	0.579

TABLE 4: Sensitivity estimation of PRIL, PA-I, and proposed PRIL-RAMP algorithm of three datasets under 50% noise data.

Dataset	Rounds	Method	MAE	RMSE	Spearman's correlation coefficient	Discard sample	Discard rate	Time (s)
Abalone	T = 1400	PA-I-50%	1.372 ± 0.413	1.743 ± 0.076	0.027 ± 0.095	0 ± 0	0.00	0.296
		PRIL-50%	1.937 ± 0.078	2.271 ± 0.021	0.033 ± 0.008	0 ± 0	0.00	0.217
		PRIL-RAMP-50%	1.094 ± 0.28	1.526 ± 0.434	0.098 ± 0.001	423 ± 11	30.21	0.817
	T = 2800	PA-I-50%	1.369 ± 0.064	2.661 ± 0.036	0.734 ± 0.051	0 ± 0	0.00	0.558
		PRIL-50%	0.600 ± 0.357	0.871 ± 0.098	0.671 ± 0.020	0 ± 0	0.00	0.425
		PRIL-RAMP-50%	0.732 ± 0.143	1.057 ± 0.096	0.538 ± 0.071	956 ± 6	34.14	0.369
	T = 4100	PA-I-50%	1.369 ± 0.064	1.695 ± 0.089	0.082 ± 0.025	0 ± 0	0.00	0.805
		PRIL-50%	1.937 ± 0.078	2.262 ± 0.026	0.010 ± 0.006	0 ± 0	0.00	0.597
		PRIL-RAMP-50%	0.941 ± 0.029	1.336 ± 0.084	0.014 ± 0.009	1743 ± 47	42.51	0.551

TABLE 4: Continued.

Dataset	Rounds	Method	MAE	RMSE	Spearman's correlation coefficient	Discard sample	Discard rate	Time (s)
Parkinsons-uprds	$T = 2000$	PA-I-50%	1.704 ± 0.000	2.030 ± 0.025	0.018 ± 0.006	0 ± 0	0.00	0.094
		PRIL-50%	1.168 ± 0.005	1.482 ± 0.059	0.044 ± 0.007	0 ± 0	0.00	0.345
		PRIL-RAMP-50%	1.101 ± 0.010	1.432 ± 0.082	0.112 ± 0.024	942 ± 34	47.10	0.333
	$T = 4000$	PA-I-50%	1.603 ± 0.025	1.932 ± 0.038	0.043 ± 0.006	0 ± 0	0.00	0.128
		PRIL-50%	1.096 ± 0.025	1.410 ± 0.053	0.087 ± 0.003	0 ± 0	0.00	0.670
		PRIL-RAMP-50%	0.961 ± 0.004	1.288 ± 0.992	0.186 ± 0.049	1876 ± 78	46.90	0.628
	$T = 5800$	PA-I-50%	1.510 ± 0.055	1.849 ± 0.089	0.057 ± 0.008	0 ± 0	0.00	0.273
		PRIL-50%	1.097 ± 0.065	1.405 ± 0.055	0.084 ± 0.027	0 ± 0	0.00	0.978
		PRIL-RAMP-50%	1.757 ± 0.143	2.021 ± 0.131	0.144 ± 0.059	2745 ± 109	47.32	0.994
Real estate valuation	$T = 140$	PA-I-50%	1.028 ± 0.071	1.276 ± 0.155	0.005 ± 0.008	0 ± 0	0.00	0.159
		PRIL-50%	1.980 ± 0.058	1.286 ± 0.006	0.013 ± 0.007	0 ± 0	0.00	0.168
		PRIL-RAMP-50%	1.182 ± 0.143	1.548 ± 0.004	0.095 ± 0.077	46 ± 11	32.85	0.178
	$T = 280$	PA-I-50%	1.042 ± 0.857	1.320 ± 0.173	0.034 ± 0.008	0 ± 0	0.00	0.324
		PRIL-50%	0.980 ± 0.083	1.286 ± 0.006	0.013 ± 0.005	0 ± 0	0.00	0.346
		PRIL-RAMP-50%	0.973 ± 0.301	1.050 ± 0.117	0.071 ± 0.066	93 ± 3	33.21	0.362
	$T = 400$	PA-I-50%	1.159 ± 0.806	1.418 ± 0.487	0.031 ± 0.065	0 ± 0	0.00	0.475
		PRIL-50%	0.980 ± 0.083	1.286 ± 0.006	0.031 ± 0.006	0 ± 0	0.00	0.517
		PRIL-RAMP-50%	0.843 ± 0.005	1.028 ± 0.004	0.034 ± 0.011	167 ± 11	41.75	0.537

TABLE 5: Sensitivity estimation of PRIL, PA-I, and proposed PRIL-RAMP algorithm of three datasets under 75% noise data.

Dataset	Rounds	Method	MAE	RMSE	Spearman's correlation coefficient	Discard sample	Discard rate (%)	Time (s)
Abalone	$T = 1400$	PA-I-75%	1.832 ± 0.857	2.190 ± 0.023	-0.164 ± 0.082	0 ± 0	0.00	0.206
		PRIL-75%	1.177 ± 0.085	1.563 ± 0.193	-0.217 ± 0.053	0 ± 0	0.00	0.190
		PRIL-RAMP-75%	1.172 ± 0.143	1.578 ± 0.078	-0.170 ± 0.012	856 ± 54	61.14	0.230
	$T = 2800$	PA-I-75%	1.720 ± 0.000	2.083 ± 0.061	-0.135 ± 0.043	0 ± 0	0.00	0.368
		PRIL-75%	1.072 ± 0.857	1.450 ± 0.616	-0.179 ± 0.696	0 ± 0	0.00	0.350
		PRIL-RAMP-75%	1.070 ± 0.092	1.447 ± 0.017	-0.085 ± 0.024	1745 ± 204	64.78	0.528
	$T = 4100$	PA-I-75%	1.767 ± 0.072	2.108 ± 0.086	-0.127 ± 0.003	0 ± 0	0.00	0.624
		PRIL-75%	1.011 ± 0.073	1.385 ± 0.907	-0.147 ± 0.029	0 ± 0	0.00	0.575
		PRIL-RAMP-75%	1.009 ± 0.039	1.375 ± 0.067	-0.028 ± 0.025	2656 ± 98	64.78	0.681
Parkinsons-uprds	$T = 2000$	PA-I-75%	2.105 ± 0.004	2.403 ± 0.123	0.050 ± 0.011	0 ± 0	0.00	0.273
		PRIL-75%	1.347 ± 0.005	1.650 ± 0.045	0.036 ± 0.088	0 ± 0	0.00	0.288
		PRIL-RAMP-75%	1.313 ± 0.003	1.619 ± 0.077	0.192 ± 0.048	1456 ± 59	72.80	0.352
	$T = 4000$	PA-I-75%	2.135 ± 0.075	2.455 ± 0.053	0.046 ± 0.051	0 ± 0	80.00	0.587
		PRIL-75%	1.295 ± 0.006	1.599 ± 0.068	0.049 ± 0.075	0 ± 0	0.00	0.365
		PRIL-RAMP-75%	1.302 ± 0.005	1.608 ± 0.104	0.249 ± 0.011	2878 ± 245	871.95	0.703
	$T = 5800$	PA-I-75%	2.096 ± 0.187	2.436 ± 0.033	0.057 ± 0.064	0 ± 0	0.00	0.899
		PRIL-75%	1.285 ± 0.714	1.586 ± 0.054	0.053 ± 0.129	0 ± 0	0.00	0.928
		PRIL-RAMP-75%	1.300 ± 0.017	1.599 ± 0.794	0.277 ± 0.098	3799 ± 324	68.56	1.036

TABLE 5: Continued.

Dataset	Rounds	Method	MAE	RMSE	Spearman's correlation coefficient	Discard sample	Discard rate (%)	Time (s)
Real estate valuation	$T=140$	PA-I-75%	1.471 ± 0.029	1.643 ± 0.168	0.068 ± 0.012	0 ± 0	0.00	0.124
		PRIL-75%	1.657 ± 0.067	1.976 ± 0.067	0.128 ± 0.045	0 ± 0	0.00	0.180
		PRIL-	1.528 ± 0.071	1.784 ± 0.057	0.455 ± 0.004	91 ± 23	65.00	0.144
		RAMP-75%	1.439 ± 0.286	1.617 ± 0.079	0.047 ± 0.008	0 ± 0	30.00	0.268
	$T=280$	PRIL-75%	1.657 ± 0.067	1.976 ± 0.066	0.128 ± 0.015	0 ± 0	0.00	0.388
		PRIL-	1.246 ± 0.029	1.531 ± 0.086	0.444 ± 0.069	187 ± 56	66.78	0.270
		RAMP-75%	1.455 ± 0.206	1.632 ± 0.062	-0.013 ± 0.003	0 ± 0	0.00	0.595
		PRIL-75%	1.657 ± 0.077	1.976 ± 0.076	0.128 ± 0.015	0 ± 0	0.00	0.516
	$T=400$	PRIL-	1.016 ± 0.184	1.339 ± 0.087	0.439 ± 0.034	302 ± 34	75.50	0.402
		RAMP-75%

FIGURE 10: Sensitivity analysis result between MAE and discard rate. (a) Abalone. (b) Parkinsons-updrs. (c) Real estate valuation.

rank and $\mathcal{O}(y^t)$ is true one. The value ranges from 0 to $K - 1$ (maximum deviation in the number of categories). It suggests that the larger the values are, the worse the prediction accuracy is. In ordinal matching, a useful and common measure is Spearman's rank correlation which measures the correspondence in rank terms of the two distributions. It measures the structural similarity and values range from 0 to 1. The higher the value, the closer it is. The discard sample denotes the samples in the model that has no updating effect, namely, samples that make $z_i^T(\mathbf{w}^t \cdot \mathbf{x}^t - \theta_i^t) < s$ in each algorithm iteration. We calculate the number of discard samples by tracking the value of s . The discard rate is the percentage of the discard sample and the total. For each

dataset, we have four noise scenarios to discuss: 0%, 25%, 50%, and 75%.

The results are summarized in Tables 2–5, and as can be seen from the columns of MAE and RMSE, the proposed PRIL-RAMP method outperforms the competing PRIL and PA-I in four noisy data cases. Especially, in the case of a high level (50% and 75%), PRIL-RAMP significantly outperforms based on the algorithm PRIL. Spearman's correlation coefficient shows that the correspondence of PRIL-RAMP in ranking terms of the two distributions is greater in three cases of noise. Empirically, as we increase the ratio of noise, the difference between the two methods in each evaluation index becomes apparent. Due to the intrinsic flaw of hinge loss, the PA-I method is highly sensitive

to noise. Ramp loss can effectively reduce the impact of noise data in some sense, though except for individual data. Besides, as we increase our sample, the number of discarded samples increases with the increase of noise, compared with PRIL and PA-I. Most of the samples are discarded when the noise level is 75%. Not surprisingly, the percentage of the noisy data significantly affects the discarded rate and the results of the algorithms. Empirically, the discarded rate is approximately equal to the noise ratio. Moreover, the estimation of small samples can lead to the instability of the model because the prediction accuracy depends on how much proportion of noise points is used to train the model. It indicates that PRIL-RAMP is a better noise-resilient method than PA-I to deal with noisy data, though the latter is more faster.

3.4.3. Noise Sensitivity Comparison. In order to compare the trend charts of MAE and discard rate, we have plotted the 2D performance variation under different sets of noise in Figure 10. As can be observed, the proposed PRIL-RAMP outperforms the two competing methods (PA-I and PRIL) significantly in the case of high-level noisy data. Because of the perceptron loss, original PRIL is very sensitive to noise data. However, MAE of PA-I is higher than PRIL, because the Hinge loss function is more sensitive. Moreover, MAE will gradually increase and stabilize with increasing noise ratio. Specifically, when the noise ratio is greater than 50%, most of the noisy samples are discarded, so the MAE will gradually stabilize though it increases with the increase of noise data. Furthermore, due to the discard samples in the case of high-level noisy data, PRIL-RAMP can reduce the impact of noise data obviously.

4. Conclusion and Future Works

In this paper, we studied online learning from noisy data streams. First, we have proposed ramp loss-based ordinal regression models PRIL-RAMP and its nonlinear variant K-PRIL-RAMP to incrementally predict the interval labels in noisy environments in an online manner. Moreover, relevant theoretical analysis of the thresholds order preservation has been carried out, which shows that the proposed ramp loss is noise-resilient in the scenarios of interval label(s). At last, experimental studies have shown that PRIL-RAMP and K-PRIL-RAMP are robust and can be used to deal with noisy data streams. Future work will involve extending the ordinal regression model to the PA model by introducing of convex loss function [12] and based on convex optimization theory [23].

Data Availability

Datasets used to support the findings of this study can be available form (<http://archive.ics.uci.edu/ml/>).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant nos. 71961004, 71461005, and 71561008, the Thesis Cultivation Project of GUET Graduate Excellent Master under grant no. 2019YJSPY01, and the Innovation Project of GUET Graduate Education under grant no. 2019YCXS081.

References

- [1] J. D. Rennie and N. Srebro, "Loss functions for preference levels: regression with discrete ordered labels," *Proceedings of The IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, vol. 11, pp. 180–186, 2005.
- [2] A. Guisan and F. E. Harrell, "Ordinal response regression models in ecology," *Journal of Vegetation Science*, vol. 11, no. 5, pp. 617–626, 2000.
- [3] O. M. Doyle, E. Westman, A. F. Marquand et al., "Predicting progression of alzheimer's disease using ordinal regression," *Plos One*, vol. 9, no. 8, pp. 1–10, 2014.
- [4] H. Ahn and K. J. Kim, "Corporate credit rating using multiclass classification models with order information. World Academy of Science, Engineering and Technology," *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineerin*, vol. 5, no. 12, pp. 1783–1788, 2011.
- [5] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 42, no. 2, pp. 109–127, 1980.
- [6] W. Chu and S. S. Keerthi, "New approaches to support vector ordinal regression," in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 145–152, New York, NY, USA, 2005.
- [7] A. Shashua and A. Levin, "Ranking with large margin principle: two approaches," *Advances in Neural Information Processing Systems*, vol. 42, pp. 961–968, 2003.
- [8] B. Gu, X. Geng, W. L. Shi et al., "Solving large-scale support vector ordinal regression with asynchronous parallel coordinate descent algorithms," *Pattern Recognition*, vol. 109, pp. 1–12, 2020.
- [9] A. Berg, M. Oskarsson, and M. Oconnor, "Deep ordinal regression with label diversity," 2020.
- [10] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," *Advances in Large Margin Classifiers*, vol. 10, pp. 115–132, 2000.
- [11] P. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer, "Online passive-aggressive algorithms," *Journal of Machine Learning Research*, vol. 7, pp. 551–585, 2006.
- [12] N. Manwani and M. Chandra, "Exact passive-aggressive algorithms for learning to rank using interval labels," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 10, pp. 19–38, 2018.
- [13] K. Crammer and Y. Singer, "Pranking with ranking," in *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, vol. 14, pp. 641–647, New York, NY, USA, 2001.
- [14] F. Edward and Harrington, "Online ranking/collaborative filtering using the perceptron algorithm," in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, pp. 250–257, New York, NY, USA, 2003.
- [15] N. Manwani, "PRIL: perceptron ranking using interval labels," in *Proceedings of the ACM India Joint International*

- Conference on Data Science and Management of Data*, pp. 1769–1803, New York, NY, USA, 2019.
- [16] L. Jian, F. Gao, P. Ren, Y. Song, and S. Luo, “A noise-resilient online learning algorithm for scene classification,” *Remote Sensing*, vol. 10, no. 11, pp. 1–17, 2018.
 - [17] X. Liang, Z. H. Xia, L. Jian, X. Niu, and A. Link, “An adaptive classification model for peptide identification,” *BMC Genomics*, vol. 16, no. 16, p. S1, 2015.
 - [18] H. Wang, S. Lu, and Z. Zhou, “Ramp loss for twin multi-class support vector classification,” *International Journal of Systems Science*, vol. 17, no. 1, pp. 1–16, 2020.
 - [19] T. Lipp and S. Boyd, “Variations and extension of the convex-concave procedure,” *Optimization and Engineering*, vol. 17, no. 2, pp. 263–287, 2016.
 - [20] C. Blake and C. J. Merz, *UCI Repository of Machine Learning Databases*, Department of Information and Computer Science, University of California, Irvine, CA, USA, 2016.
 - [21] X. Wang, Y. Jiang, M. Huang, and H. Zhang, “Robust variable selection with exponential squared loss,” *Journal of the American Statistical Association*, vol. 108, no. 502, pp. 632–643, 2013.
 - [22] P. A. Gutierrez, M. P. Ortiz, J. S. Monedero, F. F. Navarro, and C. H. Martinez, “Ordinal regression methods: survey and experimental study,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 127–146, 2015.
 - [23] X. Gao, X. Li, and S. Zhang, “Online learning with non-convex losses and non-stationary regret,” in *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, vol. 84, New York, NY, USA, 2018.