PRIL: Perceptron Ranking Using Interval Labels

by

Naresh Manwani

in

The ACM India Joint International Conference on Data Science Management of Data
(\textit{CoDS-COMAD-2019})

Swissotel, Kolkata, India

Report No: IIIT/TR/2019/-1

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
January 2019
PRIL: Perceptron Ranking Using Interval Labels

Naresh Manwani
International Institute of Information Technology, Hyderabad
Telangana, India
naresh.manwani@iiit.ac.in

ABSTRACT
In this paper, we propose an online learning algorithm called PRIL for learning ranking classifiers using interval labeled data. We show the correctness of PRIL by showing that it preserves the orderings of the thresholds in successive trials. We show that the proposed algorithm converges in finite number of steps if there exists an ideal classifier. We also give the mistake bound for the general case and provide $O(\sqrt{T})$ regret bound for the proposed algorithm. We show the effectiveness of PRIL by comparing its performance with other approaches.

KEYWORDS
Online Learning, Ranking, Interval Labels

1 INTRODUCTION
In ranking (ordinal regression), the objective is to learn a model to predict labels from an ordinal scale (labels from a discrete but ordered set). This arises when target variable consists of labels such that there is an ordering among them. For example (“worst” < “bad” < “ok” < “good” < “very good”) in product ratings provided by online retail stores (e.g., Amazon, eBay etc.). Ordinal regression has been successfully used in many diverse areas e.g., predicting the severity of Alzheimer disease [5], collaborative filtering [10], ecology [6] etc.

Ordinal regression is different from multi-class classification problem in the sense that there is an ordering among the class labels. While these numbers can be thought of as class labels, there is also an ordering which has to be taken care. On the other hand, this problem is quite different from regression in the sense that target variable can take only discrete values. If we think of product rating example by online retail stores, if the true rating of a product is “worst”, it is preferable to predict “bad” as compared to “ok”. Ordinal regression takes care of this ordering by ensuring that predictions farther from the true label incur a larger penalty than those closer to the true label.

In general, an ordinal regression requires a linear function and a set of $K - 1$ thresholds ($K$ be the number of classes) to determine the target variable. Each threshold corresponds to a class. Thus, the thresholds should have the same order as their corresponding classes. Ordinal regression decides the rank (class) based on the relative position of the linear function value with respect to different thresholds. One can learn a non-linear boundaries also by using an appropriate nonlinear transformation using kernel functions. Large margin formulations for ordinal regression are proposed in Chu and Keerthi [2], Shashua and Levin [12]. Ordering of thresholds can be maintained implicitly or explicitly. In explicit methods [2, 12], ordering is preserved by forcing it explicitly in the optimization problem. Implicit methods capture the ordering by posing separability conditions between every pair of classes [2, 7]. The approaches discussed so far are batch algorithms for ordinal regression. Crammer and Singer [3] proposed an online learning algorithm for ordinal regression which has Perceptron [11] like update equations.

In these algorithms, training data has correct class label for each feature vector. However, in many cases, we don’t have the exact labels for examples. Instead, we may have an interval in which the true label lies. In practice, whenever we get labels from multiple labelers (e.g., product ratings in e-commerce sites, movie ratings, degree of sentiment etc.), we actually get a full range of ranking values for each observation. Another reason for such interval labels could be the insufficient knowledge of the labeler. For example, in an online retail store, rating provided by labelers can be [1,3]. This means that the true rating of the product is one of the value among 1, 2 and 3. Note that this setting is different from the setting in which we have to predict the relative order of a sequence of instances [9]. The proposed approach uses the whole range as a label and learns an efficient ranking classifier.

In this paper, we propose an online algorithm for ordinal regression using interval labeled data. We name the proposed approach as PRIL (Perceptron ranking using interval labels). As per our knowledge, this is the first ever online algorithm for ordinal regression which uses interval labels. We show the correctness of the algorithm by showing that after each iteration, the algorithm maintains the orderings of the thresholds. In the ideal setting, we show that the algorithm stops after making finite number of mistakes. We also derive the mistake bounds for general setting. We also show that the regret for the algorithm scales only as $O(\sqrt{T})$.

The rest of the paper is organized as follows. In section 2, we describe the ordinal regression problem using interval labeled data. In section 3, we propose our online algorithm for ordinal regression using interval labeled data. We provide the correctness proof and mistake bounds for the proposed algorithm in section 3.2. We present the experimental results in section 4. We make the conclusions in section 5.
2 RANKING USING INTERVAL LABELED DATA

Let $X \subseteq \mathbb{R}^d$ be the instance space. Let $Y = \{1, \ldots, K\}$ be the ordered label space. Our objective is to learn ordinal regression model $h : X \rightarrow Y$ which has the following form

$$h(x) = 1 + \sum_{k=1}^{K-1} I_{[w \cdot x - \theta_k < 0]} \leq 0 \right]$$

where $w \in \mathbb{R}^d$, $\theta = [\theta_1 \ldots \theta_K]$ be the parameters to be optimized. We assume $\theta_K = \infty$. Thus, we assume $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_K$. The classifier splits the real line into $K$ consecutive intervals using thresholds $\theta_1, \ldots, \theta_K$ and then decides the class label based on which interval corresponds to $w \cdot x$.

We assume that for each example $x$, the annotator provides an interval $[y_l, y_r] \in Y \times Y$, where $y_l \leq y_r$. The interval annotation means that the true label $y$ for example $x$ lies in the interval $[y_l, y_r]$. Let $S = \{(x_1, y_{l1}, y_{r1}), \ldots, (x_I, y_{lI}, y_{rI})\}$ be the training set.

Discrepancy between the predicted label and corresponding label interval can be measured using interval insensitive loss [1].

$$L_{IMC}^I(f(x), y_l, y_r, \theta) = \sum_{i=1}^{I} \max(0, f(x) - \theta_i)$$

Here $IMC$ stands for implicit constraints ordering of thresholds $\theta_i$. For a given example-interval pair $(x_i, (y_l, y_r))$, the loss $L_{IMC}^I(f(x), y_l, \theta)$ becomes zero only when

$$f(x) - \theta_i \geq 0 \quad \forall i \in \{1, \ldots, y_l - 1\}$$

$$f(x) - \theta_i \leq 0 \quad \forall i \in \{y_r, \ldots, K\}. $$

Let $I = \{1, \ldots, y_l - 1\} \cup \{y_r, \ldots, K\}$. Then, we define $z_i$, $\forall i \in I$ as follows.

$$z_i = I_{[1, \ldots, y_l - 1]} - I_{[y_r, \ldots, K]}$$

where $z_i$ is an indicator function which takes value 1 if the argument is true. Thus, $L_{IMC}^I(f(x), y_l, y_r, \theta) = 0$ requires that $z_i(f(x) - \theta_i) \geq 0$, $\forall i \in I$. Thus, $L_{IMC}^I$ can be re-written as

$$L_{IMC}^I(f(x), y_l, y_r, \theta) = \sum_{i \in I} \max(0, -z_i(f(x) - \theta_i))$$

3 PERCEPTRON RANKING USING INTERVAL LABELED DATA

In this section, we propose an online algorithm for ranking (ordinal regression) using interval insensitive loss described in eq. (2). Our algorithm is based on stochastic gradient descent on $L_{IMC}^I$.

We derive the algorithm for linear classifier. Which means, $f(x) = w \cdot x$. Thus, the parameters to be estimated are $w$ and $\theta$. We initialize with $w^0 = 0$ and $\theta^0 = 0$. Let $w^t, \theta^t$ be the estimates of the parameters in the beginning of trial $t$. Let at trial $t$, $x^t$ be the example observed and $[y_{l}^t, y_{r}^t]$ be its label interval. $w^{t+1}$ and $\theta^{t+1}$ are found as follows.

$$w^{t+1} = w^t - \eta \nabla_w L_{IMC}^I(f(x^t, y_l^t, y_r^t), \theta_{w, \theta}^t)$$

$$\theta_{w, \theta}^{t+1} = \theta_{w, \theta}^t - \eta \frac{\partial L_{IMC}^I(f(x^t, y_l^t, y_r^t), \theta_{w, \theta}^t)}{\partial \theta_i}$$

Thus, only those constraints will participate in the update which are not satisfied. The violation of $i$th constraint leads to the update contribution of $z_i^t$ in $w^{t+1}$ and $z_i^t$ in $\theta^{t+1}$. $t \not\in I^t$. The complete algorithm is described in Algorithm 1. It is important to see that when exact labels are given to the Algorithm 1 instead of partial labels, it becomes same as the algorithm proposed in [3]. Now we describe how to extend PRIL for learning nonlinear classifiers using kernel methods.

3.1 Kernel PRIL

We can easily extend the proposed PRIL for learning nonlinear classifiers using kernel functions. We see that the classifier learnt after $t$ trials using PRIL can be completely determined using $t_s^t$, $i \in [K], s \in [t]$. We see that $w^{t+1} = \sum_{s=1}^{t_s^t} \sum_{i \in I^t} t_s^t x_i^t$ and $\theta_{w, \theta}^{t+1} = \theta_{w, \theta}^t - t_s^t$, $i \in [K]$. Also, $f^{t+1}(x)$ can be found as $f^{t+1}(x) = \sum_{s=1}^{t_s^t} \sum_{i \in I^t} t_s^t x_i^t \cdot x$. Thus, we can replace the inner product with a suitable kernel function $K : X \times X \rightarrow \mathbb{R}$ and represent $f^{t+1}(x)$ as

$$f^{t+1}(x) = \sum_{s=1}^{t_s^t} \sum_{i \in I^t} t_s^t K(x_i^t, x)$$

Algorithm 1 Perceptron Ranking using Interval Labels (PRIL).

Input: Training Dataset $S$, $\eta$

Initialize Set $t = 1$, $w^0 = \theta^0 = \ldots = \theta_{K-1} = 0$, $\theta_K = \infty$

for $i \leftarrow 1$ to $t$

Get example $x_i$ and its $(y_{l}^i, y_{r}^i)$

Set $z_i^t = +1, i \in \{1, \ldots, y_{l}^i - 1\}$

Set $z_i^t = -1, i \in \{y_{r}^i, \ldots, K\}$

Initialize $t_s^t = 0, i \in [K]$ for $i \in [K]$

end for

if $z_i^t (w^t \cdot x^t - \theta_i^t) \leq 0$ then

$\theta_i^t = z_i^t$

else

end for

Output: $h(x) = \min_{i \in [K]} \{i : w^t \cdot x^t - \theta_i^t < 0\}$
The ordinal classifier learnt after $T$ trials is

$$h(x) = \min_{i \in [K]} \left( \left\{ i : \sum_{t=1}^{T} \sum_{j \in T} r_t^i \mathcal{K}(x^i, x) - \theta_t^{i+1} < 0 \right\} \right)$$

Complete description of Kernel PRIL is provided in Algorithm 2.

**Algorithm 2 Kernel PRIL**

**Input:** Training Dataset $\mathcal{S}$

**Output:** $t_1^0, \ldots, t_K^{T-1}$, $t = 1 \ldots T$

**Initialize:** Set $t = 0, \theta^K_0 = 0, i \in [K-1]$, $\theta^K_i = \infty, t_i^0 = 0, i \in [K]; f^0() = 0$

for $t \leftarrow 1$ to $N$ do

Get example $x^t$ and its $(y^t_1, y^t_2)$;

Set $x^t_0 = +1, i \in \{1, \ldots, y^t_2 - 1\}$;

Set $x^t_0 = -1, i \in \{y^t_2, \ldots, K-1\}$;

Initialize $t_i^t = 0, \forall i \in [K]$;

for $i \in I^t$ do

if $z_t^i(f^t(x^t) - \theta^K_i) \leq 0$ then

$t_i^t = t_i^t$;

end if

end for

$f^{t+1}(x) = f^t() + \sum_{i \in I^t} r_t^i \mathcal{K}(x^i, . )$

$\theta^K_{t+1} = \theta^K_t + t_i^t, \forall i \in [K]$;

end for

$h(x) = \min_{i \in [K]} \left( \left\{ i : f^{T+1}(x) - \theta^{T+1}_t < 0 \right\} \right)$

**3.2 Analysis**

Now we will show that PRIL preserves the ordering of the thresholds $\theta_1, \ldots, \theta_{K-1}$.

**Lemma 3.1. Order Preservation:** Let $w^t$ and $\theta^t_i \leq \theta^t_{i+1} \leq \ldots \leq \theta^t_2 \leq \ldots \leq \theta^t_K$ be the current parameters for ranking classifier. Let $x^t$ be the instance at trial $t$ and $(y^t_1, y^t_2)$ be its corresponding rank interval.

Let $\theta^{t+1}_i = \theta^t_i + \theta^K_{i+1} - \theta^K_{i}$ be the updated threshold parameters after trial $t$ using PRIL. Then, $\theta^{t+1}_1 \leq \theta^{t+1}_2 \leq \ldots \leq \theta^{t+1}_K$.

Proof of above lemma is given in Appendix A.

We now show that the PRIL makes finite number of mistakes if there exists an ideal interval ranking classifier. An ideal classifier is the one for which the predicted label always lies in the true label interval.

**Theorem 3.2.** Let $\mathcal{S} = \{(x^1, y^1_1, y^1_2), \ldots, (x^T, y^T_1, y^T_2)\}$ be an input sequence to PRIL. Let $R^2 = \max_{i \in [T]} ||x^i||^2$ and $c = \min_{i \in [T]} (y^i_2 - y^i_1)$.

Let $\exists \gamma > 0, w^t \in \mathbb{R}^d$ and $\theta^K \in \mathbb{R}^{K-1}$ such that $||w^*||^2 + ||\theta^K||^2 = 1$ and $\min_{i \in [T]} z_t^i (w^t \cdot x^t - \theta^K_i) \geq \gamma, \forall t \in [T]$. Then, PRIL makes finite number of mistakes as follows.

$$\sum_{t=1}^{T} L_{MAE}^t(f^t(x^t), \theta^t, y^t_1, y^t_2) \leq \frac{(R^2 + 1)(K - c - 1)}{2\gamma}$$

where $f^t(x^t) = w^t \cdot x^t$.

The mistake bound proof of PRIL in ideal case is given in Appendix B. Note that when we have single label for every example $c = 0$, then the mistake bound above will become $\frac{(K-1)(R^2 + 1)}{\gamma^2}$, which is the same bound as discussed in [3]. In Theorem 3.2, we assumed that there exists an ideal classifier defined by $w^* \in \mathbb{R}^d$ and $\theta^K$.

Let $v^t = \{w^t, \theta^K, f^t(x^t) = w^t \cdot x^t\}$. Thus,

$$L_{MAE}^t(f^t(x^t), \theta^K, y^t_1, y^t_2) = 0, \forall t \in [T]$$

Which means, $z_t^i(v^t \cdot x^t - \theta^K_i) \geq 0, \forall i \in I^t, \forall t \in [T]$ where $z_t^i, i \in I^t$ are as described in eq. (3). Now we define $\mathcal{x}^t_i \in \mathbb{R}^{d+K-1}, \forall i \in I^t$ as follows.

$$\mathcal{x}^t_i = (x^t_1 \ldots 0 \ldots 1 \ldots 0)$$

where the component values at locations $d + 1, \ldots, d + K - 1$ are all set to ‘0’ except for the location $(d + i)$. Component value at location $(d + i)$ is set to ‘-1’ in $\mathcal{x}^t_i$. Thus, we have $z_t^i(v^t \cdot \mathcal{x}^t_i) \geq 0, \forall i \in I^t, \forall t \in [T]$. Thus, $v^t$ correctly classifies all the $\mathcal{x}^t_i, \forall i \in I^t, \forall t \in [T]$.

However, in general, for a given dataset, we may not know if such an ideal classifier exists. Next, we derive the mistake bound for this general setting.

**Theorem 3.3.** Let $\mathcal{S} = \{(x^1, y^1_1, y^1_2), \ldots, (x^T, y^T_1, y^T_2)\}$ be an input sequence to PRIL. Let $\gamma > 0, R^2 = \max_{i \in [T]} ||x^i||^2$ and $c = \min_{i \in [T]} (y^i_2 - y^i_1)$. Then, for any $w^t, \theta^K \in \mathbb{R}^{K-1}$ such that $||w||^2 + ||\theta^K||^2 = 1$, we get

$$\sum_{t=1}^{T} L_{MAE}^t(f^t(x^t), \theta^K, y^t_1, y^t_2) \leq \left( \frac{D + \sqrt{R^2 + 1})^2}{\gamma^2} \right)$$

where $K_1 = K - c - 1$, $f^t(x^t) = w^t \cdot x^t$, $D^2 = \sum_{t=1}^{T} \sum_{i \in I^t} (d^t_i)^2$, and $d^t_i = max(0, \gamma - z_t^i(w^t \cdot x^t - \theta^K_i)), i \in I^t, \forall t \in [T]$.

The proof of above theorem is provided in Appendix C.

**Proof.**

**Regret Analysis.** Let $\mathcal{S} = \{(x^1, y^1_1, y^1_2), \ldots, (x^T, y^T_1, y^T_2)\}$ be the input sequence. Let $(w^t, \theta^K)^{I^t}_{t=1}$ be the sequence of parameter vectors generated by an online ranking algorithm $\mathcal{A}$. Then, regret of algorithm $\mathcal{A}$ is defined as

$$R_T(\mathcal{A}) = \sum_{t=1}^{T} L_{MAC}^t(w^t \cdot x^t, \theta^K, y^t_1, y^t_2) - \sum_{t=1}^{T} L_{MAE}^t(w^t \cdot x^t, \theta^K, y^t_1, y^t_2)$$

For online gradient descent applied on convex cost functions, the regret bound analysis is given by Zinkevich [14]. We know that the objective function of PRIL is also convex. Motivated by that, here we find the regret bound for PRIL.

**Theorem 3.4.** Let $\mathcal{S} = \{(x^1, y^1_1, y^1_2), \ldots, (x^T, y^T_1, y^T_2)\}$ be an input sequence such that $x^t \in \mathbb{R}^d, \forall t \in [T]$. Let $R^2 = \max_{i \in [T]} ||x^i||^2$ and $c = \min_{i \in [T]} (y^i_2 - y^i_1)$. Let $\Omega = \{v \in \mathbb{R}^{d+K-1} : ||v||_2 \leq A\}$. Let $v^t = (w^t, \theta^K), t = 1 \ldots T$ be the sequence of vectors in $\Omega$ such that $\forall t \geq 1, v^{t+1} = v^t - \nabla v^t$, where $\nabla v^t$ belongs to the sub-gradient set of $L_{MAC}^t(w^t \cdot x^t, \theta^K, y^t_1, y^t_2)$ at $v^t$. Then,

$$R_T(\mathcal{PRIL}) \leq \Lambda \sqrt{R^2 + 1)(K - c - 1)}$$

The regret bound proof is provided in Appendix D.
4 EXPERIMENTS

We now discuss the experimental results to show the effectiveness of the proposed approach. We first describe the datasets used.

4.1 Dataset Description

We show the simulation results on 3 datasets. The description of these datasets are as given below.

(1) Synthetic Dataset: We generate points \( x \in \mathbb{R}^2 \) uniformly at random from the unit square \([0, 1]^2\). For each point, the rank was assigned from the set \( \{1, \ldots, 5\} \) as \( y = \max_i \{r_i \cdot 10(x_i - 0.5)(x_2 - 0.5) + \xi > b_i\} \) where \( b = (-\infty, -1, -0.1, 0.25, 1) \), \( \xi \sim \mathcal{N}(0, 0.125) \) (normally distributed with zero mean and a standard deviation of 0.125). The visualization of the synthetic dataset is provided in Figure 1. We generated 100 sequences of instance-rank pairs for synthetic dataset each of length 10000.

Figure 1: Synthetic Dataset

(2) Parkinson’s Telemonitoring Dataset: This dataset [8] contains biomedical voice measurements for 42 patients in various stages of Parkinson’s disease. There are total 5875 observations. There are 20 features and the target variable is total UPDRS. In the dataset, the total-UPDRS spans the range 7-55, with higher values representing more severe disability. We divide the range of total-UPDRS into 10 parts. We are looking for the nature of the labels even though it respects the orderings of the classes. We perform the evaluation criteria for PRIL.

(3) Abalone Dataset: The age of abalone [8] is determined by counting the number of rings. There are 8 features and 4177 observations. The number of rings vary from 1 to 29. However, the distribution is very skewed. So, we divide the whole range into 4 parts, namely 1-7, 8-9, 10-12, 13-29.

The training data is comprised of two parts. One part contains the absolute labels for feature vectors and the other contains the partial labels.

Generating Interval Labels: Let there be \( K \) categories. We consider two different methods of generating interval labels.

(1) Type1: For \( y \in \{2, \ldots, (K - 1)\} \), the interval label was randomly chosen between \([y - 1, y]\) and \([y, y + 1]\) where \( y \) is the true label. For the class label 1, the interval label was set to \([1, 2]\). For class label \( K \), the interval label was set to \([K - 1, K]\).

(2) Type2: For \( y \in \{2, \ldots, (K - 1)\} \), the interval labels were set to the interval \([y - 1, y + 1]\) where \( y \) is the true label. For the class label 1, the interval label was set to \([1, 2]\). For class label \( K \), the interval label was set to \([K - 1, K]\).

4.2 Kernel Functions Used

We used following kernel functions for different datasets.

- Synthetic: \( \mathcal{K}(x_1, x_2) = (x_1, x_2 + 1)^2 \)
- Parkinson’s Telemonitoring: \( \mathcal{K}(x_1, x_2) = x_1.x_2 \)
- Abalone: \( \mathcal{K}(x_1, x_2) = (x_1, x_2 + 1)^3 \)

4.3 Varying the Fraction of Partial Labels in PRIL

We now discuss the performance of PRIL when we vary the fraction of partial labels in the training. We train with 60%, 70%, 80%, 90% and 100% examples with partial labels. We compute the loss at each round with the same partial label used for updating the hypothesis.

For PRIL, at each time step, we compute the average of \( L_{\text{MAE}} \) (defined in eq.(1)). The results are shown in Figure 2. We see that for all the datasets the average mean absolute error (MAE) is decreases faster as compared to the number of rounds (T). Also, the average MAE decreases with the increase in the fraction of examples with partial labels. This we observe in all 3 datasets and both types of partial labels. This happens because the allowed range for predicted values is more for partial labels as compared to the exact label. Consider an example \( x^* \) with partial labels as \([y^*_1, y^*_2] \) (\( y^*_2 > y^*_1 \)) and exact label as \( y^* \). The MAE for this example with partial labels is sum of \( K - y^*_2 + y^*_2 - 1 \) losses. On the other hand, MAE becomes sum of \( K - 1 \) losses if the label is exact for \( x^* \). Thus, as we increase the fraction of partial labels in the training set, the average MAE decreases. We see that training with no partial labels gets larger values of average MAE for all the datasets.

4.4 Comparisons With Other Approaches

We compare the proposed algorithm with (a) PRank [3] by considering the exact labels, (b) Widrow-Hoff [13] by posing it as a regression problem and (c) multi-class Perceptron [4]. For PRIL, at each time step, we compute the average of \( L_{\text{MAE}} \) (defined in eq.(1)). For PRank, Widrow-Hoff and MC-Perceptron, we find the average absolute error (\( \frac{1}{T} \sum_{t=1}^{T} |y_t - y^*_t| \)). We repeat the process 100 times and average the instantaneous losses across the 100 runs.

We train PRIL with 100% partially labeled examples (Type 1 or Type 2). But when we compute the MAE, we use exact labels of examples. Which means, we used the model trained using PRIL with all partial labels and measure its performance with respect to the exact labels. This sets a harder evaluation criteria for PRIL. Moreover, in practice, we want a single predicted label and we want to see how similar it is as compared to the original label. Figure 3 shows the comparison plot of PRIL with PRank, Widrow-Hoff (WH) and Multi-Class Perceptron (MCP).

We see that PRIL does better as compared to Widrow-Hoff. This happens because Widrow-Hoff does not consider the categorical nature of the labels even though it respects the orderings of the labels. On the other hand, MCP is a complex model for solving a ranking problem. It also does not consider the ordering of the labels.
Figure 2: Experiment: Varying the fraction of examples with partial labels. Variation in average MAE performance by varying the fraction of partial labels in the training set. The loss is computed using the partial labels.

We see that for Synthetic and Parkinsons datasets PRIL performs better than MCP.

We observe that PRIL performance is comparable as compared to PRank \cite{3}. On Abalone and Parkinsons datasets, it performs similar to PRank. Which means that PRIL is able to recover the underlying classifier even if we have partial labels. This is a very interesting finding as we don’t need to worry to provide exact labels. All we need is a range around the exact labels. Thus, PRIL appears to be a better way to deal with ordinal classification when we have uncertainties in the labels.

5 CONCLUSIONS

We proposed a new online algorithm called PRIL for learning ordinal classifiers when we have interval labels for the examples. We show the correctness of the proposed algorithm. We show that PRIL converges after making finite number of mistakes whenever there exist an ideal classifier for the given interval labels. We extend the mistake bound result for the general setting when an ideal classifier may not exist. We show that regret for PRIL is $O(\sqrt{T})$. We experimentally show that PRIL is a very effective algorithm when we have interval labels.

A PROOF OF LEMMA 3.1

Note that $\theta^t_i \in \mathbb{Z}, \forall i \in \{1, \ldots, K-1\}, \forall t \in \{1, \ldots, N\}$ as PRIL initializes $\theta^1_i = 0, \forall i \in \{1, \ldots, K-1\}$. To show that PRIL preserves the ordering of the thresholds, we consider following different cases.
Let $\theta^T = [w^T \theta^T]$, and $v^t = [(w^T \theta^t)]$. Let Algorithm 1 makes a mistake at trial $t$. Let $M^0 = \{i \mid z_i^t(w^T \theta^t) \leq 0\}$. $M^t$ be the set of indices of the constraints which are not satisfied at trial $t$. Let $m^t = |M^t|$ be the number of those constraints. Thus,

$$v^{t+1} = v^t + \sum_{i \in M^t} z_i^t (w^T \theta^t) \theta^t_i$$

where we used the fact that $z_i^t(w^T \theta^t) \geq y$. Summing both sides from $t = 1$ to $T$ and using $v^1 = 0$, we get:

$$v^{T+1} = v^1 + y \sum_{t=1}^{T} m^t = y \sum_{t=1}^{T} m^t$$

This completes the proof.
Now, we will upper bound $||v^{T+1}||_2^2$:

$$||v^{T+1}||_2^2 - ||v^T||_2^2 = 2 \sum_{i=1}^{K-1} \tilde{r}_i \tilde{v}.x^t - \sum_{i=1}^{K-1} |\tilde{z}_i^T|_2^2$$

$$= -2 \sum_{i=1}^{K-1} \tilde{r}_i \tilde{v} \cdot \tilde{v} + \sum_{i=1}^{K-1} \tilde{r}_i \tilde{v}^2$$

$$\leq ||v^T||_2^2 + 2 \sum_{i=1}^{K-1} \tilde{r}_i (\tilde{x}^t - \tilde{v}_t) + (m^t)^2 R^2 + m^t$$

We used the fact that $\sum_{i \in \mathcal{M}(\tilde{r}_i)^2 = m^t}$; $\langle \sum_{i \in \mathcal{M}(\tilde{r}_i)^2 \leq (m^t)^2$. Using $z_2^i(w.x^t - \theta_i) \leq 0 \ \forall \ \mathcal{M}$, we get $||v^{T+1}||_2^2 - ||v^T||_2^2 \leq (m^t)^2 R^2 + m^t$. Summing both sides from $t = 1$ to $T$ and using $v^T = 0$, we get:

$$||v^{T+1}||_2^2 \leq R^2 \sum_{i=1}^{T} (m^t)^2 + T m^t$$

Now, using Cauchy-Schwartz inequality, we get

$$\sqrt{T} v^t \leq \sqrt{||v^{T+1}||_2^2} \leq \sqrt{||v^{T+1}||_2^2}$$

Now using eq. (5) and (6), we get

$$\frac{T \sum_{i=1}^{T} (m^t)^2}{R^2 \sum_{i=1}^{T} (m^t)^2 + T m^t} \leq \frac{1}{\sqrt{v^t}}$$

But, $m^t \leq K - \sum_{i \in \mathcal{I}} \theta_i^t - 1 \leq K - c - 1$, then $\sum_{i=1}^{T} (m^t)^2 \leq \sum_{i=1}^{T} m^t (K - c - 1)$. Which gives

$$\frac{T \sum_{i=1}^{T} m^t}{R^2 \sum_{i=1}^{T} (m^t)^2 + T m^t} \leq \frac{R^2 (K - c - 1) + 1}{R^2} \leq \frac{R^2 (K - c - 1)}{R^2}$$

But $m^t = L_{IAE}^T(f(x^t), \theta^t, y^t, y^t)$ Which means,

$$\sum_{i=1}^{T} L_{IAE}^T(f(x^t), \theta^t, y^t, y^t) \leq \frac{R^2 (K - c - 1)}{R^2}$$

This completes the proof.

**D. PROOF OF THEOREM 3.4**

Let $v = (w, \theta) \in \Omega$. Using the convexity property of $L_{IAE}^T$, we get

$$L_{IAE}^T (w, x^t, \theta^t, y^t, y^t) - L_{IAE}^T (w, x^t, \theta, y^t, y^t) \leq \frac{||v - v^{t+1}||^2}{\eta} \text{ using Algorithm 1}$$

where $v = (w, \theta) \in \Omega$. Using the convexity property of $L_{IAE}^T$, we get

$$\frac{1}{\eta} ||v - v^{t+1}||^2 \leq \frac{||v - v^{t+1}||^2}{\eta} \leq \frac{||v - v^{t+1}||^2}{\eta}$$

But,

$$||v - v^{t+1}||^2 = \sum_{i \in \mathcal{I}} (z_i^t)^2 \leq \sum_{i \in \mathcal{I}} (z_i^t)^2 \leq \frac{R^2 + 1}{R^2}$$

Thus, $\frac{1}{\eta} ||v - v^{t+1}||^2 \leq \frac{1}{\eta} ||v - v^{t+1}||^2$.
Summing eq.(7) on both sides from 1 to $T$, we get,
\[
\sum_{t=1}^{T} L_{1}^{LMC} (v^t, x^t, \theta^t, y^t, y^t') - \sum_{t=1}^{T} L_{2}^{LMC} (v^t, x^t, \theta^t, y^t, y^t') \\
\leq \frac{1}{2\eta} [||v - v^T||^2 - ||v - v^{T-1}||^2 + T \eta^2 (R_2^2 + 1)(K - c - 1)] \\
\leq \frac{1}{2\eta} [||v||^2 + T \eta^2 (R_2^2 + 1)(K - c - 1)]
\]
where we have used the fact that $||v - v^{T+1}||^2 \geq 0$. We know that $\min_{v \in \Omega} ||v||^2 = \Lambda^2$. Since the above holds for any $v = (w, \Theta) \in \Omega$, we have
\[
R_T(\text{PRIL}) \leq \frac{1}{2\eta} \left( \Lambda^2 + T \eta^2 (R_2^2 + 1)(K - c - 1) \right)
\]
We use $\eta = \frac{\Delta}{\sqrt{T(R_2^2 + 1)(K - c - 1)}}$ as it minimizes the regret bound above.

Thus, we get
\[
R_T(\text{PRIL}) \leq \Lambda \sqrt{T(R_2^2 + 1)(K - c - 1)}
\]
This completes the proof.

REFERENCES


