LETTER Communicated by Edward Harrington

Online Ranking by Projecting

Koby Crammer
kobics@cs.huji.ac.il
Yoram Singer
singer@cs.huji.ac.il
School of Computer Science and Engineering, Hebrew University, Jerusalem 91904, Israel

We discuss the problem of ranking instances. In our framework, each instance is associated with a rank or a rating, which is an integer in 1 to k. Our goal is to find a rank-prediction rule that assigns each instance a rank that is as close as possible to the instance's true rank. We discuss a group of closely related online algorithms, analyze their performance in the mistake-bound model, and prove their correctness. We describe two sets of experiments, with synthetic data and with the EachMovie data set for collaborative filtering. In the experiments we performed, our algorithms outperform online algorithms for regression and classification applied to ranking.

1 Introduction

The ranking problem we discuss in this article shares common properties with both classification and regression problems. As in classification problems, the goal is to assign one of k possible labels to a new instance. Similar to regression problems, the set of k labels is structured, as there is a total order relation between the labels. We refer to the labels as ranks and without loss of generality assume that the ranks constitute the set $\{1, 2, \ldots, k\}$. Settings in which it is natural to rank or rate instances rather than classify are common in tasks such as information retrieval and collaborative filtering. We use the latter as our running example. In collaborative filtering, the goal is to predict a user's rating on new items such as books or movies given the user's past ratings of the similar items. The goal is to determine whether a movie fan will like a new movie and to what degree, which is expressed as a rank. An example for possible ratings might be, "`run-to-see`, `very-good`, `good`, `only-if-you-must`, and `do-not-bother`." While the different ratings carry meaningful semantics, from a learning-theoretic point of view, we model the ratings as a totally ordered set (whose size is five in the example above).

The interest in ordering or ranking of objects is by no means new and is still the source of ongoing research in many fields, such as mathematical economics, social science, and computer science. For an overview of rank-

ing problems from a learning-theoretic point of view see Cohen, Schapire, and Singer (1999). One of the main results underscored in this article is a complexity gap between classification learning and ranking learning. To sidestep the inherent intractability problems of ranking learning, several approaches have been suggested. One possible approach is to cast a ranking problem as a regression problem. Another is to reduce a total order into a set of preferences over pairs (Freund, Iyer, Schapire, & Singer, 2003; Herbrich, Graepel, & Obermayer, 2000). The first case imposes a metric on the set of ranking rules that might not be realistic, while the second approach is time-consuming since it requires increasing the sample size from n to $O(n^2)$.

In this letter, we consider an alternative approach that directly maintains a totally ordered set via projections. Our starting point is similar to that of Herbrich et al. (2000) in the sense that we project each instance into the real numbers. However, our work then deviates and operates directly on rankings by associating each ranking with distinct subinterval of the real numbers and adapting the support of each subinterval while learning.

The article is organized as follows. In the next section, we describe a simple and efficient online algorithm that manipulates concurrently the direction onto which we project the instances and the division into subintervals. In section 3, we prove the correctness of the algorithm and analyze its performance in the mistake-bound model on various assumptions. Section 4 contains the description and analysis of a norm-optimized version of the basic ranking algorithm. We then shift our attention to a multiplicative algorithm that is described and analyzed in section 5. We provide empirical validation of the merits of the various ranking algorithms we devise in section 6. This section describes experiments comparing the ranking algorithms to online classification and regression algorithms. Finally, we conclude with a brief discussion and mention a few open problems.

Before moving on to the core of the article, we point to a few closely related articles. First, a preliminary version of this letter appeared at Neural Information Processing Systems, 2001, under the title "Pranking with Ranking" (Crammer & Singer, 2001a). This article extends the conference version in numerous directions by providing complete analysis as well as three new algorithms that were not discussed in the conference version. Second, in a recent work by Shashua and Levin (2002), an SVM-based algorithm for instance ranking was described. The algorithms of Shashua and Levin share numerous properties with the algorithms presented in this article. However, it is designed for batch settings, while our focus is on online algorithms. Last, we would like to note that Harrington (2003) demonstrated empirically an improved generalization performance of our algorithm using an "averaging" technique, while preserving the order of the thresholds.

2 The Basic PRank Algorithm

This letter focuses on online algorithms for ranking instances. We are given a sequence $(\bar{x}^1, y^1), \ldots, (\bar{x}^t, y^t), \ldots$ of instance-rank pairs. For concreteness,

we assume that each instance $\bar{x}^t$ is in $\mathbb{R}^n$, and its corresponding rank y^t is an element from finite set $\mathcal{Y}$ with a total order relation. We assume without loss of generality that $\mathcal{Y} = \{1, 2, \ldots, k\}$ with $>$ as the order relation. The total order over the set $\mathcal{Y}$ induces a partial order over the instances in the following natural sense. We say that $\bar{x}^t$ is preferred over $\bar{x}^s$ if $y^t > y^s$. We also say that $\bar{x}^t$ and $\bar{x}^s$ are not comparable if neither $y^t > y^s$ nor $y^t < y^s$. We denote this case simply as $y^t = y^s$. Note that the induced partial order is of a unique form in that the instances form k equivalence classes that are totally ordered.[1] We stress that although the elements of $\mathcal{Y}$ are denoted by integers, we do not use the fact that the integers also belong to a metric space. We assume only that the set $\mathcal{Y}$ is discrete and its elements are totally ordered. A ranking rule H is a mapping from the instance space to the set of rank values, $H\colon \mathbb{R}^n \to \mathcal{Y}$. The family of ranking rules we discuss in this article employs a vector $\bar{w} \in \mathbb{R}^n$ and a set of k thresholds $b_1 \leq \cdots \leq b_{k-1} \leq b_k = \infty$. For convenience, we denote by $\bar{b} = (b_1, \ldots, b_{k-1})$ the vector of thresholds, excluding b_k, which is fixed to ∞. Given a new instance $\bar{x}$, the ranking rule first computes the inner product between $\bar{w}$ and $\bar{x}$. The predicted rank is then defined to be the index of the first (smallest) threshold b_r for which $\bar{w} \cdot \bar{x} < b_r$. Such a ranking rule divides the space into parallel, equally ranked regions: all the instances that satisfy $b_{r-1} < \bar{w} \cdot \bar{x} < b_r$ are assigned the same rank r. Formally, given a ranking rule defined by $\bar{w}$ and $\bar{b}$, the predicted rank of an instance $\bar{x}$ is $H(\bar{x}) = \min_{r \in \{1,\ldots,k\}} \{r : \bar{w} \cdot \bar{x} - b_r < 0\}$. Note that the above minimum is always well defined since we set $b_k = \infty$.

The analysis that we use in this letter is based on the mistake-bound model for online learning. The algorithms we describe work in rounds. On round t, the learning algorithm gets an instance $\bar{x}^t$. Given $\bar{x}^t$, the algorithm outputs a rank, $\hat{y}^t = \min_r \{r : \bar{w} \cdot \bar{x}^t - b_r < 0\}$. It then receives the correct rank y^t and updates its ranking rule by modifying $\bar{w}$ and $\bar{b}$. We say that our algorithm made a ranking mistake if $\hat{y}^t \neq y^t$ and wish to make the predicted rank as close (in a sense described later in this article) as possible to the true rank. Formally, the goal of the learning algorithm is to minimize the ranking loss, which is defined to be the number of thresholds between the true rank and the predicted rank. Using the representation of ranks as integers in $\{1, \ldots, k\}$, the ranking loss after T rounds is equal to the cumulative difference between the predicted rank values and true rank values, $\sum_{t=1}^{T} |\hat{y}^t - y^t|$. The algorithm we describe updates its ranking rule only on rounds on which the predicted rank value was incorrect. Such algorithms are called *conservative*.

We now describe the update rule of the algorithm, which is motivated by the perceptron algorithm for classification; hence, we call it the PRank algorithm, shorthand for perceptron ranking. For simplicity, we omit the index of the round when referring to an input instance rank pair $(\bar{x}, y)$ and the ranking rule $\bar{w}$ and $\bar{b}$. Since $b_1 \leq b_2 \leq \cdots \leq b_{k-1} \leq b_k$, the predicted

[1] For a discussion of this type of partial orders see Kemeny and Snell (1962).

rank is correct if $\bar{w} \cdot \bar{x} > b_r$ for $r = 1, \ldots, y-1$ and $\bar{w} \cdot \bar{x} < b_r$ for $r = y, \ldots, k-1$. We represent the above inequalities by expanding the rank y into $k-1$ virtual variables $y_1, \ldots, y_{k-1}$. We set $y_r = +1$ for the case $\bar{w} \cdot \bar{x} > b_r$ and $y_r = -1$ for $\bar{w} \cdot \bar{x} < b_r$. Put another way, a rank value y induces the vector $(y_1, \ldots, y_{k-1}) = (+1, \ldots, +1, -1, \ldots, -1)$ where the maximal index r for which $y_r = +1$ is $y-1$. Thus, the prediction of a ranking rule is correct if $y_r(\bar{w} \cdot \bar{x} - b_r) > 0$ for all r. If the algorithm makes a mistake by ranking $\bar{x}$ as $\hat{y}$ instead of y, then there is at least one threshold, indexed r, for which the value of $\bar{w} \cdot \bar{x}$ is on the wrong side of b_r, that is, $y_r(\bar{w} \cdot \bar{x} - b_r) \leq 0$. To correct the mistake, we need to "move" the values of $\bar{w} \cdot \bar{x}$ and b_r toward each other. We do so by modifying only the values of the b_r's for which $y_r(\bar{w} \cdot \bar{x} - b_r) \leq 0$ and replace them with $b_r - y_r$. We also replace the value of $\bar{w}$ with $\bar{w} + (\sum y_r)\bar{x}$ where the sum is taken over the indices r for which there was a prediction error, that is, $y_r(\bar{w} \cdot \bar{x} - b_r) \leq 0$.

An illustration of the update rule is given in Figure 1. In the example, we used the set $\mathcal{Y} = \{1, \ldots, 5\}$. (Note that $b_5 = \infty$ is omitted from all the plots in Figure 1.) The correct rank of the instance is $y = 4$, and thus the value of $\bar{w} \cdot \bar{x}$ should fall in the fourth interval, between b_3 and b_4. However, in the illustration, the value of $\bar{w} \cdot \bar{x}$ fell below b_1 and the predicted rank is $\hat{y} = 1$. The threshold values $b_1,\ b_2$, and b_3 are a source of the error since the value of $b_1,\ b_2,\ b_3$ is higher than $\bar{w} \cdot \bar{x}$. To compensate for the mistake, the algorithm decreases $b_1,\ b_2$, and b_3 by a unit value and replaces them with $b_1 - 1,\ b_2 - 1$, and $b_3 - 1$. It also modifies $\bar{w}$ to be $\bar{w} + 3\bar{x}$ since

$$\sum_{r : y_r(\bar{w} \cdot \bar{x} - b_r) \leq 0} y_r = 3.$$

Thus, the inner product $\bar{w} \cdot \bar{x}$ increases by $3\|\bar{x}\|^2$. This update is illustrated at the middle plot of Figure 1. The prediction rule after the update is illustrated on the right-hand side of Figure 1. Note that after the update, the predicted rank of $\bar{x}$ is $\hat{y} = 3$, which is closer to the true rank $y = 4$. The pseudocode of algorithm is given in Figure 2.

To conclude this section, we note that PRank can be straightforwardly combined with Mercer kernels (Vapnik, 1998) and voting techniques (Freund & Schapire, 1999) often used for improving the performance of margin classifiers in batch and online settings (Cristianini & Shawe-Taylor, 2000). To do so, we assume access to an inner-product space $\mathcal{X}$ equipped with a kernel operator K: $\mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We now need to replace any inner-product operation the algorithm performs with an implicit inner-product operation defined via the kernel operator. We thus keep inner-product operators rather than explicit vectors. For instance, the update of PRank becomes

$$K(\bar{w}^{t+1}, \cdot) \leftarrow K(\bar{w}^t, \cdot) + \left(\sum_r \tau_r^t \right) K(\bar{x}^t, \cdot),$$

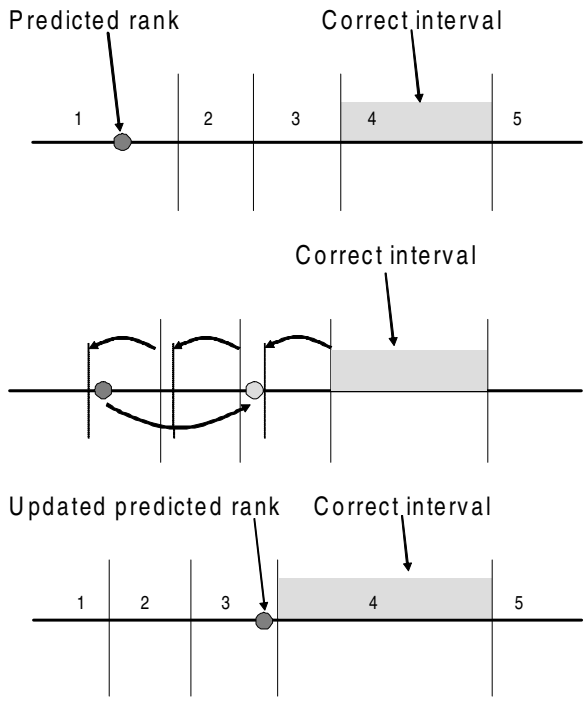

Figure 1: An Illustration of the update rule. The ranking rule predicts a rank of $\hat{y} = 1$ instead of $y = 4$ (top). The update decreases the thresholds b_1, b_2, b_3 by one unit and replaces $\bar{w}$ with $\bar{w} + 3\bar{x}$ (center), yielding that the predicted rank of $\bar{x}$ after the update is $\hat{y} = 3$ (bottom).

and thus the inner product $\bar{w}^T \cdot \bar{x}^t$ becomes

$$\sum_{s=1}^{t-1} \sum_{r} \tau_r^t K(\bar{x}^s, \bar{x}^t).$$

3 Analysis

Before we prove the mistake bound of the algorithm, we first need to show that it maintains a consistent hypothesis. That is, we need to show that PRank preserves the correct order of the thresholds; otherwise, it might be impossible to induce a rank prediction rule. We prove that the consistency of thresholds is maintained by showing inductively that for any ranking rule that can be derived by the algorithm along its run, $(\bar{w}^1, \bar{b}^1), \ldots, (\bar{w}^{T+1}, \bar{b}^{T+1})$ the set of inequalities $b_r^t \leq \cdots \leq b_{k-1}^t$ hold for all t. Clearly, since the initialization of the thresholds is such that $b_1^1 \leq b_2^1 \leq \cdots \leq b_{k-1}^1$, then it suffices

Initialize: Set $\bar{w}^1 = \bar{\mathbf{0}}$, $b_1^1, \ldots, b_{k-1}^1 = 0, b_k^1 = \infty$
Loop: For $t = 1, 2, \ldots, T$

- Receive a new instance $\bar{x}^t \in \mathbb{R}^n$
- Predict:

$$\hat{y}^t = \min_{r \in \{1,\ldots,k\}} \{r : \bar{w}^t \cdot \bar{x}^t - b_r^t < 0\}$$

- Receive a new rank-value y^t
- If $\hat{y}^t \neq y^t$ update $\bar{w}^t$ (otherwise set $\bar{w}^{t+1} = \bar{w}^t$, $\forall r$: $b_r^{t+1} = b_r^t$):

1. For $r = 1, \ldots, k-1$: If $y^t \leq r$ Then $y_r^t = -1$
   Else $y_r^t = 1$
2. For $r = 1, \ldots, k-1$: If $(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t \leq 0$ Then $\tau_r^t = y_r^t$
   Else $\tau_r^t = 0$
3. Update:

$$\bar{w}^{t+1} \leftarrow \bar{w}^t + \left(\sum_r \tau_r^t\right) \bar{x}^t$$
$$\text{For } r = 1, \ldots, k-1 : \quad b_r^{t+1} \leftarrow b_r^t - \tau_r^t$$

Output: $H(\bar{x}) = \min_{r \in \{1,\ldots,k\}} \{r : \bar{w}^{T+1} \cdot \bar{x} - b_r^{T+1} < 0\}$.

Figure 2: The PRank algorithm.

to show that the claim holds inductively. For simplicity of the proof below, let us write the update rule of PRank in an alternative form. Let $[[\pi]]$ be 1 if the predicate π holds and 0 otherwise. We now rewrite the value of τ_r^t (from Figure 2) as $\tau_r^t = y_r^t [[(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t \leq 0]]$. Note also that the values of b_r^t are integers for all r and t since for all r, we initialize $b_r^1 = 0$ and $b_r^{t+1} - b_r^t \in \{-1, 0, +1\}$.

Lemma 1 (order preservation). *Let $\bar{w}^t$ and $\bar{b}^t$ be the current ranking rule, where $b_1^t \leq \cdots \leq b_{k-1}^t$, and let $(\bar{x}^t, y^t)$ be an instance rank pair fed to PRank on round t. Denote by $\bar{w}^{t+1}$ and $\bar{b}^{t+1}$ the resulting ranking rule after the update of PRank. Then $b_1^{t+1} \leq \cdots \leq b_{k-1}^{t+1}$.*

Proof. In order to show that PRank preserves a nondecreasing order of the thresholds, we use the definition of the algorithm for y_r^t. We define $y_r^t = +1$ for $r < y^t$ and $y_r^t = -1$ for $r \geq y^t$. To prove that $b_{r+1}^{t+1} \geq b_r^{t+1}$, we rewrite the threshold update as

$$b_r^{t+1} = b_r^t - y_r^t [[(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t \leq 0]]. \tag{3.1}$$

In order to prove that $b_{r+1}^{t+1} \geq b_r^{t+1}$ for all feasible r, it is sufficient to show that

$$b_{r+1}^t - b_r^t \geq y_{r+1}^t[[(\bar{w}^t \cdot \bar{x}^t - b_{r+1}^t)y_{r+1}^t \leq 0]] - y_r^t[[(\bar{w}^t \cdot \bar{x}^t - b_r^t)y_r^t \leq 0]]. \quad (3.2)$$

Since by our inductive assumption $b_{r+1}^t \leq b_r^t$ and b_r^t , $b_{r+1}^t \in \mathbb{Z}$, we get that the value of $b_{r+1}^t - b_r^t$ on the left-hand side of equation 3.2 is a nonnegative integer. Recall also that $y_r^t = 1$ if $y^t > r$ and $y_r^t = -1$ otherwise, and therefore $y_{r+1}^t \leq y_r^t$. We now need to analyze two cases. We first consider the case where $y_{r+1}^t \neq y_r^t$, which implies that $y_{r+1}^t = -1$, $y_r^t = +1$. In this case, the right-hand side of equation 3.2 is at most zero, and the claim trivially holds. The second case is when $y_{r+1}^t = y_r^t$. Here we get that the value of the right-hand side of equation 3.2 cannot exceed 1. If $b_{r+1}^t > b_r^t$, then since both values are integers, the bound of 1 on the right-hand side of equation 3.2 implies that the value of b_r^t cannot exceed the value of b_{r+1}^t. We are thus left with the case where $b_r^t = b_{r+1}^t$ and $y_{r+1}^t = y_r^t$. For this case, we have that the terms $y_{r+1}^t[[(\bar{w}^t \cdot \bar{x}^t - b_{r+1}^t)y_{r+1}^t < 0]]$ and $y_r^t[[(\bar{w}^t \cdot \bar{x}^t - b_r^t)y_r^t < 0]]$ attain the same value. Therefore, the right-hand side of equation 3.2 is zero and $b_r^{t+1} = b_{r+1}^{t+1}$. This completes the proof.

We now turn to the mistake-bound analysis of the algorithm. In order to simplify the analysis of the algorithm, we introduce the following notation. Given a hyperplane $\bar{w}$ and a set of $k-1$ thresholds $\bar{b}$, we denote by $\bar{v} \in \mathbb{R}^{n+k-1}$ the vector that is a concatenation of $\bar{w}$ and $\bar{b}$, that is, $\bar{v} = (\bar{w}, \bar{b})$. For brevity, we refer to the vector $\bar{v}$ as a *ranking rule*. Given two vectors $\bar{v}' = (\bar{w}', \bar{b}')$ and $\bar{v} = (\bar{w}, \bar{b})$, we have $\bar{v}' \cdot \bar{v} = \bar{w}' \cdot \bar{w} + \bar{b}' \cdot \bar{b}$ and $\|\bar{v}\|^2 = \|\bar{w}\|^2 + \|\bar{b}\|^2$. Note that a vector $\bar{v}$ induces a partial order of the vectors $\mathbb{R}^n$.

Theorem 1 (mistake bound). *Let $(\bar{x}^1, y^1), \ldots, (\bar{x}^T, y^T)$ be an input sequence to PRank where $\bar{x}^t \in \mathbb{R}^n$ and $y^t \in \{1, \ldots, k\}$. Denote by $R^2 = \max_t \|\bar{x}^t\|^2$. If there exists a ranking rule $\bar{v}^* = (\bar{w}^*, \bar{b}^*)$ with $b_1^* \leq \cdots \leq b_{k-1}^*$ of a unit norm that classifies the entire sequence correctly with margin $\gamma = \min_{r,t}\{(\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t\} > 0$, the ranking loss of the algorithm $\sum_{t=1}^T |\hat{y}^t - y^t|$ is at most*

$$(k-1)\frac{R^2+1}{\gamma^2} .$$

Proof. Let us examine an example $(\bar{x}^t, y^t)$ that the algorithm received on round t. By definition, the algorithm ranked the example using the ranking rule $\bar{v}^t$, which is composed of $\bar{w}^t$ and the set of thresholds $\bar{b}^t$. Similarly, $\bar{v}^{t+1}$ is the ranking rule $(\bar{w}^{t+1}, \bar{b}^{t+1})$ after round t. Therefore, $\bar{w}^{t+1} = \bar{w}^t + \left(\sum_r \tau_r^t\right)\bar{x}^t$ and $b_r^{t+1} = b_r^t - \tau_r^t$ for $r = 1, 2, \ldots, k-1$. Let us denote by $n^t = |\hat{y}^t - y^t|$ the

difference between the true rank and the predicted rank. Since τ_r^t is zero for all the indices r such that $\text{sign}(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t > 0$, then it is straightforward to verify that $n^t = \sum_r |\tau_r^t|$. Note that if there was not a ranking mistake on round t, then $\tau_r^t = 0$ for $r = 1, \ldots, k-1$, and thus also $n^t = 0$. To prove the theorem, we bound $\sum_t n^t$ from above by bounding $\|\bar{v}^t\|^2$ from above and below. First, we derive a lower bound on $\|\bar{v}^t\|^2$ by bounding $\bar{v}^* \cdot \bar{v}^{t+1}$. Substituting the values of $\bar{w}^{t+1}$ and $\bar{b}^{t+1}$, we get that

$$\bar{v}^* \cdot \bar{v}^{t+1} = \bar{v}^* \cdot \bar{v}^t + \sum_{r=1}^{k-1} \tau_r^t (\bar{w}^* \cdot \bar{x}^t - b_r^*). \tag{3.3}$$

We further bound the term on the right-hand side by considering two cases corresponding to whether τ_r^t is positive or zero. Using the definition of τ_r^t from the pseudocode in Figure 2, we first examine the case where $(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t \leq 0$ and therefore $\tau_r^t = y_r^t$. From the assumption that $\bar{v}^*$ ranks the examples correctly with a margin of at least γ, we get that $\tau_r^t (\bar{w}^* \cdot \bar{x}^t - b_r^*) \geq \gamma$. The second case is when $(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t > 0$. In this case, we have $\tau_r^t = 0$ and thus $\tau_r^t (\bar{w}^* \cdot \bar{x}^t - b_r^*) = 0$. Combining the two cases and summing now over r, we get

$$\sum_{r=1}^{k-1} \tau_r^t (\bar{w}^* \cdot \bar{x}^t - b_r^*) \geq \sum_{r=1}^{k-1} |\tau_r^t| \gamma = n^t \gamma. \tag{3.4}$$

Combining equations 3.3 and 3.4, we get that $\bar{v}^* \cdot \bar{v}^{t+1} \geq \bar{v}^* \cdot \bar{v}^t + n^t \gamma$. Unfolding the sum, we get that after T rounds, the projection of the vector $\bar{v}^{T+1}$ on $\bar{v}^*$ satisfies

$$\bar{v}^* \cdot \bar{v}^{T+1} \geq \sum_t n^t \gamma = \gamma \sum_t n^t. \tag{3.5}$$

Recall that Cauchy-Schwartz inequality implies that

$$\|\bar{v}^{T+1}\|^2 \|\bar{v}^*\|^2 \geq (\bar{v}^{T+1} \cdot \bar{v}^*)^2.$$

Using equation 3.5 with Cauchy-Schwartz inequality and the assumption that $\bar{v}^*$ is of a unit norm, we get the following lower bound:

$$\|\bar{v}^{T+1}\|^2 \geq \left(\sum_t n^t \right)^2 \gamma^2. \tag{3.6}$$

We next bound the norm of $\bar{v}^{T+1}$ from above. As before, assume that an example $(\bar{x}^t, y^t)$ was ranked using the ranking rule $\bar{v}^t$, and denote by $\bar{v}^{t+1}$

the ranking rule at the end of round t. We now expand the values of $\bar{w}^{t+1}$ and $\bar{b}^{t+1}$ whose sum is the norm of $\bar{v}^{t+1}$ and get

$$\|\bar{v}^{t+1}\|^2 = \|\bar{w}^t\|^2 + \|\bar{b}^t\|^2 + 2\sum_r \tau_r^t \left(\bar{w}^t \cdot \bar{x}^t - b_r^t\right) + \left(\sum_r \tau_r^t\right)^2 \|\bar{x}^t\|^2 + \sum_r (\tau_r^t)^2.$$

Since $\tau_r^t \in \{-1, 0, +1\}$, we have that $(\sum_r \tau_r^t)^2 \leq (n^t)^2$ and $\sum_r (\tau_r^t)^2 = n^t$, and we therefore get

$$\|\bar{v}^{t+1}\|^2 \leq \|\bar{v}^t\|^2 + 2\sum_r \tau_r^t \left(\bar{w}^t \cdot \bar{x}^t - b_r^t\right) + (n^t)^2 \|\bar{x}^t\|^2 + n^t. \tag{3.7}$$

We further develop the second term using the update rule of the algorithm and get

$$\sum_r \tau_r^t \left(\bar{w}^t \cdot \bar{x}^t - b_r^t\right) = \sum_r [[(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t \leq 0]] \left((\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t\right) \leq 0. \tag{3.8}$$

Plugging equation 3.8 into equation 3.7 and using the bound $\|\bar{x}^t\|^2 \leq R^2$, we get that

$$\|\bar{v}^{t+1}\|^2 \leq \|\bar{v}^t\|^2 + (n^t)^2 R^2 + n^t.$$

Thus, the ranking rule we obtain after T rounds of the algorithm is upper bounded by

$$\|\bar{v}^{T+1}\|^2 \leq R^2 \sum_t (n^t)^2 + \sum_t n^t. \tag{3.9}$$

Combining the lower bound $\|\bar{v}^{T+1}\|^2 \geq (\sum_t n^t)^2 \gamma^2$ with the upper bound of equation 3.9, we have that

$$\left(\sum_t n^t\right)^2 \gamma^2 \leq \|\bar{v}^{T+1}\|^2 \leq R^2 \sum_t (n^t)^2 + \sum_t n^t.$$

Dividing the above equations by $\gamma^2 \sum_t n^t$, we finally get

$$\sum_t n^t \leq \frac{R^2 [\sum_t (n^t)^2] / [\sum_t n^t] + 1}{\gamma^2}. \tag{3.10}$$

Since by definition, n^t is at most $k - 1$, it implies that

$$\sum_t (n^t)^2 \leq \sum_t n^t (k-1) = (k-1) \sum_t n^t.$$

Plugging this inequality into equation 3.10, we get the desired bound,

$$\sum_{t=1}^{T} |\hat{y}^t - y^t| = \sum_{t=1}^{T} n^t \leq \frac{(k-1)R^2+1}{\gamma^2} \leq (k-1)\frac{R^2+1}{\gamma^2}.$$

We now turn our attention to the inseparable case in which there does not exist a positive margin value γ. To analyze the inseparable case, we use an analysis technique suggested by Freund and Schapire (1999). (This proof technique was first informally given by Vapnik, 1998.) To prove a mistake bound for the inseparable case, each example $(\bar{x}^t, y^t)$ is augmented with a slack variable, denoted d^t. Informally, for each time step t, the variable d^t designates how much the margin assumption is violated by the example. Formally, we obtain the following mistake bound for the inseparable case:

Theorem 2 (mistake bound for inseparable case). *Let $(\bar{x}^1, y^1), \ldots, (\bar{x}^T, y^T)$ be an input sequence for PRank where $\bar{x}^t \in \mathbb{R}^n$ and $y^t \in \{1, \ldots, k\}$. Denote by $R^2 = \max_t \|\bar{x}^t\|^2$. Let $\bar{v}^* = (\bar{w}^*, \bar{b}^*)$ be a ranking rule of a unit norm with $b_1^* \leq \cdots \leq b_{k-1}^*$. Let $\gamma > 0$ and define*

$$d^t = \max\{0, \gamma - \min_r\{(\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t\}\}. \tag{3.11}$$

Denote by $D^2 = \sum_t (d^t)^2$. Then the ranking loss of the algorithm is bounded above by

$$\sum_{t=1}^{T} |\hat{y}^t - y^t| \leq (k-1)\frac{(D + \sqrt{R^2+1})^2}{\gamma^2}.$$

The proof of the theorem is based on the proof technique for theorem 1 and is given in the appendix.

To conclude this section, we describe and briefly analyze a simplified version of PRank. This version shares the same algorithmic skeleton as PRank and differs only in the way it modifies its ranking rule. Rather than modifying all of the thresholds in the error set, this version chooses a single threshold to modify and update $\bar{w}$ accordingly. Since a single threshold is modified on each round, we use the abbreviation Si-PRank to refer to this version. More formally, we replace step 3 in Figure 2 with the following steps,

3. Define update index:
   If $\hat{y}^t > y^t$ Then $r = \hat{y}^t - 1$ Else $r = \hat{y}^t$
4. Update:

$$\begin{aligned} \bar{w}^{t+1} &\leftarrow \bar{w}^t + \tau_r^t \bar{x}^t \\ b_r^{t+1} &\leftarrow b_r^t - \tau_r^t \\ b_s^{t+1} &\leftarrow b_s^t \quad (s \neq r). \end{aligned}$$

It is to verify that this choice of a single threshold to update preserves the overall order of the threshold. The proof is immediate consequence of the lemma 1. In addition, following exactly the same proof technique of theorem 1, we get that the mistake bound of PRank also holds for Si-PRank.[2] Surprisingly, as we see later in section 6, in practice Si-PRank performs slightly better than PRank.

4 A Norm-Optimized Version of PRank

The PRank algorithm presented in the previous section performs the same form of update whenever a ranking error occurs. Furthermore, even when the projection of x^t onto $\bar{w}^t$ yields the correct rank value, the value of $\bar{w}^t \cdot x^t$ might lie very close to one of the thresholds b_r^t. The difference between the projection of x^t and the closest threshold plays a similar role to the notion of margin in classification problems and was used in theorem 1 to derive a mistake bound for PRank. In this section, we present a version of PRank that solves on each round a mini-optimization problem that balances between two opposing requirements. On one hand, we require that the new ranking rule $\bar{v}^{t+1}$ be as similar as possible to the previous ranking rule $\bar{v}^t$, which encompasses all our knowledge on past examples. On the other hand, we force the new ranking rule to rank the most recent example $\bar{x}^t$ correctly and with a large enough margin. Formally, assuming that there was a rank prediction error on round t, then we require that the new rule $(\bar{w}^{t+1}, \bar{b}^{t+1})$ would satisfy, $\min_r\{(\bar{w}^{t+1} \cdot \bar{x}^t - b_r^{t+1})y_r^t\} \geq \beta$, where β is a positive constant. These two requirements yield the following optimization problem:

$$\begin{aligned} \min_{\bar{w},\bar{b}} \quad & \frac{1}{2}\|(\bar{w},\bar{b}) - (\bar{w}^t,\bar{b}^t)\|^2 \\ \text{subject to:} \quad & (\bar{w} \cdot \bar{x}^t - b_r)y_r^t \geq \beta \quad \text{for } r = 1,\ldots,k-1. \end{aligned} \tag{4.1}$$

We call this version the Norm-Optimized PRank algorithm, or No-PRank in short. The pseudocode of the algorithm appears in Figure 3.

Before analyzing the algorithm, let us first further develop the optimization problem given in equation 4.1 by expanding its Lagrangian function,

$$\mathcal{L} = \frac{1}{2}\|(\bar{w},\bar{b}) - (\bar{w}^t,\bar{b}^t)\|^2 - \sum_{r=1}^{k-1} \tau_r[(\bar{w} \cdot \bar{x}^t - b_r)y_r^t - \beta]. \tag{4.2}$$

[2] In fact, an alternative bound can be given for Si-PRank, which states that the number of rounds on which an error occurs (indicated by a strictly positive value of the loss) is upper bounded by

$$\frac{R^2+1}{\gamma^2}.$$

Input: Minimal margin parameter β
Initialize: Set $\bar{w}^1 = \bar{0}$, $b_1^1, \ldots, b_{k-1}^1 = 0, b_k^1 = \infty$
Loop: For $t = 1, 2, \ldots, T$

- Receive a new instance $\bar{x}^t \in \mathbb{R}^n$.
- Predict:

$$\hat{y}^t = \min_{r \in \{1,\ldots,k\}} \{r : \bar{w}^t \cdot \bar{x}^t - b_r^t < 0\}$$

- Receive a new rank-value y^t
- If $y^t \neq \hat{y}^t$ update $\bar{w}^t$ and $\bar{b}^t$: (otherwise set $\bar{w}^{t+1} = \bar{w}^t$, $\bar{b}^{t+1} = \bar{b}^t$):
  1. For $r = 1, \ldots, k-1$: If $y^t \leq r$ Then $y_r^t = -1$
     Else $y_r^t = 1$.
  2. Update: set $(\bar{w}^{t+1}, \bar{b}^{t+1}) \in \mathbb{R}^{n+k-1}$ to be the minimizer of:

$$\min_{\bar{w},\bar{b}} \ \frac{1}{2}\|(\bar{w}, \bar{b}) - (\bar{w}^t, \bar{b}^t)\|^2$$

$$\text{subject to:} \ \ (\bar{w} \cdot \bar{x}^t - b_r) y_r^t \geq \beta \quad \text{for } r = 1, \ldots, k-1.$$

Output: $H(\bar{x}) = \min_{r \in \{1,\ldots,k\}} \{r : \bar{w}^{T+1} \cdot \bar{x} - b_r^{T+1} < 0\}$.

Figure 3: The Norm-Optimized PRank algorithm.

Here, τ_r are nonnegative Lagrange multipliers. Taking the derivative of equation 4.2 with respect to $\bar{w}$ and comparing it to zero, we get

$$\frac{\partial}{\partial \bar{w}} \mathcal{L} = \bar{w} - \bar{w}^t - \bar{x}^t \sum_r \tau_r y_r^t = 0 \qquad \Rightarrow \qquad \bar{w} = \bar{w}^t + \left(\sum_r \tau_r y_r^t \right) \bar{x}^t \ . \quad (4.3)$$

Repeating the process for b_r, we get

$$\frac{\partial}{\partial b_r} \mathcal{L} = b_r - b_r^t + \tau_r y_r^t = 0 \qquad \Rightarrow \qquad b_r = b_r^t - \tau_r y_r^t \ . \quad (4.4)$$

Plugging the value of $\bar{w}$ from equation 4.3 and b from equation 4.4 into equation 4.2, we get the following dual problem:

$$\begin{aligned} \min_{\tau} \mathcal{Q}_t(\bar{\tau}) = & \frac{1}{2}\|\bar{x}^t\|^2 \left(\sum_r \tau_r y_r^t \right)^2 + \frac{1}{2} \sum_r \tau_r^2 \\ & + \sum_r \tau_r \left[y_r^t \left(\bar{w}^t \cdot \bar{x}^t - b_r^t \right) - \beta \right] \\ & \text{s.t} \ \ 0 \leq \tau_r \ \ r = 1, \ldots, k-1. \end{aligned} \quad (4.5)$$

Before proceeding to discuss the formal properties of the No-PRank algorithm, it is worth noting that the new vector obtained at the end of round t, $\bar{w}^{t+1}$ is a linear combination of $\bar{w}^t$ and the current instance $\bar{x}^t$. Therefore, it is rather straightforward to use No-PRank in conjunction with kernel methods by maintaining $\bar{w}^t$ in its dual form as a weighted combination of the instances $\bar{x}^1, \ldots, \bar{x}^t$. Note also that the optimization problem of No-PRank reduces to a simple optimization problem with a quadratic objective function and nonnegativity constraints and can be solved using standard convex optimization tools (Fletcher, 1987).

Let us now discuss the formal properties of No-PRank. The following simple lemma states that No-PRank is a conservative online algorithm as it modifies its ranking rule if and only if the minimal margin requirements are not attained for the current instance.

Lemma 2 (conservativeness). *Let $(\bar{x}^t, y^t)$ denote the input to No-PRank on round t, and let y^t_r be -1 if $y^t \leq r$ and $+1$ otherwise. Then if $y^t_r \left(\bar{w}^t \cdot \bar{x}^t - b^t_r\right) \geq \beta$ for all $r = 1, \ldots, k-1$ the optimum of No-PRank's optimization problem is attained at $\tau_r = 0$ for all r.*

Proof. Consider the dual form of No-PRank's optimization problem given by equation 4.5. The objective function $\mathcal{Q}_t(\bar{\tau})$ is composed of three summands. The first two are clearly nonnegative and attain a value of zero iff all the multipliers τ_r are zero. If $y^t_r(\bar{w} \cdot \bar{x} - b^t_r) \geq \beta$, then the third summand is linear in $\bar{\tau}$ with nonnegative coefficients. Thus, its minimum is also attained when τ_r is zero for all r.

Next we show that No-PRank preserves the order of the thresholds along its run and thus can always serve as a valid ranking rule.

Lemma 3 (order preservation). *Let $\bar{w}^t$ and $\bar{b}^t$ be the current ranking rule, where $b^t_1 \leq \cdots \leq b^t_{k-1}$, and let $(\bar{x}^t, y^t)$ be an instance-rank pair fed to No-PRank on round t. Denote by $\bar{w}^{t+1}$ and $\bar{b}^{t+1}$ the resulting ranking rule after the update of No-PRank. Then $b^{t+1}_1 \leq \cdots \leq b^{t+1}_{k-1}$.*

Proof. In order to show that No-PRank maintains a correct (monotonically increasing) order of the thresholds, we use the variables employed by the algorithm along its run. Namely, we set $y^t_r = +1$ for $r < y^t$ and $y^t_r = -1$ for $r \geq y^t$. To prove that $b^{t+1}_{r+1} \geq b^{t+1}_r$ for all r, we expand $\bar{b}^{t+1}$ and show that

$$b^t_{r+1} - b^t_r \geq y^t_{r+1}\tau^t_{r+1} - y^t_r\tau^t_r. \tag{4.6}$$

We need to analyze two different settings. The first setting we analyze is when $y^t_{r+1} \neq y^t_r$, which implies that $y^t_{r+1} = -1$ and $y^t_r = +1$. In this case, the right-hand side of equation 4.6 is at most zero, while the left-hand side of

the equation is at least zero. Thus, the claim holds. The other case is when $y^t_{r+1} = y^t_r = y$. In this case, equation 4.6 becomes

$$b^t_{r+1} - b^t_r \geq y(\tau^t_{r+1} - \tau^t_r). \tag{4.7}$$

Assume by contradiction that the optimal value of equation 4.5 does not satisfy equation 4.7 for some r. We now construct another feasible set for $\bar{\tau}$ that yields a lower value of the objective function $\mathcal{Q}$. Let us define

$$\tau'_s = \begin{cases} \tau_s - y\epsilon & s = r+1 \\ \tau_s + y\epsilon & s = r \\ \tau_s & \text{otherwise,} \end{cases}$$

for some value of $\epsilon > 0$, which is determined below. Informally, the value of ϵ is set to be small enough so that the constraint $\tau'_s \geq 0$ still holds. (This is possible since $\tau_{r+1} > 0$.) Since $\bar{\tau}'$ and $\bar{\tau}$ differ only at their r and $r+1$ components, we get

$$\begin{aligned} \mathcal{Q}(\bar{\tau}') - \mathcal{Q}(\bar{\tau}) = &\frac{1}{2}({\tau'_r}^2 + {\tau'_{r+1}}^2) + \tau'_r[y_r(\bar{w}^t \cdot \bar{x}^t - b^t_r) - \beta] \\ &+ \tau'_{r+1}[y_{r+1}(\bar{w}^t \cdot \bar{x}^t - b^t_{r+1}) - \beta] \\ &- \frac{1}{2}({\tau_r}^2 + {\tau_{r+1}}^2) - \tau_r[y_r(\bar{w}^t \cdot \bar{x}^t - b^t_r) - \beta] \\ &- \tau_{r+1}[y_{r+1}(\bar{w}^t \cdot \bar{x}^t - b^t_{r+1}) - \beta]. \end{aligned}$$

Expanding $\bar{\tau}'$, we now get

$$\begin{aligned} \mathcal{Q}(\bar{\tau}') - \mathcal{Q}(\bar{\tau}) = &\frac{1}{2}((\tau_r + y\epsilon)^2 + (\tau_{r+1} - y\epsilon)^2) \\ &+ (\tau_r + y\epsilon)[y_r(\bar{w}^t \cdot \bar{x}^t - b^t_r) - \beta] \\ &+ (\tau_{r+1} - y\epsilon)[y_{r+1}(\bar{w}^t \cdot \bar{x}^t - b^t_{r+1}) - \beta] \\ &- \frac{1}{2}({\tau_r}^2 + {\tau_{r+1}}^2) - \tau_r[y_r(\bar{w}^t \cdot \bar{x}^t - b^t_r) - \beta] \\ &- \tau_{r+1}[y_{r+1}(\bar{w}^t \cdot \bar{x}^t - b^t_{r+1}) - \beta] \\ = &(y\epsilon)^2 + y\epsilon(\tau_r - \tau_{r+1}) \\ &+ y\epsilon[y_r(\bar{w}^t \cdot \bar{x}^t - b^t_r) - \beta] \\ &- y\epsilon[y_{r+1}(\bar{w}^t \cdot \bar{x}^t - b^t_{r+1}) - \beta]. \end{aligned}$$

Denoting both y_r and y_{r+1} simply as y and using the fact $y \in \{-1, +1\}$, we get

$$\begin{aligned} \mathcal{Q}(\bar{\tau}') - \mathcal{Q}(\bar{\tau}) &= (y\epsilon)^2 + y\epsilon(\tau_r - \tau_{r+1}) - \epsilon b_r + \epsilon b_{r+1} \\ &= \epsilon[\epsilon - (y(\tau_{r+1} - \tau_r) - (b_{r+1} - b_r))]. \end{aligned}$$

Since we assumed by contradiction that $y(\tau_{r+1} - \tau_r) - (b_{r+1} - b_r) = A > 0$, we can choose $0 < \epsilon \leq A/2$ and get

$$\mathcal{Q}(\bar{\tau}') - \mathcal{Q}(\bar{\tau}) = \epsilon(A/2 - A) = -\epsilon A/2 < 0,$$

which contradicts the assumption that τ is the optimal solution of equation 4.5.

We now turn to the analysis of the performance of No-PRank. We first bound the sum of the weight employed by the dual program as they accumulate along the run of No-PRank $\sum_t \sum_r \tau_r^t$. Based on the bound on the weights, we then prove a mistake bounds on the number of rounds on which an error occurred, that is, $y^t \neq \hat{y}^t$. As we see shortly, the bound depends on both the value of the attainable margin γ and the margin parameter, β, employed by the algorithm.

Theorem 3 (bound on weights). *Let $(\bar{x}^1, y^1), \ldots, (\bar{x}^T, y^T)$ be an input sequence to No-PRank where $\bar{x}^t \in \mathbb{R}^n$ and $y^t \in \{1, \ldots, k\}$. Let $\bar{v}^* = (\bar{w}^*, \bar{b}^*)$ be a ranking rule with $b_1^* \leq \cdots \leq b_{k-1}^*$ and $\|\bar{w}^*\|^2 + \sum_r (b_r^*)^2 = 1$. Assume that $\bar{v}^*$ classifies the entire sequence correctly with a margin value $\gamma = \min_{r,t}\{(\bar{w}^* \cdot \bar{x}^t - b_r^*) y_r^t\} > 0$. Then the total sum of the weights generated by No-PRank is bounded by*

$$\sum_{t=1}^{T} \sum_r \tau_r^t \leq 2\frac{\beta}{\gamma^2},$$

where β is a predefined parameter of the algorithm (see Figure 3).

Proof. Let us concentrate on an example $(\bar{x}^t, y^t)$ that the algorithm received on round t. By construction, the algorithm ranked the example using the ranking rule $\bar{v}^t$, which is composed of $\bar{w}^t$ and the thresholds $\bar{b}^t$. Similarly, we denote by $\bar{v}^{t+1}$ the updated rule $(\bar{w}^{t+1}, \bar{b}^{t+1})$ after round t, that is,

$$\bar{w}^{t+1} = \bar{w}^t + \left(\sum_r y_r^t \tau_r^t\right) \bar{x}^t \quad \text{and} \quad b_r^{t+1} = b_r^t - y_r^t \tau_r^t \text{ for } r = 1, 2, \ldots, k-1.$$

To bound $\sum_t \sum_r \tau_r^t$ from above, we derive bounds on $\|\bar{v}^t\|^2$ from both above and below. First, we derive a lower bound on $\|\bar{v}^t\|^2$ by bounding $\bar{v}^* \cdot \bar{v}^{t+1}$. Substituting the values of $\bar{w}^{t+1}$ and $\bar{b}^{t+1}$, we get

$$\bar{v}^* \cdot \bar{v}^{t+1} = \bar{v}^* \cdot \bar{v}^t + \sum_{r=1}^{k-1} \tau_r^t y_r^t (\bar{w}^* \cdot \bar{x}^t - b_r^*).$$

Using the assumption that $\bar{v}^*$ ranks the data correctly with a margin of at least γ, we get that $y_r^t(\bar{w}^* \cdot \bar{x}^t - b_r^*) \geq \gamma$:

$$\bar{v}^* \cdot \bar{v}^{t+1} \geq \bar{v}^* \cdot \bar{v}^t + \sum_{r=1}^{k-1} \tau_r^t \gamma = \bar{v}^* \cdot \bar{v}^t + \gamma \sum_{r=1}^{k-1} \tau_r^t.$$

Unfolding the sum, we get that after T rounds, the algorithm satisfies

$$\bar{v}^* \cdot \bar{v}^{T+1} \geq \gamma \sum_{t,r} \tau_r^t.$$

Plugging this result into Cauchy-Schwartz inequality,

$$\|\bar{v}^{T+1}\|^2 \|\bar{v}^*\|^2 \geq (\bar{v}^{T+1} \cdot \bar{v}^*)^2,$$

and using the assumption that $\bar{v}^*$ is of a unit norm, we get the lower bound,

$$\|\bar{v}^{T+1}\|^2 \geq \left(\sum_{t,r} \tau_r^t \right)^2 \gamma^2. \tag{4.8}$$

We next bound the norm of $\bar{v}^{T+1}$ from above. As before, assume that an example $(\bar{x}^t, y^t)$ was ranked using the ranking rule $\bar{v}^t$, and denote by $\bar{v}^{t+1}$ the ranking rule after the round. We now expand the values of $\bar{w}^{t+1}$ and $\bar{b}^{t+1}$ in the norm of $\bar{v}^{t+1}$ and get

$$\begin{aligned} \|\bar{v}^{t+1}\|^2 &= \|\bar{w}^t\|^2 + \|\bar{b}^t\|^2 + 2 \sum_r \tau_r^t \, y_r^t (\bar{w}^t \cdot \bar{x}^t - b_r^t) \\ &\quad + \|\bar{x}^t\|^2 \left(\sum_r \tau_r^t y_r^t \right)^2 + \sum_r (y_r^t \tau_r^t)^2. \end{aligned}$$

We add and subtract the term $2\beta \sum_r \tau_r^t$ on the right-hand side of the above equation and get

$$\begin{aligned} \|\bar{v}^{t+1}\|^2 &= \|\bar{w}^t\|^2 + \|\bar{b}^t\|^2 + 2 \sum_r \tau_r^t [y_r^t (\bar{w}^t \cdot \bar{x}^t - b_r^t) - \beta] \\ &\quad + \|\bar{x}^t\|^2 \left(\sum_r \tau_r^t y_r^t \right)^2 + \sum_r \left(y_r^t \tau_r^t \right)^2 + 2\beta \sum_r \tau_r^t \\ &= \|\bar{w}^t\|^2 + \|\bar{b}^t\|^2 + 2\mathcal{Q}(\bar{\tau}^t) + 2\beta \sum_r \tau_r^t, \end{aligned} \tag{4.9}$$

where we used the definition of $\mathcal{Q}$ from equation 4.5 to obtain the last equality.

Note that $\bar{\tau} = 0$ is a feasible solution for the optimization problem posed in equation 4.5. The value attained by $\mathcal{Q}$ for this particular choice of $\bar{\tau}$ is zero. Since $\bar{\tau}^t$ is the minimizer $\mathcal{Q}(\bar{\tau})$, we must have that

$$\mathcal{Q}(\bar{\tau}^t) \leq 0. \tag{4.10}$$

Substituting equation 4.10 in equation 4.9, we obtain the following bound on the norm of $\bar{v}^{t+1}$ in terms of the norm of $\bar{v}^t$:

$$\|\bar{v}^{t+1}\|^2 \leq \|\bar{v}^t\|^2 + 2\beta \sum_r \tau_r^t. \tag{4.11}$$

Thus, the ranking rule we obtain after T rounds of the algorithm satisfies the upper bound:

$$\|\bar{v}^{T+1}\|^2 \leq 2\beta \sum_{t,r} \tau_r^t. \tag{4.12}$$

Combining the lower bound of equation 4.8 with the upper bound of equation 4.12, we have that

$$\left(\sum_{t,r} \tau_r^t\right)^2 \gamma^2 \leq \|\bar{v}^{T+1}\|^2 \leq 2\beta \sum_{t,r} \tau_r^t.$$

Dividing both sides by $\gamma^2 \sum_{t,r} \tau_r^t$, we finally get

$$\sum_{t,r} \tau_r^t \leq \frac{2\beta}{\gamma^2}. \tag{4.13}$$

We now discuss some properties of the algorithm and its corresponding loss bound. As mentioned above, the algorithm updates its ranking rule only on rounds on which the margin is less than β, which reflects a minimal margin requirement. Informally, β can be viewed as a minimal difference requirement between the projection of the example $\bar{x}^t$ onto $\bar{w}^t$ and any of the thresholds b_r^t (see Figure 4). Thus, the larger β is, the better is the separation between the rank levels. Formally, the algorithm attempts to enclose all the examples in subintervals of the real numbers defined by the thresholds. As stated above, based on theorem 3, we can derive a mistake bound for No-PRank. Surprisingly, the dependency on the margin parameter β cancels out, and the end result is a mistake bound that depends on only the geometrical properties of the problem: the normalized margin.

Corollary 1 (mistake bound). *Assume the conditions of theorem 3 hold, and denote by* $R = \max_t \|\bar{x}^t\|$. *Then the number of rounds for which No-PRank made*

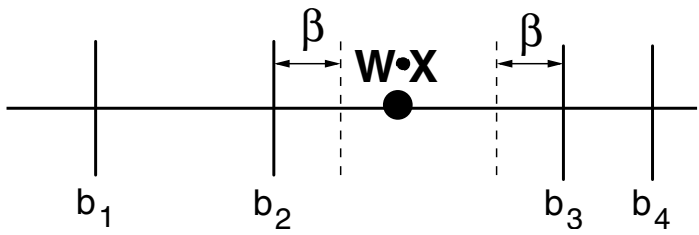

Figure 4: An illustration of the margin required by the No-PRank algorithm.

a mistake is upper bounded by

$$2\frac{R^2+1}{\gamma^2}.$$

Proof. To prove the corollary, we use theorem 3 and show that whenever an error occurred on round t, the sum of the dual weights $\sum_r \tau_r^t$ is at least $\beta/(R^2+1)$. Since $\bar{w}^{t+1}$ must satisfy the constraints of equation 4.1, we now show that for $r = 1, \ldots, k-1$,

$$(\bar{w}^{t+1} \cdot \bar{x}^t - b_r^{t+1}) y_r^t \geq \beta.$$

Substituting $\bar{w}^{t+1} = \bar{w}^t + \bar{x}^t \sum_r \tau_r^t y_r^t$ and b_r^{t+1} from equation 4.3 and $b_r^{t+1} = b_r^t - \tau_r y_r^t$ from equation 4.4, we get that the dual variables $\tau_1^t, \ldots, \tau_{k-1}^t$ satisfy

$$\beta \leq \left[\left(\bar{w}^t + \bar{x}^t \sum_s \tau_s^t y_s^t \right) \cdot \bar{x}^t - b_r^t + \tau_r y_r^t \right] y_r^t.$$

Rearranging terms in the above equation, we get

$$\begin{aligned} \beta &\leq \left[\left(\bar{w}^t + \bar{x}^t \sum_s \tau_s^t y_s^t \right) \cdot \bar{x}^t - b_r^t + \tau_r y_r^t \right] y_r^t \\ &= (\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t + y_r^t \|\bar{x}^t\|^2 \sum_s \tau_s^t y_s^t + \tau_r^t (y_r^t)^2. \end{aligned} \tag{4.14}$$

Since a rank prediction error occurred on round t, there exists r such that $(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t \leq 0$. In addition, since $y_r^t \in \{-1, +1\}$, we have that $y_r^t \sum_s \tau_s^t y_s^t \leq \sum_s \tau_s^t$. Using these two inequalities in equation 4.14 in conjunction with

the fact that $\tau_r^t \geq 0$, we get

$$\begin{aligned}\beta &\leq 0 + \|\bar{x}^t\|^2 \sum_r \tau_r^t + \sum_r \tau_r^t \\ &\leq (R^2+1) \sum_r \tau_r^t .\end{aligned}$$

We have thus shown that if there was a rank prediction error on round t, then

$$\frac{\beta}{R^2+1} \leq \sum_r \tau_r^t . \tag{4.15}$$

To prove the corollary, we combine theorem 3 with equation 4.15. Denote by M the number of rounds with prediction errors. We now show that on each such round, $\beta/(R^2+1) \leq \sum_r \tau_r^t$, and for the rest of the rounds, we simply bound the sum $\sum_r \tau_r^t$ from below by 0. We thus get that

$$M \frac{\beta}{R^2+1} \leq \sum_{t,r} \tau_r^t \, .$$

Applying theorem 3, we get

$$M \frac{\beta}{R^2+1} \leq 2\frac{\beta}{\gamma^2} \quad \Rightarrow \quad M \leq 2\frac{R^2+1}{\gamma^2}.$$

Note that the mistake bounds of theorem 1 and corollary 1 are identical up to a multiplicative $(k-1)/2$ factor. Furthermore, if we employ the trivial fact that $|y^t - \hat{y}^t| \leq k-1$, we can immediately obtain a bound on $\sum_t |y^t - \hat{y}^t|$ from corollary 1 that is a 2 factor of the bound given in theorem 1. However, the bound for No-PRank is more refined, as in addition to the mistake bound, we can also bound the total sum of the weights $\sum_{t,r} \tau_r^t$. Unfortunately, since τ_r^t may be arbitrarily close to zero, the more refined analysis does not yield a better mistake bound. One possible direction for improving the mistake bound itself may be obtained by modifying the minimal margin requirement from β to $\beta|y^t - \hat{y}^t|$. Another viable approach is to replace the fixed margin constraint $(\bar{w}^{t+1} \cdot \bar{x}^t - b_{t+1}^r) y_r^t \geq \beta$ with a margin requirement that is dependent on the difference between the correct label y^t and the specific threshold r. Specifically, we define a new set of constraints $(\bar{w}^{t+1} \cdot \bar{x}^t - b_{t+1}^r) y_r^t \geq \beta a_r$, where $a_r = y^t - r$ for $r < y^t$ and $a_r = r + 1 - y^t$ for $r \geq y^t$. Therefore, the further the threshold r from the interval containing the projection of the instance on the hyperplane, the larger the margin we require. We leave these possible extensions to future research.

5 A Multiplicative Version of PRank

In this section, we give a multiplicative version of PRank that we term Mu-PRank. This version is analogous to the basic PRank update, but it modifies $\bar{w}$ and $\bar{b}$ in a multiplicative manner. That is, on each round with rank prediction error, the current weight vector $\bar{w}^t$ and thresholds $\bar{b}^t$ are multiplied by factors that depend on $\bar{x}^t$. The motivation for this version is the multiplicative updates employed by the work of Warmuth and colleagues on online prediction (see, e.g., Kivinen & Warmuth, 1997). Mu-PRank maintains a normalized ranking rule, that is, $\|(\bar{w}^t, \bar{b}^t)\|_1 = 1$ for all t. As in PRank, on round t, Mu-PRank computes the vector $\bar{\tau}^t$, which determines the update of $\bar{w}^t$ and $\bar{b}^t$. It then takes the exponent of the updated ranking rule and normalizes it so that the ℓ_1 norm of $(\bar{w}^{t+1}, \bar{b}^{t+1})$ will be 1. The pseudocode of the algorithm is given in Figure 5.

Examining the logarithm of $\bar{b}^t$ on each round, it is rather simple to ver-

Input: Learning rate η
Initialize: Set $w_i^1 = \frac{1}{n+k-1} \quad i = 1, \ldots, n$, $b_1^1, \ldots, b_{k-1}^1 = \frac{1}{n+k-1}$, $b_k^1 = \infty$
Loop: For $t = 1, 2, \ldots, T$

- Receive a new instance $\bar{x}^t \in \mathbb{R}^n, \quad \|\bar{x}^t\|_\infty \leq 1$
- Predict:

$$\hat{y}^t = \min_{r \in \{1,\ldots,k\}} \{r : \bar{w}^t \cdot \bar{x}^t - b_r^t < 0\}$$

- Receive a new rank-value y^t
- If $y^t \neq \hat{y}^t$ update $\bar{w}^t$ and $\bar{b}^t$ (otherwise set $\bar{w}^{t+1} = \bar{w}^t$, $\bar{b}^{t+1} = \bar{b}^t$):

  1. For $r = 1, \ldots, k-1$: If $y^t \leq r$ Then $y_r^t = -1$ Else $y_r^t = 1$
  2. For $r = 1, \ldots, k-1$: If $(\bar{w}^t \cdot \bar{x}^t - b_r^t) y_r^t \leq 0$ Then $\tau_r^t = y_r^t$ Else $\tau_r^t = 0$
  3. Define:

$$\mathcal{Z}^t = \sum_{i=1}^{n} w_i^t e^{\eta x_i^t \sum_r \tau_r^t} + \sum_{r=1}^{k-1} b_r^t e^{-\eta \tau_r^t}$$

  4. Update:

$$\begin{aligned} &\text{For } i = 1, \ldots, n \quad : w_i^{t+1} \leftarrow w_i^t e^{\eta x_i^t \sum_r \tau_r^t} / \mathcal{Z}^t \\ &\text{For } r = 1, \ldots, k-1 : b_r^{t+1} \leftarrow b_r^t e^{-\eta \tau_r^t} / \mathcal{Z}^t \end{aligned}$$

Output: $H(\bar{x}) = \min_{r \in \{1,\ldots,k\}} \{r : \bar{w}^{T+1} \cdot \bar{x} - b_r^{T+1} < 0\}$

Figure 5: The multiplicative algorithm.

ify that, like PRank, Mu-PRank preserves the order of the thresholds. The additional step Mu-PRank employs in its update stage is the normalization of its ranking rule. However, since this normalization is applied to all the thresholds, it clearly keeps the order of the thresholds intact. More formally, $\log(b_r^t)$ can be written as $\eta u_r^t + \log(C_t)$, where u_r^t is an integer and C_t is independent of r. Applying exactly the same proof technique used in lemma 1 to u_r^t gives the following corollary:

Lemma 4 (order preservation). *Let $\bar{w}^t$ and $\bar{b}^t$ denote the current ranking rule and assume that $b_1^t \leq \cdots \leq b_{k-1}^t$. Then, after $(\bar{x}^t, y^t)$ is fed to Mu-PRank, the new rule $(\bar{w}^{t+1}, \bar{b}^{t+1})$ preserves the order of the thresholds, $b_1^{t+1} \leq \cdots \leq b_{k-1}^{t+1}$.*

Next, we analyze the mistake bound of Mu-PRank. Note that Mu-PRank employs a learning rate parameter, η. The mistake bound of Mu-PRank given in the following theorem depends on this value. If an priori lower bound on the margin γ is known, we can fix the value of η so as to minimize the loss bound of Mu-PRank. The resulting bound is described in corollary 2, which follows theorem 4.

Theorem 4 (mistake bound). *Let $(\bar{x}^1, y^1), \ldots, (\bar{x}^T, y^T)$ be an input sequence to Mu-PRank, where $\bar{x}^t \in \mathbb{R}^n$, $\|\bar{x}^t\|_\infty \leq 1$, and $y^t \in \{1, \ldots, k\}$. Assume that there exists a ranking rule $\bar{v}^* = (\bar{w}^*, \bar{b}^*)$ with $b_1^* \leq \cdots \leq b_{k-1}^*$ and $\|\bar{w}^*, \bar{b}^*\|_1 = 1$, which classifies the entire sequence correctly with margin $\gamma = \min_{r,t}\{(\bar{w}^* \cdot \bar{x}^t - b_r^*) y_r^t\} > 0$. Then the ranking loss of Mu-PRank, $\sum_{t=1}^T |\hat{y}^t - y^t|$ is, at most,*

$$\frac{\log\left(k + n - 1\right)}{\log\left(\frac{2}{e^{\eta(k-1)} + e^{-\eta(k-1)}}\right) + \eta\gamma}.$$

Proof. As before, let $\bar{v}^t = (\bar{w}^t, \bar{b}^t)$ and $\bar{v}^{t+1} = (\bar{w}^{t+1}, \bar{b}^{t+1})$ denote the ranking rules at the beginning and end of round t, respectively. To prove the theorem, we analyze the decrease in the Kullback-Leibler (KL) divergence (Cover & Thomas, 1991) between $\bar{v}^t$ and $\bar{v}^*$. The KL divergence of two discrete probability distributions $\vec{p}, \vec{q}$ is $\mathrm{D}_{KL}(\vec{p}\|\vec{q}) = \sum_i p_i \log(p_i/q_i)$. To prove the theorem, we examine the change of the KL divergence in two consecutive rounds:

$$\Delta_t = \mathrm{D}_{KL}(\bar{v}^*\|\bar{v}^{t+1}) - \mathrm{D}_{KL}(\bar{v}^*\|\bar{v}^t).$$

We bound $\sum_t \Delta_t$ from above and below. We first bound this sum from above.

As a reminder, we use n^t to denote $|\hat{y}^t - y^t|$. Using this notation, we get

$$\begin{aligned}
\Delta_t &= \sum_{i=1}^{n} w_i^* \log\left(\frac{w_i^*}{w_i^t}\right) + \sum_{r=1}^{k-1} b_r^* \log\left(\frac{b_r^*}{b_r^t}\right) \\
&= \sum_{i=1}^{n} w_i^* \log\left(\frac{\mathcal{Z}^t}{e^{\eta x_i^t \sum_r \tau_r^t}}\right) + \sum_{r=1}^{k-1} b_r^* \log\left(\frac{\mathcal{Z}^t}{e^{-\eta\tau_r^t}}\right) \\
&= \left[\sum_{i=1}^{n} w_i^* + \sum_{r=1}^{k-1} b_r^*\right] \log\left(\mathcal{Z}^t\right) - \eta \sum_{r=1}^{k-1} \tau_r^t (\bar{w}^* \cdot x^t - b_r^*) \\
&\leq \log(\mathcal{Z}^t) - \eta\gamma \sum_{r=1}^{k-1} |\tau_r^t| = \log(\mathcal{Z}^t) - \eta\gamma n^t, \qquad (5.1)
\end{aligned}$$

where we used the definition of n^t and τ_r^t in conjunction with the fact that $\bar{v}^*$ achieves a margin of γ to obtain the inequality above. We now bound $\log(\mathcal{Z}^t)$. We need the following inequality,

$$e^{\eta x} \leq \frac{a+x}{2a} e^{\eta a} + \frac{a-x}{2a} e^{-\eta a}, \qquad (5.2)$$

which holds for $a, \eta > 0$ and $x \in [-a, a]$. (The proof of this inequality is an immediate application of the convexity of the exponent function.) Now recall that

$$\mathcal{Z}^t = \sum_{i=1}^{n} w_i^t e^{\eta x_i^t \sum_r \tau_r^t} + \sum_{r=1}^{k-1} b_r^t e^{-\eta\tau_r^t}. \qquad (5.3)$$

We bound the left-hand side and the right-hand side of the above sum separately. For the left-hand side, we bound each term $e^{\eta x_i^t \sum_r \tau_r^t}$. Using equation 5.2 in conjunction with the fact that $|x_i^t| \leq 1$ and $|\sum_r \tau_r^t| \leq k-1$, we get

$$e^{\eta x_i^t \sum_r \tau_r^t} \leq \frac{k-1+x_i^t \sum_r \tau_r^t}{2(k-1)} e^{\eta(k-1)} + \frac{k-1-x_i^t \sum_r \tau_r^t}{2(k-1)} e^{-\eta(k-1)}. \qquad (5.4)$$

Similarly, $|\tau_r^t| \leq 1 \leq k-1$, and thus

$$e^{\eta\tau_r^t} \leq \frac{k-1+\tau_r^t}{2(k-1)} e^{\eta(k-1)} + \frac{k-1-\tau_r^t}{2(k-1)} e^{-\eta(k-1)}. \qquad (5.5)$$

Using the bounds from equations 5.4 and 5.5 in 5.3, we get

$$\mathcal{Z}^t \leq \sum_i w_i^t \left[\frac{k-1+x_i^t \sum_r \tau_r^t}{2(k-1)} e^{\eta(k-1)} + \frac{k-1-x_i^t \sum_r \tau_r^t}{2(k-1)} e^{-\eta(k-1)}\right]$$

$$
\begin{aligned}
&+ \sum_r b_r^t \left[\frac{k-1+\tau_r^t}{2(k-1)} e^{\eta(k-1)} + \frac{k-1-\tau_r^t}{2(k-1)} e^{-\eta(k-1)} \right] \\
&= \sum_i w_i^t \frac{1}{2}(e^{\eta(k-1)} + e^{-\eta(k-1)}) + \sum_r b_r^t \frac{1}{2}(e^{\eta(k-1)} + e^{-\eta(k-1)}) \\
&+ \sum_r \tau_r^t (\bar{w}^t \cdot x^t - b_r^t) \frac{e^{\eta(k-1)} + e^{-\eta(k-1)}}{2(k-1)}. \qquad (5.6)
\end{aligned}
$$

The definition of τ_r^t implies that $\tau_r^t(\bar{w}^t \cdot x^t - b_r^t) \leq 0$. (Either there was a ranking error or $\tau_r^t = 0$). In addition, since Mu-PRank normalizes its ranking rule at the end of each round, we know that $\|(\bar{w}^t, \bar{b}^t)\|_1 = \sum_i w_i^t + \sum_r b_r^t = 1$. Using these facts in equation 5.6, we get the following bound on $\mathcal{Z}^t$:

$$
\mathcal{Z}^t \leq \frac{1}{2}(e^{\eta(k-1)} + e^{-\eta(k-1)}). \qquad (5.7)
$$

Using equation 5.7 in equation 5.1, we obtain an upper bound on Δ_t:

$$
\Delta_t \leq \log \left[\frac{1}{2}(e^{\eta(k-1)} + e^{-\eta(k-1)}) \right] - \eta\gamma n^t.
$$

Since a ranking error occurred, we know that $n^t \geq 1$. In addition, the argument of the logarithm is at least 1; we can rearrange terms and write

$$
\Delta_t \leq n^t \left\{ \log \left[\frac{1}{2}(e^{\eta(k-1)} + e^{-\eta(k-1)}) \right] - \eta\gamma \right\}. \qquad (5.8)
$$

Summing over t, we get

$$
\sum_t \Delta_t \leq \left\{ \log \left[\frac{1}{2} \left(e^{\eta(k-1)} + e^{-\eta(k-1)} \right) \right] - \eta\gamma \right\} \sum_t n^t. \qquad (5.9)
$$

To bound $\sum \Delta_t$ from below, we unravel the sum and get

$$
\begin{aligned}
\sum_t \Delta_t &= \sum_t (\mathrm{D}_{KL}(\bar{v}^* \| \bar{v}^{t+1}) - \mathrm{D}_{KL}(\bar{v}^* \| \bar{v}^t)) \\
&= \mathrm{D}_{KL}(\bar{v}^* \| \bar{v}^{T+1}) - \mathrm{D}_{KL}(\bar{v}^* \| \bar{v}^1) \\
&\geq -\mathrm{D}_{KL}(\bar{v}^* \| \bar{v}^1),
\end{aligned}
$$

where the inequality is due to the fact that the KL divergence is always nonnegative. Using the value of $\bar{w}^1$, we get that $\mathrm{D}_{KL}(\bar{v}^* \| \bar{v}^1) = \log(k-1+n)$. Combining the lower bound on $\sum_t \Delta_t$ with equation 5.9, we get

$$
\log(k-1+n) \geq \left\{ \log \left(\frac{2}{e^{\eta(k-1)} + e^{-\eta(k-1)}} \right) + \eta\gamma \right\} \sum_t n^t.
$$

Rearranging terms, we finally get the desired bound:

$$\sum_t n^t \leq \frac{\log\left(k-1+n\right)}{\log\left(\frac{2}{e^{\eta(k-1)}+e^{-\eta(k-1)}}\right)+\eta\gamma}.$$

As discussed above, the bound of theorem 4 depends on the learning rate η. If γ or a lower bound on γ is known, we can set η to be

$$\eta = \frac{1}{2(k-1)} log\left(\frac{k-1+\gamma}{k-1-\gamma}\right).$$

For this choice of η, we get the following corollary:

Corollary 2. *If we run Mu-PRank with*

$$\eta = \frac{1}{2(k-1)} log\left(\frac{k-1+\gamma}{k-1-\gamma}\right),$$

then under the assumptions of theorem 4, the cumulative ranking loss obtained by Mu-PRank is bounded above by

$$\sum_t n^t \leq (k-1)^2 \frac{\log\left(n+k-1\right)}{\gamma^2}.$$

This corollary implies that the cumulative ranking loss of Mu-PRank is inversely proportional to square of the margin γ. Note that a direct comparison of this bound to the mistake bound of PRank (see theorem 1) is not possible as the notion of margin here is different. (For PRank, we used the ℓ_2 norm of the instances and the ranking rule to define the margin, while for Mu-PRank, we tacitly use the ℓ_1 norm of the ranking rule in conjunction with the ℓ_∞ of the instances.) This relation between the bounds also holds between additive and multiplicative online algorithms for classification (Rosenblatt, 1958; Littlestone, 1987, 1988, 1989). Putting these differences in the notion of margin aside, the two bounds exhibit the same quadratic dependency on the margin.

6 Experiments

In this section, we describe experiments we performed that compared the variants of PRank discussed in the previous section with two other online learning algorithms applied to ranking: a multiclass generalization of the Perceptron algorithm (Crammer & Singer, 2001b), denoted MCP, and the Widrow-Hoff algorithm (Widrow & Hoff, 1960) for online regression learning, which we denote by WH. For WH we fixed its learning rate to a

constant value. The hypotheses that the variants of PRank and the two other algorithms maintain share similarities but are different in their complexity: PRank maintains a vector $\bar{w}$ of dimension n and a vector of $k-1$ modifiable thresholds $\bar{b}$, totaling $n+k-1$ parameters; MCP maintains k prototypes, which are vectors of dimension n, yielding kn parameters; WH maintains a single vector $\bar{w}$ of size n. Therefore, MCP builds the most complex hypothesis, while WH builds the simplest.

We describe two sets of experiments with two different data sets. The data set used in the first experiment is synthetic and was generated in a similar way to the data set used by Herbrich et al. (2000). We first generated points $\bar{x} = (x_1, x_2)$ uniformly at random from the unit square $[0, 1]^2$. Each point was assigned a rank y from the set $\{1, \ldots, 5\}$ according to the following ranking rule, $y = \max_r \{r : 10((x_1 - 0.5)(x_2 - 0.5)) + \xi > b_r\}$ where $\bar{b} = (-\infty, -1, -0.1, 0.25, 1)$ and ξ is a normally distributed noise with a zero mean and a standard deviation of 0.125. We generated sequences of instance rank pairs, each of length 8000. We fed the sequences to PRank, Si-PRank, MCP, and WH. We then obtained predictions for each instance. We converted the real-valued predictions of WH into ranks by rounding each prediction to its closest rank value. As in Herbrich et al. (2000), we used a nonhomogeneous polynomial of degree 2, $K(\bar{x}_1, \bar{x}_2) = ((\bar{x}_1 \cdot \bar{x}_2) + 1)^2$ as the inner product operation between each input instance and the hyperplanes that the four additive algorithms maintain. At each time step, we computed for each algorithm the accumulative ranking loss normalized by the instantaneous sequence length. Formally, the time-averaged loss after T rounds is $(1/T) \sum_t^T |\hat{y}^t - y^t|$. We computed the loss for each algorithm we tested for $T = 1, \ldots, 8000$. To increase the statistical significance of the results, we repeated the process 100 times, picking a new random instance rank sequence of length 8000 each time, and then averaged the instantaneous losses across the 100 runs. The results are depicted on the left-hand side of Figure 6. The size of the symbols in the plot is larger than 95% confidence intervals for each result depicted in the figure. In this experiment, the performance of MCP is constantly worse than the performance of WH and PRank. WH initially suffers the smallest instantaneous loss, but after about 500 rounds, both PRank and Si-PRank start to outperform MCP and WH. Eventually the ranking loss that PRank and Si-PRank suffer is significantly lower than both WH and MCP.

In the second set of experiments, we used the EachMovie data set (McJones, 1997). This data set is used for collaborative filtering tasks and contains ratings of movies provided by 61,265 people. Each person in the data set viewed a subset of movies from a collection of 1623 titles. Each viewer rated each movie that she saw using one of six possible ratings: 0, 0.2, 0.4, 0.6, 0.8, 1. We chose subsets of people who viewed a significant number of movies, extracting for evaluation people who have rated at least 100 movies. There were 7542 such viewers. We chose at random one per-

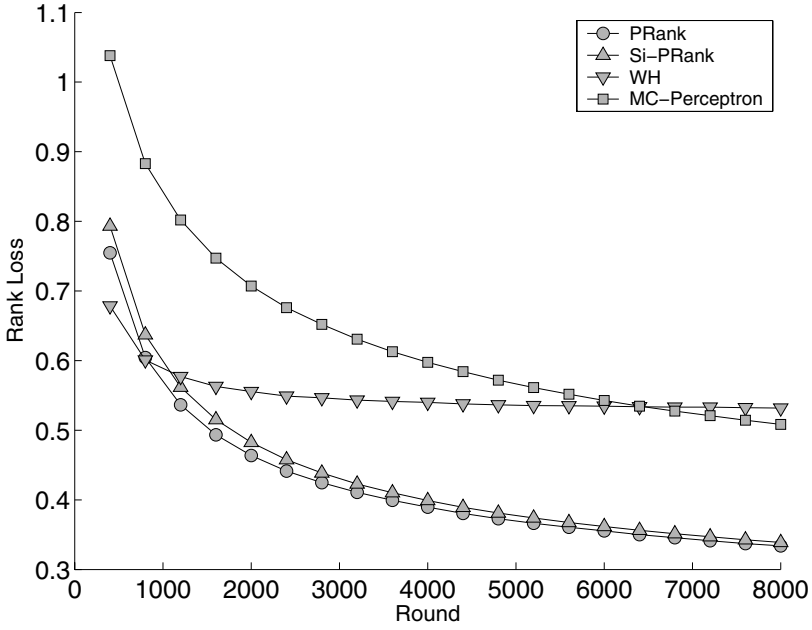

Figure 6: Comparison of the time-averaged ranking loss of PRank, Si-PRank, WH, and MCP on synthetic data.

son among these viewers and set the person's ratings to be the target rank. We used the ratings of the other viewers as features. Thus, the goal is to learn to predict the "taste" of a random user with the user's past ratings serving as a feedback and the ratings of fellow viewers as features. The prediction rule associates a weight with each fellow viewer and therefore can be seen as learning correlations between the tastes of different viewers. Next, we subtracted 0.5 from each rating; therefore the possible rating values are $-0.5, -0.3, -0.1, 0.1, 0.3, 0.5$. This linear transformation enabled us to assign a value of zero to movies that have not been rated. We fed each pair of feature and rank level in an online fashion. Since we chose viewers who rated at least 100 movies, we were able to perform at least 100 rounds of online predictions and updates. We repeated this experiment 500 times, each time choosing a random viewer as the target rank. The results are given in the left-hand plots of Figure 7. The error bars in the plot indicate 95% confidence levels. We repeated the experiment using viewers who have seen at least 200 movies. (There were 1802 such viewers.) The results of this experiment are given on the right-hand plots of Figure 7. The plots at the top of the figure compare the additive algorithms: PRank, Si-PRank, MCP, and WH. Along the entire run of the algorithms, PRank and Si-PRank are significantly

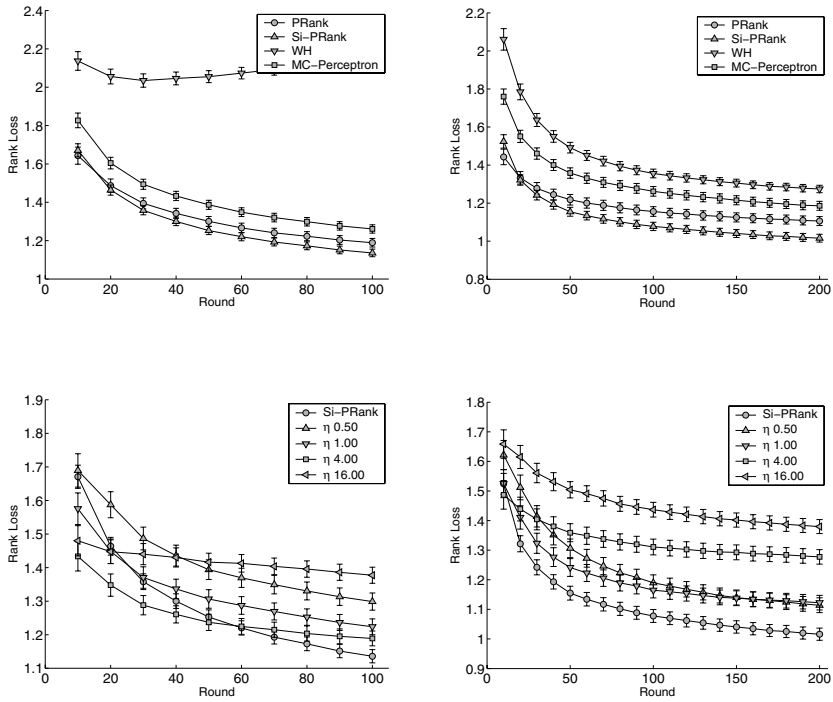

Figure 7: Comparison of the time-averaged ranking loss of the variants of PRank, WH, and MCP on the EachMovie data set using viewers who rated at least 200 movies (top) and at least 100 movies (bottom).

better than WH and consistently better than the multiclass perceptron algorithm, although the last employs a hypothesis that is substantially bigger. Comparing the two additive variants of PRank, we see that the ranking loss of Si-PRank is slightly lower than that of PRank. The plots at the bottom of the figure compare the best of the additive algorithms, Si-PRank, with the Mu-PRank run with different learning rates ($\eta = 0.5, 1, 4, 16$). One of the goals of this experiment is to check the dependency of the performance of Mu-PRank on its learning rate.

It is clear from the two bottom plots of Figure 7 that the performance of Mu-PRank is sensitive to the choice of the learning rate. Note, though, that the best learning rate is problem dependent: the best learning rate is $\eta = 4$ for the first partition of EachMovie, while $\eta = 0.5$ results in the smallest ranking loss in the case of the second partition. This type of behavior is also exhibited by multiplicative algorithms in other problems such as binary classification (Kivinen & Warmuth, 1997). Nonetheless, Mu-PRank outperforms most of the other algorithms we compared for a broad range

of learning rates. Focusing first on the left plot, we observe that after about 60 rounds, Si-PRank achieves the best performance, while at the start of the training process, its ranking loss is almost inferior to most of the copies of Mu-PRank. Summing up, both the additive and multiplicative versions of PRank outperform regression and classification algorithms when evaluated with respect to the ranking loss. While the superior performance of PRank is not surprising, it provides an empirical validation of the formal analysis presented in the previous sections.

7 Conclusion

In this article, we described a family of algorithms for instance ranking. The roots of the algorithms go back the perceptron algorithm (Rosenblatt, 1958). One of the major results in the article is the description of a new approach for solving ranking problems. While most of the previous approaches reduce the problem of ranking to a classification of pairs, we presented an alternative approach that builds a connection between rank levels and subintervals of the reals. An open problem that arises from both the theoretical analysis and the empirical results is deciding what version of PRank to use. In particular, PRank and Si-PRank share the same mistake bound, while in our experiments, Si-PRank performed better than PRank.

All of the versions of PRank presented in this article are online algorithms, and for their analysis we used the mistake-bound model. An interesting research direction is the design and analysis of algorithms for batch settings in which all of the training examples are given at once. As mentioned in section 1, Shashua and Levin (2002) have described a batch algorithm for ranking. An interesting question is how the different variants relate to the algorithm of Shashua and Levin. In addition to continuing our research on online algorithms for ranking problems, an interesting research direction is the design and generalization analysis of batch algorithms for various ranking losses.

Appendix: Technical Proofs

A.1 Proof of Theorem 2. To prove the theorem, we introduce a slight modification of theorem 1 by relaxing the assumption that $\bar{v}^*$ is of a unit norm and adding the norm of $\bar{v}^*$ to the mistake bound of the basic PRank algorithm. We thus write the mistake bound of theorem 1 as

$$\sum_{t=1}^{T} |\hat{y}^t - y^t| \leq (k-1)\frac{(R^2+1)\|\bar{v}^*\|^2}{\gamma^2}.$$

Since the case $D = 0$ reduces to setting of theorem 1, we can assume that $D > 0$. We prove the theorem by transforming the inseparable problem into a separable one. We do so by expanding each original instance $\bar{x}^t \in \mathbb{R}^n$ into

a vector $\bar{z}^t \in \mathbb{R}^{n+T}$ as follows. The first n coordinates of $\bar{z}^t$ are set to $\bar{x}^t$. The $n+t$ coordinate of $\bar{z}^t$ is set to Δ, which is a positive real number whose value is set below. The rest of the coordinates of $\bar{z}^t$ are set to zero. We similarly extend the ranking rule $(\bar{w}^*, \bar{b}^*)$ to $(\bar{u}^*, \bar{c}^*) \in \mathbb{R}^{(n+T)} \times \mathbb{R}^{k-1}$ as follows. We rewrite the value of d^t from equation 3.11 as

$$d^t = \max\left\{\gamma - \min_{r \geq y^t}\{(\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t\}, 0, \gamma - \min_{r < y^t}\{(\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t\}\right\}, \quad \text{(A.1)}$$

and define s^t to be the following indicator function:

$$s^t = \begin{cases} -1 & d^t = \gamma - \min_{r \geq y^t}\{(\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t\} \\ 0 & d^t = 0 \\ +1 & d^t = \gamma - \min_{r < y^t}\{(\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t\} \end{cases}. \quad \text{(A.2)}$$

Note that if $s^t = -1$, then $d^t = (\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t$ for some $r \geq y^t$, and thus $y_r^t = -1$. Similarly, if $s^t = +1$, then $d^t = (\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t$ for some $r < y^t$ and $y_r^t = +1$. We set the first n columns $\bar{u}^*$ to be $\bar{w}^*$, and the $n+t$ coordinate of $\bar{u}^*$ is set to be $\frac{s^t d^t}{\Delta}$ and the rest of the coordinates of $\bar{u}^*$ are set to zero.

We now show that $(\bar{u}^*, \bar{c}^*)$ achieves a margin value γ on the expanded sequence. Using the definition of s^t and d^t, we get

$$\begin{aligned}(\bar{u}^* \cdot \bar{z}^t - c_r^*)y_r^t &= (\bar{w}^* \cdot \bar{x}^t + \frac{s^t d^t}{\Delta}\Delta - b_r^*)y_r^t \\ &= (\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t + s^t d^t y_r^t \\ &= (\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t + d^t \\ &\geq (\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t + \gamma - (\bar{w}^* \cdot \bar{x}^t - b_r^*)y_r^t \\ &= \gamma. \end{aligned} \quad \text{(A.3)}$$

Note that by construction,

$$\|\bar{z}^t\|^2 \leq R^2 + \Delta^2.$$

Since the norm of $(\bar{w}^*, \bar{b}^*)$ is 1, we have

$$\|\bar{u}^*\|^2 + \|\bar{c}^*\|^2 = \|\bar{w}^*\|^2 + \|\bar{b}^*\|^2 + \sum_t \left(\frac{s^t d^t}{\Delta}\right)^2 = 1 + \frac{D^2}{\Delta^2}.$$

We now apply the bound of theorem 1 in the form discussed above and get that

$$\sum_{t=1}^{T} |\hat{y}^t - y^t| \leq (k-1)\frac{(R^2 + \Delta^2 + 1)(1 + \frac{D^2}{\Delta^2})}{\gamma^2}. \quad \text{(A.4)}$$

Setting $\Delta^2 = D\sqrt{R^2 + 1}$ in the equation above yields the desired bound.

It remains to show that the predictions of the algorithm on each element of the original sequence and on the expanded sequences are identical. This part follows exactly the same line of proof used in theorem 1 from Freund and Schapire (1998) and is thus omitted.

Acknowledgments

Thanks to Sanjoy Dasgupta and Rob Schapire for numerous discussions on ranking problems and algorithms. Thanks also to Eleazar Eskin and Uri Maoz for carefully reading the manuscript. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication reflects only the authors' views.

References

Cohen, W., Schapire, R. E., & Singer, Y. (1999). Learning to order things. *Journal of Artificial Intelligence Research*, *10*, 243–270.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Crammer, K., & Singer, Y. (2001a). Pranking with ranking. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems, 14*. Cambridge, MA: MIT Press.

Crammer, K., & Singer, Y. (2001b). Ultraconservative online algorithms for multiclass problems. In *Proceedings of the Fourteenth Annual Conference on Computational Learning Theory*. Amsterdam: Springer.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge: Cambridge University Press.

Fletcher, R. (1987). *Practical methods of optimization* (2nd ed.). New York: Wiley.

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research, 4*, 933–969.

Freund, Y., & Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning, 37*, 277–296.

Harrington, E. F. (2003). Online ranking/collaborative filtering using the perceptron algorithm. In *Proceedings of the Twentieth International Conference on Machine Learning*. Washington, DC: AIII Press.

Herbrich, R., Graepel, T., & Obermayer, K. (2000). Large marging rank boundaries for ordinal regression. In A. Smola, B. Schölkopf, & D. Schuurmans (Eds.), *Advances in large margin classifiers*. Cambridge, MA: MIT Press.

Kemeny, J. G., & Snell, J. L. (1962). *Mathematical models in the social sciences*. Cambridge, MA: MIT Press.

Kivinen, J., & Warmuth, M. K. (1997). Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, *132*(1), 1–64.

Littlestone, N. (1987). Learning when irrelevant attributes abound. In *28th Annual Symposium on Foundations of Computer Science* (pp. 68–77). Los Angeles: IEEE.

Littlestone, N. (1988). Learning when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, *2*, 285–318.

Littlestone, N. (1989). *Mistake bounds and logarithmic linear-threshold learning algorithms*. Unpublished doctoral dissertation, University of California, Santa Cruz.

McJones, P. (1997). *Eachmovie collaborative filtering data set*. DEC Systems Research Center. Available online at: http://www.research.digital.com/SRC/eachmovie/.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*, 386–407.

Shashua, A., & Levin, A. (2002). Ranking with large margin principle: Two approaches. In S. Becker, S. Thrün, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15*. Cambridge, MA: MIT Press.

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *1960 IRE WESCON Convention Record*, 96–104.

Received October 31, 2003; accepted June 1, 2004.

This article has been cited by:

2. Shivani Agarwal. 2010. Learning to rank on graphs. *Machine Learning* . [CrossRef]