Margin-Based Ranking Meets Boosting in the Middle*

Cynthia Rudin[1], Corinna Cortes[2], Mehryar Mohri[3], and Robert E. Schapire[4]

[1] Howard Hughes Medical Institute, New York University
4 Washington Place, Room 809, New York, NY 10003
rudin@nyu.edu
[2] Google Research, 1440 Broadway, New York, NY 10018
corinna@google.com
[3] Courant Institute, New York University, 719 Broadway, New York, NY 10003
mohri@cs.nyu.edu
[4] Princeton University, Department of Computer Science
35 Olden St., Princeton NJ 08544
schapire@cs.princeton.edu

Abstract. We present several results related to ranking. We give a general margin-based bound for ranking based on the L_∞ covering number of the hypothesis space. Our bound suggests that algorithms that maximize the ranking margin generalize well.
We then describe a new algorithm, Smooth Margin Ranking, that precisely converges to a maximum ranking-margin solution. The algorithm is a modification of RankBoost, analogous to Approximate Coordinate Ascent Boosting.
We also prove a remarkable property of AdaBoost: under very natural conditions, AdaBoost maximizes the exponentiated loss associated with the AUC and achieves the same AUC as RankBoost. This explains the empirical observations made by Cortes and Mohri, and Caruana and Niculescu-Mizil, about the excellent performance of AdaBoost as a ranking algorithm, as measured by the AUC.

1 Introduction

Consider the following supervised learning problem: Sylvia would like to get some recommendations for good movies before she goes to the theater. She would like a ranked list that agrees with her tastes as closely as possible, since she will see the movie closest to the top of the list that is playing at the local theater. For many pairs of movies she has seen, she will tell the learning algorithm whether she likes the first movie better than the second. This allows her to rank whichever pairs of movies she wishes, allowing for the possibility of ties between movies, and the possibility that certain movies cannot necessarily be compared by her, e.g, she may not compare action movies with cartoons. Another advantage of this type of scoring over real-valued scoring is that Sylvia does not need to normalize her

* This work is partially supported by NSF grant CCR-0325463.

own scores in order to compare with the rankings of another person; she just compares rankings on pairs of movies. Sylvia does not need to be consistent, since she may rank $a > b > c > a$. Each pair of movies such that Sylvia ranks the first above the second is called a "crucial pair".

The learning algorithm has access to a set of n individuals ("weak rankers", or "ranking functions") who also rank pairs of movies. The learning algorithm must combine the views of the weak rankers in order to match Sylvia's preferences, and generate a recommendation list that will generalize her views. This type of problem was studied in depth by Freund et al. [7], where the RankBoost algorithm was introduced.

In order to give some indication that an algorithm will generalize well (e.g., we want the ranking algorithm to predict movies that Sylvia will like), one often considers generalization bounds. Generalization bounds show that a small probability of error will most likely be achieved through a balance of the empirical error and the complexity of the hypothesis space. This complexity can by measured by an informative quantity, such as the VC dimension, covering number, Rademacher complexity, or a more specialized quantity, such as the bipartite rank shatter coefficient [1], which was used to derive a generalization bound specifically for the case of bipartite ranking. The "bipartite" ranking problem is a special case of the ranking problem where there are only two classes, a positive class, i.e., "good movies", and a negative class, i.e., "bad movies".

When deriving generalization bounds, it is illustrative to consider the "separable" case, where all training instances are correctly handled by the learning algorithm so the empirical error is zero. The separable case in ranking means that the algorithm's chosen ranking is consistent with all crucial pairs; the algorithm ranks the first instance in each crucial pair above the second. In the bipartite ranking problem, the separable case means that all positive instances are ranked above all negative instances, and the Area Under the ROC Curve (AUC) is exactly one.

In the separable case for classification, one important indicator of a classifier's generalization ability is the "margin", e.g., for boosting [15] and support vector machines. Although the empirical success of an algorithm depends on many factors (e.g., the type of data and how noisy it is), margin-based bounds often do provide a reasonable explanation (though not a complete understanding) of the success of many algorithms, both empirically and theoretically. Although there has been some work devoted to generalization bounds for ranking [7, 1], the bounds that we are aware of are not margin-based, and thus do not provide this useful discrimination between ranking algorithms in the separable case.

In Section 3, we provide a margin-based bound for ranking in a general setting. Our bound uses the L_∞ covering number as the complexity measure for the hypothesis space.

Since we are providing a general margin-based bound for ranking, we derive algorithms which create large margins. For the classification problem, it was proved that AdaBoost does not always maximize the margin [12]. In fact, AdaBoost does not even necessarily make progress towards increasing the margin at

every iteration. In analogy, RankBoost does not directly maximize the ranking margin, and it may not increase the margin at every iteration. In Section 4.1 we introduce a Smooth Margin Ranking algorithm, and prove that it makes progress towards increasing the "smooth" ranking margin at every iteration; this is the main step needed to prove convergence and convergence rates. This algorithm is analogous to Approximate Coordinate Ascent Boosting [14, 13] in its derivation, but the analogous proof that progress occurs at each iteration is much trickier; hence we present a sketch of this proof here.

In the bipartite ranking problem, we want our recommendation list to minimize the misranking error, e.g., the probability that a bad movie is ranked above a good movie. The empirical version of this misranking error is closely related to the AUC. RankBoost [7] minimizes an exponentiated version of this misranking error, in analogy with the classification algorithm AdaBoost, which minimizes an exponentiated version of the margins of training instances.

Although AdaBoost and RankBoost were derived analogously for the settings of classification and ranking, the parallels between these algorithms are deeper than their derivations. Cortes and Mohri [5] and Caruana and Niculescu-Mizil [3] have noted that AdaBoost experimentally seems to be very good at the bipartite ranking problem, even though it was RankBoost that was explicitly designed to solve this problem, not AdaBoost. That is, AdaBoost often achieves a large AUC. In Section 5, we show an important reason for these observations. Namely, if the weak learning algorithm is capable of producing the constant classifier, i.e., the classifier whose value is always one, then remarkably, AdaBoost and RankBoost will produce the same solution.

We proceed from the most general to the most specific. In Section 3 we provide a margin based bound for general ranking, which holds for each element of the hypothesis space. In Sections 4.1 and 4.2 we fix the form of hypothesis space to match that of RankBoost, i.e., the space of binary functions. Here, we discuss coordinate-based ranking algorithms such as RankBoost, and introduce the Smooth Margin Ranking algorithm. In Section 5, we focus on the bipartite ranking problem. Here, we discuss conditions for AdaBoost to act as a bipartite ranking algorithm by minimizing the exponentiated loss associated with the AUC. Sections 3 and 4.2 focus on the separable case, and Sections 4.1 and 5 focus on the non-separable case.

The main contributions of this paper are: 1) a margin-based ranking bound, 2) a theorem stating that our Smooth Margin Ranking algorithm makes progress at every iteration towards increasing the smooth ranking margin, and 3) conditions when AdaBoost acts as a bipartite ranking algorithm.

2 Notation

The training set, denoted by S, is $\{\mathbf{x}_i\}_{i=1,\ldots,m}$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^N$. The set $\mathcal{X}$ may be finite or infinite. In the case of the movie ranking problem, the $\mathbf{x}_i$'s are the movies and $\mathcal{X}$ is the database. The instances $\mathbf{x}_i \in \mathcal{X}$ are chosen independently and at random (iid) from a fixed but unknown probability distribution $\mathcal{D}$ on $\mathcal{X}$.

The notation $\mathbf{x} \sim \mathcal{D}$ means $\mathbf{x}$ is chosen randomly according to $\mathcal{D}$, and $S \sim \mathcal{D}^m$ means the m elements of the training set S are chosen iid according to $\mathcal{D}$.

The values of the "truth" function $\pi : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$, which is defined over pairs of instances, are analogous to the "labels" in classification. If $\pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1$, the pair $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$ is a crucial pair, i.e., $\bar{\mathbf{x}}$ should be ranked more highly than $\tilde{\mathbf{x}}$. We require only that $\pi(\bar{\mathbf{x}}, \bar{\mathbf{x}}) = 0$, meaning $\bar{\mathbf{x}}$ cannot be ranked higher than itself, and also $\pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1$ implies $\pi(\tilde{\mathbf{x}}, \bar{\mathbf{x}}) = 0$, meaning that if $\bar{\mathbf{x}}$ is ranked higher than $\tilde{\mathbf{x}}$, that $\tilde{\mathbf{x}}$ cannot be ranked higher than $\bar{\mathbf{x}}$. (It is possible that these assumptions may be dropped.) It is possible to have $\pi(a, b) = 1$, $\pi(b, c) = 1$, and $\pi(c, a) = 1$; this forces us to be in the non-separable case. The quantity $E := \mathbb{E}_{\bar{\mathbf{x}}, \tilde{\mathbf{x}} \sim \mathcal{D}} \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}})$ is the expected fraction of pairs in the database that are crucial pairs, $0 \leq E \leq 1/2$. We assume that π is a deterministic (non-noisy) function, and that for each pair of training instances $\mathbf{x}_i, \mathbf{x}_k$, we receive $\pi(\mathbf{x}_i, \mathbf{x}_k)$.

Our goal is to construct a ranking function $f : \mathcal{X} \to \mathbb{R}$, which gives a real valued score to each instance in $\mathcal{X}$. We do not care about the actual values of each instance, only the relative values; for crucial pair $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$, we do not care if $f(\bar{\mathbf{x}}) = .4$ and $f(\tilde{\mathbf{x}}) = .1$, only that $f(\bar{\mathbf{x}}) > f(\tilde{\mathbf{x}})$. Also, $f \in L_\infty(\mathcal{X})$ (or if $|\mathcal{X}|$ is finite, $f \in \ell_\infty(\mathcal{X})$).

In the usual setting of boosting for classification, $\forall \mathbf{x}$, $|f(\mathbf{x})| \leq 1$, and the *margin of training instance i* (with respect to classifier f) is $y_i f(\mathbf{x}_i)$, where y_i is the classification label, $y_i \in \{-1, 1\}$ [15]. The *margin of classifier f* is the minimum margin over all training instances, $\min_i y_i f(\mathbf{x}_i)$. Intuitively, the margin tells us how much f can change before one of the training instances is misclassified; it gives us a notion of how stable the classifier is.

For the ranking setting, we define an analogous notion of margin. Here, we can normalize f so that $|f| \leq 1$. The *margin of crucial pair i,k* with respect to ranking function f will be defined as $f(\mathbf{x}_i) - f(\mathbf{x}_k)$. The *margin of ranking function f*, is the minimum margin over all crucial pairs,

$$\mu_f := \min_{\{i,k | [\pi(\mathbf{x}_i, \mathbf{x}_k) = 1]\}} f(\mathbf{x}_i) - f(\mathbf{x}_k).$$

Intuitively, the margin tells us how much the ranking function can change before one of the crucial pairs is misranked. As with classification, we are in the separable case whenever the margin of f is positive.

3 A Margin-Based Bound for Ranking

In this section, we provide a bound which gives us an intuition for separable-case ranking and yields theoretical encouragement for margin-based ranking algorithms. The quantity we hope to minimize is analogous to the misclassification probability for classification; for two randomly chosen instances, if they are a crucial pair, we want to minimize the probability that these instances will be misranked. That is, we want to minimize:

$$\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} := \mathbb{P}_{\mathcal{D}}\{f(\bar{\mathbf{x}}) \leq f(\tilde{\mathbf{x}}) | [\pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1]\} = \frac{\mathbb{E}_{\bar{\mathbf{x}}, \tilde{\mathbf{x}} \sim \mathcal{D}}[\mathbf{1}_{[f(\bar{\mathbf{x}}) \leq f(\tilde{\mathbf{x}})]} \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}})]}{E}. \tag{1}$$

The numerator of (1) is the fraction of pairs that are both crucial and incorrectly ranked by f, and the denominator, $E := \mathbb{E}_{\tilde{\mathbf{x}},\tilde{\mathbf{x}}\sim\mathcal{D}}\pi(\tilde{\mathbf{x}},\tilde{\mathbf{x}})$ is the fraction of pairs that are crucial pairs. Thus, $\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\}$ is the proportion of crucial pairs that are incorrectly ranked by f.

Since we do not know $\mathcal{D}$, we may use only empirical quantities that rely on our training sample. An empirical quantity analogous to $\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\}$ is:

$$\begin{aligned}\mathbb{P}_S\{\text{misrank}_f\} := \mathbb{P}_S\{\text{margin}_f \le 0\} &:= \mathbb{P}_S\{f(\mathbf{x}_i) \le f(\mathbf{x}_k) | [\pi(\mathbf{x}_i,\mathbf{x}_k) = 1]\} \\ &:= \frac{\sum_{i=1}^m \sum_{k=1}^m [\mathbf{1}_{(f(\mathbf{x}_i)\le f(\mathbf{x}_k))}\pi(\mathbf{x}_i,\mathbf{x}_k)]}{\sum_{i=1}^m \sum_{k=1}^m \pi(\mathbf{x}_i,\mathbf{x}_k)}.\end{aligned}$$

We make this definition more general, by allowing it to include a margin of $\theta \ge 0$:

$$\begin{aligned}\mathbb{P}_S\{\text{margin}_f \le \theta\} &:= \mathbb{P}_S\{f(\mathbf{x}_i) - f(\mathbf{x}_k) \le \theta | [\pi(\mathbf{x}_i,\mathbf{x}_k) = 1]\} \\ &= \frac{\sum_{i=1}^m \sum_{k=1}^m [\mathbf{1}_{(f(\mathbf{x}_i)-f(\mathbf{x}_k)\le\theta)}\pi(\mathbf{x}_i,\mathbf{x}_k)]}{\sum_{i=1}^m \sum_{k=1}^m \pi(\mathbf{x}_i,\mathbf{x}_k)},\end{aligned}$$

i.e., $\mathbb{P}_S\{\text{margin}_f \le \theta\}$ is the fraction of crucial pairs in $S \times S$ with margin no larger than θ.

We want to bound $\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\}$ in terms of an empirical, margin-based term and a complexity term. The type of complexity we choose is the L_∞ covering number of the hypothesis space $\mathcal{F}$, $\mathcal{F} \subset L_\infty(\mathcal{X})$. (Upper bounds on the covering number can be calculated; see [6]). The covering number $\mathcal{N}(\mathcal{F},\sigma)$ is defined as the minimum number of balls of radius σ needed to cover $\mathcal{F}$, using the L_∞ metric. The following theorem is proved in Appendix A:

Theorem 1. *For $\epsilon > 0$, $\theta \ge 0$, for all $f \in \mathcal{F}$,*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left[\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \le \mathbb{P}_S\{\text{margin}_f \le \theta\} + \epsilon\right] \ge 1 - \mathcal{N}\left(\mathcal{F}, \frac{\epsilon\theta}{8}\right) 2\exp\left[\frac{-m(\epsilon E)^2}{8}\right].$$

That is, with probability depending on m, E, θ, ϵ, and $\mathcal{F}$, the misranking probability is less than the fraction of instances with margin below θ, plus ϵ.

We have chosen to write our bound in terms of E, but we could equally well have used an analogous empirical quantity, namely $\mathbb{E}_{\mathbf{x}_i,\mathbf{x}_k\sim S}\pi(\mathbf{x}_i,\mathbf{x}_k) = \frac{1}{m(m-1)}\sum_{i=1}^m \sum_{k=1}^m \pi(\mathbf{x}_i,\mathbf{x}_k)$. This is an arbitrary decision; we cannot maximize E in practice because the data is random. Either way, the bound tells us that the margin should be an important quantity to consider in the design of algorithms.

As a special case of the theorem, we consider the case of a finite hypothesis space $\mathcal{F}$, where the covering number achieves its largest value (for any θ), i.e., $\mathcal{N}\left(\mathcal{F}, \frac{\epsilon\theta}{4}\right) = |\mathcal{F}|$. Now we can solve for ϵ:

$$\delta := |\mathcal{F}| 2\exp\left[-\frac{m(\epsilon E)^2}{8}\right] \Longrightarrow \epsilon = \frac{1}{\sqrt{m}}\sqrt{\frac{8}{E^2}\left(\ln 2|\mathcal{F}| + \ln(1/\delta)\right)}.$$

This could be compared directly with Theorem 1 of Schapire et al [15]. Cucker and Smale [6] have reduced the factor ϵ^2 to a factor of ϵ in certain cases; it is possible that this bound may be tightened, especially in the case of a convex combination of weak rankers. An interesting open problem is to prove generalization

bounds using Rademacher complexity or a more specialized bound analogous to those of Koltchinskii and Panchenko [10]; here the trick would be to find an appropriate symmetrization step. In any case, our bound indicates that the margin is an important quantity for generalization.

4 Coordinate-Based Ranking Algorithms

In the previous section we presented a uniform bound that holds for all $f \in \mathcal{F}$. In the following we discuss how a learning algorithm might pick one of those functions, in order to make $\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\}$ as small as possible, based on intuition gained from the bound of Theorem 1; the bound reveals the margin to be a useful quantity in the learning process, so it deserves consideration in our design of algorithms.

In Section 4.1, we discuss RankBoost's objective function $\tilde{F}$. Then, we describe a coordinate descent algorithm on this objective. In Section 4.2 we define the smooth ranking margin $\tilde{G}$, present the Smooth Margin Ranking algorithm, and prove that it makes progress towards increasing $\tilde{G}$ at each iteration and converges to a maximum margin solution.

4.1 Coordinate Descent on RankBoost's Objective

We consider the hypothesis space $\mathcal{F}$ to be the class of convex combinations of "weak" rankers $\{h_j\}_{j=1,\ldots,n}$, where $h_j : X \to \{0,1\}$. We assume that if h_j is a weak ranker, that $1 - h_j$ is not chosen as a weak ranker; this assumption avoids the complicated seminorm notation in earlier work [13]. The function f is constructed as a normalized linear combination of the h_j's: $f = \sum_j \lambda_j h_j \Big/ ||\boldsymbol{\lambda}||_1$, where $||\boldsymbol{\lambda}||_1 = \sum_j \lambda_j$.

We construct a structure $\mathbf{M}$, which describes how each individual weak ranker j ranks each crucial pair i, k. We define $\mathbf{M}$ element-wise as: $M_{ikj} := h_j(\mathbf{x}_i) - h_j(\mathbf{x}_k)$. Thus, $M_{ikj} \in \{-1, 0, 1\}$. Since $\mathbf{M}$ has three indices, we need to define right multiplication: $(\mathbf{M}\boldsymbol{\lambda})_{ik} := \sum_{j=1}^{n} M_{ikj}\lambda_j = \sum_{j=1}^{n} \lambda_j h_j(\mathbf{x}_i) - \lambda_j h_j(\mathbf{x}_k)$ for $\boldsymbol{\lambda} \in \mathbb{R}^n$ and left multiplication: $(\mathbf{d}^T\mathbf{M})_j := \sum_{i,k|[\pi(\mathbf{x}_i,\mathbf{x}_k)=1]} d_{ik} M_{ikj}$ for $\mathbf{d} \in \mathbb{R}^{\#\text{crucial}}$, where "#crucial" is the number of crucial pairs.

Just as AdaBoost can be represented as a coordinate descent algorithm on a specific objective function of $\boldsymbol{\lambda}$ (see [9]), so can RankBoost. The objective function for RankBoost is:

$$\tilde{F}(\boldsymbol{\lambda}) := \sum_{\{i,k|[\pi(\mathbf{x}_i,\mathbf{x}_k)=1]\}} e^{-(\mathbf{M}\boldsymbol{\lambda})_{ik}}.$$

We perform standard coordinate descent on $\tilde{F}$ to derive "Coordinate Descent RankBoost". The direction chosen at iteration t (i.e., the choice of weak ranker j_t) in the "optimal" case (where the best weak ranker is chosen at each iteration) is given by: $j_t \in \operatorname{argmax}_j \left[-\frac{d\tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_j)}{d\alpha} \Big|_{\alpha=0} \right] = \operatorname{argmax}_j (\mathbf{d}_t^T \mathbf{M})_j$, where the

"weights" $d_{t,ik}$ are defined over pairs of instances by: $d_{t,ik} = 0$ for non-crucial pairs, and for crucial pair i, k: $d_{t,ik} := e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}}/\tilde{F}(\boldsymbol{\lambda}_t)$. One can see that the chosen weak ranker is a natural choice, namely, j_t is the most accurate weak ranker with respect to the weighted crucial training pairs.

Define $\mathcal{I}_+ := \{i, k | M_{ikj_t} = 1, \pi(\mathbf{x}_i, \mathbf{x}_k) = 1\}$ (although $\mathcal{I}_+$ is different for each t, we eliminate the subscript), also $\mathcal{I}_- := \{i, k | M_{ikj_t} = -1, \pi(\mathbf{x}_i, \mathbf{x}_k) = 1\}$. Also define $d_+ := \sum_{\mathcal{I}_+} d_{t,ik}$ and $d_- := \sum_{\mathcal{I}_-} d_{t,ik}$. The step size at iteration t is α_t, where α_t satisfies the equation for the line search along direction j_t:

$$0 = -\frac{d\tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t})}{d\alpha}\Big|_{\alpha=\alpha_t} \quad \Rightarrow \quad \alpha_t = \frac{1}{2} \ln \frac{d_+}{d_-}. \tag{2}$$

Thus, we have derived the first algorithm, Coordinate Descent RankBoost.

RankBoost, as it is described by Freund et al. [7], is similar, but differs by the ordering of steps: the formula for α_t is calculated first (via (2)), and afterwards j_t is determined using knowledge of the formula for α_t. For RankBoost, there may not be a natural interpretation of this type of weak learning algorithm as there is for Coordinate Descent RankBoost.

It is interesting that for AdaBoost's objective function, the plain coordinate descent algorithm and the variation (choosing the coordinate with knowledge of the step size) actually turn out to both yield the same algorithm, i.e., AdaBoost.

4.2 Smooth Margin Ranking

The value of $\tilde{F}$ does not directly tell us anything about the margin, only whether the margin is positive. Using any $\boldsymbol{\lambda}$ that yields a positive margin, we can actually make $\tilde{F}$ arbitrarily small by multiplying $\boldsymbol{\lambda}$ by a large positive constant, so the objective is arbitrarily small, yet the margin may not be maximized. Actually, the same problem occurs for AdaBoost. It has been proven [12] that for certain $\mathbf{M}$'s, AdaBoost does not converge to a maximum margin solution, nor does it even make progress towards increasing the margin at every iteration. Since the calculations are identical for RankBoost, there are certain cases in which we can similarly expect RankBoost not to converge to a maximum margin solution.

In earlier work, we proposed a smooth margin function which one can maximize in order to achieve a maximum margin solution for the classification problem [13]. We also proposed a coordinate ascent algorithm on this function which makes progress towards increasing the smooth margin at every iteration. Here, we present the analogous smooth ranking function and the Smooth Margin Ranking algorithm. The smooth ranking function $\tilde{G}$ is defined as follows:

$$\tilde{G}(\boldsymbol{\lambda}) := \frac{-\ln \tilde{F}(\boldsymbol{\lambda})}{||\boldsymbol{\lambda}||}.$$

With proofs identical to those of Rudin et al. [13], one can show that:

$$\tilde{G}(\boldsymbol{\lambda}) < \mu(\boldsymbol{\lambda}) \leq \rho, \quad \text{where} \tag{3}$$

$\rho = \min_{\{\mathbf{d} | \sum_{ik} d_{ik}=1, d_{ik} \geq 0\}} \max_j (\mathbf{d}^T \mathbf{M})_j = \max_{\{\bar{\boldsymbol{\lambda}} | \sum_j \bar{\lambda}_j = 1, \bar{\lambda}_j \geq 0\}} \min_i (\mathbf{M}\bar{\boldsymbol{\lambda}})_i$, i.e., the smooth ranking margin is less than the true margin, and the true margin is no greater than ρ, the min-max value of the game defined by $\mathbf{M}$ (see [8]).

We define the Smooth Margin Ranking algorithm, which is approximately coordinate ascent on $\tilde{G}$. As usual, the input to the algorithm is matrix $\mathbf{M}$. We will only define this algorithm when $\tilde{G}(\boldsymbol{\lambda})$ is positive, so that we only use this algorithm once the data has become separable; we can use RankBoost or Coordinate Descent RankBoost to get us to this point.

We will define iteration $t+1$ in terms of the quantities known at iteration t, namely: the current value of the objective $g_t := \tilde{G}(\boldsymbol{\lambda}_t)$, the weights $d_{t,ik} := e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}}/\tilde{F}(\boldsymbol{\lambda}_t)$, the direction $j_t = \operatorname{argmax}_j (\mathbf{d}_t^T\mathbf{M})_j$, and the edge $r_t := (\mathbf{d}_t^T\mathbf{M})_{j_t}$. The choice of j_t is the same as for Coordinate Descent RankBoost, also see [13]. The step size α_t is chosen to obey (6) below, but we need more definitions before we state its value. We define recursive equations for $\tilde{F}$ and $\tilde{G}$, and then use these to build up to (6). We also have $s_t = ||\boldsymbol{\lambda}_t||_1$ and $s_{t+1} = s_t + \alpha_t$, and $g_{t+1} = \tilde{G}(\boldsymbol{\lambda}_t + \alpha_t \mathbf{e}_{j_t})$.

As before, $\mathcal{I}_+ := \{i,k | M_{ikj_t} = 1, \pi(\mathbf{x}_i,\mathbf{x}_k) = 1\}$, $\mathcal{I}_- := \{i,k | M_{ikj_t} = -1, \pi(\mathbf{x}_i,\mathbf{x}_k) = 1\}$, and now, $\mathcal{I}_0 := \{i,k | M_{ikj_t} = 0, \pi(\mathbf{x}_i,\mathbf{x}_k) = 1\}$. Also $d_+ := \sum_{\mathcal{I}_+} d_{t,ik}$, $d_- := \sum_{\mathcal{I}_-} d_{t,ik}$, and $d_0 := \sum_{\mathcal{I}_0} d_{t,ik}$. So, by definition, $d_+ + d_- + d_0 = 1$. Now, r_t becomes $r_t = d_+ - d_-$. Define the factor τ_t and its "derivative" τ_t':

$$\tau_t := d_+ e^{-\alpha_t} + d_- e^{\alpha_t} + d_0, \text{ and } \tau_t' := -d_+ e^{-\alpha_t} + d_- e^{\alpha_t}.$$

We derive a recursive equation for $\tilde{F}$, true for any α:

$$\tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t}) = \sum_{\{i,k|\pi(\mathbf{x}_i,\mathbf{x}_k)=1\}} e^{(-\mathbf{M}\boldsymbol{\lambda}_t)_{ik}} e^{-M_{ikj_t}\alpha} = \tilde{F}(\boldsymbol{\lambda}_t)(d_+ e^{-\alpha} + d_- e^{\alpha} + d_0).$$

Thus, we have defined τ_t so that $\tilde{F}(\boldsymbol{\lambda}_{t+1}) = \tilde{F}(\boldsymbol{\lambda}_t + \alpha_t \mathbf{e}_{j_t}) = \tilde{F}(\boldsymbol{\lambda}_t)\tau_t$. We use this to write a recursive equation for $\tilde{G}$:

$$\tilde{G}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t}) = \frac{-\ln(\tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t}))}{s_t + \alpha} = g_t \frac{s_t}{s_t + \alpha} - \frac{\ln(d_+ e^{-\alpha} + d_- e^{\alpha} + d_0)}{s_t + \alpha}.$$

For our algorithm, we set $\alpha = \alpha_t$ in the above expression:

$$g_{t+1} = g_t \frac{s_t}{s_t + \alpha_t} - \frac{\ln \tau_t}{s_t + \alpha_t} \;\Rightarrow\; g_{t+1} - g_t = -\frac{1}{s_{t+1}} \left[g_t \alpha_t + \ln \tau_t \right]. \tag{4}$$

With this notation we write the equation for α_t for Smooth Margin Ranking. For plain coordinate ascent, the update α^* solves:

$$0 = \frac{d\tilde{G}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t})}{d\alpha}\Big|_{\alpha=\alpha^*} = \frac{1}{s_t + \alpha^*} \left[-\tilde{G}(\boldsymbol{\lambda}_t + \alpha^* \mathbf{e}_{j_t}) + \left[\frac{-d\tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t})/d\alpha \Big|_{\alpha=\alpha^*}}{\tilde{F}(\boldsymbol{\lambda}_t)} \right] \right].$$

We could solve this equation numerically for α^* to get a smooth margin coordinate ascent algorithm, however, we avoid this line search. To get the update rule for Smooth Margin Ranking, we set α_t to solve:

$$0 = \frac{1}{s_t + \alpha_t} \left[-\tilde{G}(\boldsymbol{\lambda}_t) + \left[\frac{-d\tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t})/d\alpha \Big|_{\alpha=\alpha_t}}{\tilde{F}(\boldsymbol{\lambda}_t)} \right] \right] = \frac{1}{s_t + \alpha_t} \left(-g_t + \frac{-\tau_t' \tilde{F}(\boldsymbol{\lambda}_t)}{\tau_t \tilde{F}(\boldsymbol{\lambda}_t)} \right)$$

$$g_t \tau_t = -\tau_t'. \tag{5}$$

This expression can be solved analytically for α_t, which makes the algorithm as easy to implement as RankBoost:

$$\alpha_t = \ln\left[\frac{-g_t d_0 + \sqrt{g_t^2 d_0^2 + (1+g_t)(1-g_t)(1+r_t-d_0)(1-r_t-d_0)}}{(1+g_t)(1-r_t-d_0)}\right]. \quad (6)$$

The following theorem states that the algorithm makes significant progress towards increasing the value of $\tilde{G}$ at every iteration. An analogous statement was an essential tool for proving properties of Approximate Coordinate Ascent Boosting [13], although the proof here (for which we give a sketch) is significantly more difficult, since we cannot use the important trick used in our previous work for (the equivalent of) Lemma 1. As usual, the weak learning algorithm always achieves an edge of at least ρ for the calculation to hold.

Theorem 2.

$$g_{t+1} - g_t \geq \frac{1}{2}\frac{\alpha_t(r_t - g_t)}{s_{t+1}}.$$

Sketching the proof, we consider α_t, τ_t, d_+, and d_- as functions of three basic independent variables $r := r_t$, $g := g_t$ and, d_0, with ranges $0 < r < 1$, $0 \leq g < r$, and $0 \leq d_0 \leq 1 - r$. Define

$$\Gamma_{r,g,d_0} := \frac{-\ln \tau_t}{\alpha_t}, \text{ and } \mathcal{B}_{r,g,d_0} := \frac{\Gamma_{r,g,d_0} - g}{r - g}.$$

Lemma 1. $\mathcal{B}_{r,g,d_0} > 1/2$.

This lemma is a monstrous calculus problem in three variables, for which the proof will be given in a longer version of this paper. Using only this lemma, we can prove the theorem directly.

Proof. (of Theorem 2) Let us unravel the notation a bit. Lemma 1 says:

$$\frac{-\ln \tau_t}{\alpha_t} = \Gamma_{r,g,d_0} > \frac{r_t + g_t}{2} \quad \Rightarrow \quad -\ln \tau_t > \frac{(r_t + g_t)\alpha_t}{2}.$$

Incorporating the recursive equation (4),

$$g_{t+1} - g_t = \frac{1}{s_{t+1}}\left[-g_t\alpha_t - \ln \tau_t\right] > \frac{\alpha_t}{s_{t+1}}\left[-g_t + \frac{(r_t + g_t)}{2}\right] = \frac{1}{2}\frac{\alpha_t(r_t - g_t)}{s_{t+1}}. \quad \square$$

Theorem 2 is the main step in proving convergence theorems, for example:

Theorem 3. *The smooth ranking margin ranking algorithm converges to a maximum margin solution, i.e.,* $\lim_{t\to\infty} g_t = \rho$.

Besides Theorem 2, the only other key step in the proof of Theorem 3 is:

Lemma 2.

$$\lim_{t\to\infty} \frac{\alpha_t}{s_{t+1}} = 0.$$

We omit the proof of Lemma 2, which uses (4), monotonicity and boundedness of the g_t sequence, and then (5).

Proof. (Of Theorem 3) The values of g_t constitute a non-decreasing sequence which is uniformly bounded by 1. Thus, a limit g_∞ exists, $g_\infty := \lim_{t\to\infty} g_t$. By (3), we know $g_t \leq \rho$ for all t. Thus, $g_\infty \leq \rho$. Suppose $g_\infty < \rho$, i.e., that $\rho - g_\infty \neq 0$. One can use an identical calculation to the one in Rudin et al. [13] to show that this assumption, together with Theorem 2 and Lemma 2 imply that $\lim_{t\to\infty} \alpha_t = 0$. Using this fact along with (5), we find:

$$g_\infty = \lim_{t\to\infty} g_t = \liminf_{t\to\infty} g_t = \liminf_{t\to\infty} \frac{-\tau_t'}{\tau_t} = \liminf_{t\to\infty} \frac{-(-d_+ e^{-\alpha_t} + d_- e^{\alpha_t})}{d_+ e^{-\alpha_t} + d_- e^{\alpha_t} + d_0} = \liminf_{t\to\infty} \frac{r_t}{1} \geq \rho.$$

This is a contradiction with the original assumption that $g_\infty < \rho$. It follows that $g_\infty = \rho$, or $\lim_{t\to\infty}(\rho - g_t) = 0$. Thus, the smooth ranking algorithm converges to a maximum margin solution. □

5 AdaBoost and RankBoost in the Bipartite Problem

In the bipartite ranking problem, every training instance falls into one of two categories, the "positive class" Y_+ and the "negative class" Y_-. Here, $\pi(\mathbf{x}_i, \mathbf{x}_k) = 1$ only when $\mathbf{x}_i \in Y_+$ and $\mathbf{x}_k \in Y_-$. Define $y_i = +1$ when $\mathbf{x}_i \in Y_+$, and $y_i = -1$ otherwise. The function $\tilde{F}$ now becomes an exponentiated version of the AUC, that is, since the step function obeys $\mathbf{1}_{x<0} \leq e^{-x}$, we have:

$$|Y_+||Y_-|(1 - \text{AUC}(\boldsymbol{\lambda})) = \sum_{i\in Y_+} \sum_{k\in Y_-} \mathbf{1}_{(\mathbf{M}\boldsymbol{\lambda})_{ik}<0} \leq \sum_{i\in Y_+} \sum_{k\in Y_-} e^{-(\mathbf{M}\boldsymbol{\lambda})_{ik}} = \tilde{F}(\boldsymbol{\lambda}).$$

The AUC has been written as the Wilcoxon-Mann-Whitney statistic (see [5]).

We now show that AdaBoost minimizes RankBoost's loss function $\tilde{F}$ under a very natural condition, namely, whenever the positive and negative instances contribute equally to AdaBoost's loss function.

We define AdaBoost's matrix $\mathbf{M}^{Ada}$ element-wise by $M_{ij}^{Ada} = y_i h_j(\mathbf{x}_i)$. Each crucial pair i, k has $i \in Y_+$ and $k \in Y_-$, so elements of $\mathbf{M}$ are: $M_{ikj} = h_j(\mathbf{x}_i) - h_j(\mathbf{x}_k) = y_i h_j(\mathbf{x}_i) + y_k h_j(\mathbf{x}_k) = M_{ij}^{Ada} + M_{kj}^{Ada}$. (To change from AdaBoost's usual $\{-1, 1\}$ hypotheses to RankBoost's usual $\{0, 1\}$ hypotheses, divide by 2.) Define vector $\mathbf{q}_{\boldsymbol{\lambda}}$ element-wise by $q_{\boldsymbol{\lambda},i} := e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda})_i}$ for $i = 1, ..., m$. Using this notation, we will write the objective functions for both AdaBoost and RankBoost. First, we define the following:

$$F_+(\boldsymbol{\lambda}) := \sum_{i\in Y_+} q_{\boldsymbol{\lambda},i} \text{ and } F_-(\boldsymbol{\lambda}) := \sum_{k\in Y_-} q_{\boldsymbol{\lambda},k}.$$

The objective function for AdaBoost is $F(\boldsymbol{\lambda}) := F_+(\boldsymbol{\lambda}) + F_-(\boldsymbol{\lambda})$. The objective function for RankBoost is: $\tilde{F}(\boldsymbol{\lambda}) = F_+(\boldsymbol{\lambda})F_-(\boldsymbol{\lambda})$. Thus, the balance between the positive and negative instances is different between the algorithms.

We define "F-skew", which measures the imbalance between positive and negative instances.

$$\text{F-skew}(\boldsymbol{\lambda}) := F_+(\boldsymbol{\lambda}) - F_-(\boldsymbol{\lambda}).$$

Theorem 4. *Assume* $\mathbf{M}^{Ada}$ *is such that* $\inf_{\boldsymbol{\lambda}} \tilde{F}(\boldsymbol{\lambda}) > 0$ *(the non-separable case). For any sequence* $\{\boldsymbol{\lambda}_t\}_{t=1}^{\infty}$ *such that*

$$\lim_{t\to\infty} F(\boldsymbol{\lambda}_t) = \inf_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}) \tag{7}$$

and $\lim_{t\to\infty} \text{F-skew}(\boldsymbol{\lambda}_t) = 0$*, then*

$$\lim_{t\to\infty} \tilde{F}(\boldsymbol{\lambda}_t) = \inf_{\boldsymbol{\lambda}} \tilde{F}(\boldsymbol{\lambda}). \tag{8}$$

Proof. It is possible that F or $\tilde{F}$ may have no minimizers. So, to describe (7) and (8), we use the trick from Collins et al. [4], who considered F and $\tilde{F}$ as functions of a variable where the infimum can be achieved. Define, for matrix $\bar{\mathbf{M}} \in \mathbb{R}^{\bar{m}\times n}$, the function

$$F_{\bar{\mathbf{M}}}(\boldsymbol{\lambda}) := \sum_{i=1}^{\bar{m}} e^{-(\bar{\mathbf{M}}\boldsymbol{\lambda})_i}.$$

Define $\bar{\mathcal{P}} := \{\mathbf{p} | \forall i\ p_i \geq 0,\ \forall j\ (\mathbf{p}^T\bar{\mathbf{M}})_j = 0\}$ and $\bar{\mathcal{Q}} := \{\mathbf{q} | \forall i\ q_i = e^{-(\bar{\mathbf{M}}\boldsymbol{\lambda})_i}$ for some $\boldsymbol{\lambda}\}$. We may thus consider $\bar{F}_{\bar{\mathbf{M}}}$ as a function of $\bar{\mathbf{q}}$, $\bar{F}_{\bar{\mathbf{M}}}(\bar{\mathbf{q}}) = \sum_{i=1}^{\bar{m}} \bar{q}_i$, where $\bar{\mathbf{q}} \in \bar{\mathcal{Q}}$. We know that since all $\bar{q}_i$'s are positive, the infimum of $\bar{F}$ occurs in a bounded region of $\bar{\mathbf{q}}$ space, which is just what we need.

Theorem 1 of Collins et al., which is taken directly from Lafferty, Della Pietra, and Della Pietra [11] implies that the following are equivalent:

1. $\bar{\mathbf{q}}^* \in \bar{\mathcal{P}} \cap$ closure $(\bar{\mathcal{Q}})$.
2. $\bar{\mathbf{q}}^* \in \text{argmin}_{\bar{\mathbf{q}} \in \text{ closure}(\bar{\mathcal{Q}})} \bar{F}_{\bar{\mathbf{M}}}(\bar{q})$.

Moreover, either condition is satisfied by exactly one vector $\bar{q}^*$.

The objective function for AdaBoost is $F = \bar{F}_{\mathbf{M}^{Ada}}$ and the objective for RankBoost is $\tilde{F} = \bar{F}_{\mathbf{M}}$, so the theorem holds for both objectives separately. For function F, denote $\bar{\mathbf{q}}^*$ as $\mathbf{q}^*$, also $\bar{\mathcal{P}}$ as $\mathcal{P}^{Ada}$ and $\bar{\mathcal{Q}}$ as $\mathcal{Q}^{Ada}$. For function $\tilde{F}$, denote $\bar{\mathbf{q}}^*$ as $\tilde{\mathbf{q}}^*$, also $\bar{\mathcal{P}}$ as $\tilde{\mathcal{P}}$ and $\bar{\mathcal{Q}}$ as $\tilde{\mathcal{Q}}$. Rewriting $\mathbf{q}^* \in \mathcal{P}^{Ada}$:

$$\sum_{i\in Y_+} q_i^* M_{ij}^{Ada} + \sum_{k\in Y_-} q_k^* M_{kj}^{Ada} = 0\ \ \forall\ j. \tag{9}$$

Define $\mathbf{q}_t$ element-wise by: $q_{t,i} := e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t)_i}$, for $i = 1, ..., m$ where the $\boldsymbol{\lambda}_t$'s are a sequence that obey (7), for example, a sequence produced by AdaBoost. Thus, $\mathbf{q}_t \in \mathcal{Q}^{Ada}$ automatically. Since $F(\mathbf{q}_t)$ converges to the minimum of F, one can show that the sequence of $\mathbf{q}_t$'s converges to $\mathbf{q}^*$ in ℓ_p. Now define vectors $\tilde{\mathbf{q}}_t$ element-wise by $\tilde{q}_{t,ik} := q_{t,i}q_{t,k} = \exp[-(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t)_i - (\mathbf{M}^{Ada}\boldsymbol{\lambda}_t)_k] = \exp[-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}]$. Automatically, $\tilde{\mathbf{q}}_t \in \tilde{\mathcal{Q}}$. For any pair i, k the limit of the $\tilde{q}_{t,ik}$'s is $\tilde{q}_{ik}^{\infty} := q_i^* q_k^*$. Thus, we need only to show $\tilde{\mathbf{q}}^{\infty} = \tilde{\mathbf{q}}^*$. We will do this by showing $\tilde{\mathbf{q}}^{\infty} \in \tilde{\mathcal{P}}$; due to the uniqueness of $\tilde{\mathbf{q}}^*$ as $\tilde{\mathcal{P}} \cap \text{closure}(\tilde{\mathcal{Q}})$, this will yield $\tilde{\mathbf{q}}^{\infty} = \tilde{\mathbf{q}}^*$.

Our assumption that the F-skew vanishes can be rewritten as:

$$\lim_{t\to\infty} [\sum_{i\in Y_+} q_{t,i} - \sum_{k\in Y_-} q_{t,k}] = 0, \quad \text{i.e.,}$$

$$\sum_{i \in Y_+} q_i^* = \sum_{k \in Y_-} q_k^*. \tag{10}$$

Consider the quantities $(\tilde{\mathbf{q}}^{\infty T}\mathbf{M})_j$. Remember, if these quantities are zero for every j, then $\tilde{\mathbf{q}}^\infty \in \tilde{\mathcal{P}}$ and we have proved the theorem.

$$(\tilde{\mathbf{q}}^{\infty T}\mathbf{M})_j = (\sum_{k \in Y_-} q_k^*)(\sum_{i \in Y_+} q_i^* M_{ij}^{Ada}) + (\sum_{i \in Y_+} q_i^*)(\sum_{k \in Y_-} q_k^* M_{kj}^{Ada}).$$

Incorporating (10), which is the condition that F-skew$(\mathbf{q}^*) = 0$, (11) becomes:

$$(\tilde{\mathbf{q}}^{\infty T}\mathbf{M})_j = (\sum_{i \in Y_+} q_i^*)[\sum_{i \in Y_+} q_i^* M_{ij}^{Ada} + \sum_{k \in Y_-} q_k^* M_{kj}^{Ada}].$$

In fact, according to (9), the bracket in this expression is zero for all j. Thus, $\tilde{\mathbf{q}}_\infty \in \tilde{\mathcal{P}}$. We have proved the theorem. □

Corollary 1. *If the constant weak hypothesis* $\forall \mathbf{x},\ h_j(\mathbf{x}) = 1$ *is one of the weak classifiers used to construct* $\mathbf{M}^{Ada}$*, and the* $\{\boldsymbol{\lambda}_t\}_t$ *sequence obeys (7), then* $\lim_{t \to \infty}$ F-skew$(\boldsymbol{\lambda}_t) = 0$*, and* $\{\boldsymbol{\lambda}_t\}_t$ *thus obeys (8) by Theorem 4.*

That is, any algorithm which minimizes F (such as AdaBoost) solves the ranking problem whenever the weak learning algorithm is capable of producing the constant hypothesis.

Proof. Recall that $\mathbf{q}^* \in \mathcal{P}^{Ada}$. Specifically writing this condition just for the constant weak classifier yields:

$$0 = \sum_{i \in Y_+} q_i^* y_i 1 + \sum_{k \in Y_-} q_k^* y_k 1 = \sum_{i \in Y_+} q_i^* - \sum_{k \in Y_-} q_k^* = \lim_{t \to \infty} \text{F-skew}(\boldsymbol{\lambda}_t). \qquad \square$$

Thus, AdaBoost and RankBoost are closely related indeed, since under this very weak condition (e.g., when the constant weak classifier is included), AdaBoost minimizes RankBoost's objective function. Given these results, it is now understandable (but still surprising) that AdaBoost performs so well as a ranking algorithm. One can directly use the convergence of the $\mathbf{q}_t$'s to show that AdaBoost produces exactly the same AUC value as RankBoost under this weak condition (in addition to the same value of the exponential loss). We expand on this in future work.

6 Conclusion

The three main results presented in this paper yield many new directions for future research. We gave a margin-based bound for general ranking. It is worth investigating the design of more specialized margin-based bounds for ranking. We described a new ranking algorithm, Smooth Margin Ranking, that maximizes the margin. It would be natural to compare the empirical performance of

the Smooth Margin Ranking algorithm and RankBoost. Finally, given the AUC optimization result proved for AdaBoost, one may ask why RankBoost, or another ranking algorithm, is needed in the non-separable case? The answer may lie in the convergence rate of AdaBoost versus that of RankBoost, which we are currently studying. Our observations suggest that RankBoost (understandably) has faster convergence to a high AUC value.

References

[1] Shivani Agarwal, Thore Graepel, Ralf Herbich, Sariel Har-Peled, and Dan Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

[2] Olivier Bousquet. New approaches to statistical learning theory. *Annals of the Institute of Statistical Mathematics*, 55(2):371–389, 2003.

[3] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms using difference performance metrics. Technical Report TR2005-1973, Cornell University, 2005.

[4] Michael Collins, Robert E. Schapire, and Yoram Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1/2/3), 2002.

[5] Corinna Cortes and Mehryar Mohri. AUC optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems 16*, 2004.

[6] Felipe Cucker and Steve Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.*, (39):1–49, 2002.

[7] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. In *Machine Learning: Proceedings of the Fifteenth International Conference*, 1998.

[8] Yoav Freund and Robert E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

[9] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 38(2):337–374, April 2000.

[10] Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1), February 2002.

[11] John D. Lafferty, Stephen Della Pietra, and Vincent Della Pietra. Statistical learning algorithms based on Bregman distances. In *Proceedings of the Canadian Workshop on Information Theory*, 1997.

[12] Cynthia Rudin, Ingrid Daubechies, and Robert E. Schapire. The dynamics of AdaBoost: Cyclic behavior and convergence of margins. *Journal of Machine Learning Research*, 5:1557–1595, December 2004.

[13] Cynthia Rudin, Robert E. Schapire, and Ingrid Daubechies. Analysis of boosting algorithms using the smooth margin function: A study of three algorithms. Submitted, 2004.

[14] Cynthia Rudin, Robert E. Schapire, and Ingrid Daubechies. Boosting based on a smooth margin. In *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*, pages 502–517, 2004.

[15] Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, October 1998.

A Proof of Theorem 1

We owe inspiration for this proof to the works of Cucker and Smale [6], Koltchinskii and Panchenko [10], and Bousquet [2].

We define a Lipschitz function $\phi : \mathbb{R} \to \mathbb{R}$ (with Lipschitz constant $\text{Lip}(\phi)$) which acts as our loss function, and gives us the margin. We will later use the same piecewise linear definition of ϕ as Koltchinskii and Panchenko [10], but for now, we require $\forall z,\ 0 \leq \phi(z) \leq 1$ and $\phi(z) = 1$ for $z < 0$. Since $\phi(z) \geq \mathbf{1}_{[z \leq 0]}$, we can define an upper bound for the misranking probability, namely $\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_{\mathcal{D}}\phi_f$, where:

$$\mathbb{P}_{\mathcal{D}}\phi_f := \frac{\mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}} \sim \mathcal{D}}[\phi(f(\bar{\mathbf{x}}) - f(\tilde{\mathbf{x}}))\pi(\bar{\mathbf{x}},\tilde{\mathbf{x}})]}{\mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}} \sim \mathcal{D}}\pi(\bar{\mathbf{x}},\tilde{\mathbf{x}})}.$$

The empirical error associated with $\mathbb{P}_{\mathcal{D}}\phi_f$ is:

$$\mathbb{P}_S\phi_f := \frac{\sum_{i=1}^m \sum_{k=1}^m \phi(f(\mathbf{x}_i) - f(\mathbf{x}_k))\pi(\mathbf{x}_i,\mathbf{x}_k)}{\sum_{i=1}^m \sum_{k=1}^m \pi(\mathbf{x}_i,\mathbf{x}_k)}.$$

First, we upper bound the misranking probability by two terms: the empirical error term $\mathbb{P}_S\phi_f$, and a term characterizing the deviation of $\mathbb{P}_S\phi_f$ from $\mathbb{P}_{\mathcal{D}}\phi_f$ uniformly:

$$\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_{\mathcal{D}}\phi_f = \mathbb{P}_{\mathcal{D}}\phi_f - \mathbb{P}_S\phi_f + \mathbb{P}_S\phi_f \leq \sup_{\bar{f} \in \mathcal{F}} (\mathbb{P}_{\mathcal{D}}\phi_{\bar{f}} - \mathbb{P}_S\phi_{\bar{f}}) + \mathbb{P}_S\phi_f.$$

The proof of the theorem involves an upper bound on the first term. First, define $L(f)$ as follows: $L(f) := \mathbb{P}_{\mathcal{D}}\phi_f - \mathbb{P}_S\phi_f$. The following lemma (for which the proof is omitted) is true for every training set S:

Lemma 3. *For any two functions $f_1, f_2 \in L_\infty(\mathcal{X})$,*

$$L(f_1) - L(f_2) \leq 4\text{Lip}(\phi)||f_1 - f_2||_\infty.$$

The following step is due to Cucker and Smale [6]. Let $\ell_\epsilon := \mathcal{N}\left(\mathcal{F}, \frac{\epsilon}{8\text{Lip}(\phi)}\right)$, the covering number of $\mathcal{F}$ by L_∞ disks of radius $\frac{\epsilon}{8\text{Lip}(\phi)}$. Define $f_1, f_2, ..., f_{\ell_\epsilon}$ to be the centers of such a cover, i.e., the collection of L_∞ disks D_p centered at f_p and with radius $\frac{\epsilon}{8\text{Lip}(\phi)}$ is a cover for $\mathcal{F}$. The following lemma (proof omitted) shows we do not lose too much by using f_p as a representative for disk D_p.

Lemma 4.

$$\mathbb{P}_{S \sim \mathcal{D}^m}\{\sup_{f \in D_p} L(f) \geq \epsilon\} \leq \mathbb{P}_{S \sim \mathcal{D}^m}\{L(f_p) \geq \frac{\epsilon}{2}\}.$$

Now we incorporate the fact that the training set is chosen randomly.

Lemma 5.

$$\mathbb{P}_{S \sim \mathcal{D}^m}\{L(f) \geq \epsilon/2\} \leq 2\exp\left[-\frac{m(\epsilon E)^2}{8}\right].$$

Proof. To make notation easier for this lemma, we introduce some shorthand notation:

$$\text{top}_{\mathcal{D}} := \mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}} \sim \mathcal{D}}\phi(f(\bar{\mathbf{x}}) - f(\tilde{\mathbf{x}}))\pi(\bar{\mathbf{x}},\tilde{\mathbf{x}}), \quad \text{bot}_{\mathcal{D}} := E := \mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}} \sim \mathcal{D}}\pi(\bar{\mathbf{x}},\tilde{\mathbf{x}}),$$
$$\text{top}_S := \frac{1}{m(m-1)}\sum_{i=1}^m \sum_{k=1}^m \phi(f(\mathbf{x}_i) - f(\mathbf{x}_k))\pi(\mathbf{x}_i,\mathbf{x}_k), \quad \text{bot}_S := \frac{1}{m(m-1)}\sum_{i=1}^m \sum_{k=1}^m \pi(\mathbf{x}_i,\mathbf{x}_k).$$

Since diagonal terms are $\pi(\mathbf{x}_i, \mathbf{x}_i) = 0$, $\text{top}_{\mathcal{D}} = \mathbb{E}_{S\sim\mathcal{D}^m}\text{top}_S$ and $\text{bot}_{\mathcal{D}} = \mathbb{E}_{S\sim\mathcal{D}^m}\text{bot}_S$. Thus, we can bound the difference between top_S and $\text{top}_{\mathcal{D}}$ using large deviation bounds; same for bot_S and $\text{bot}_{\mathcal{D}}$. One can show that the replacement of one instance changes top_S (or bot_S) by at most $1/m$. Thus, McDiarmid's inequality implies, for every $\epsilon_1 > 0$:

$$\mathbb{P}\{\text{top}_{\mathcal{D}} - \text{top}_S \geq \epsilon_1\} \leq \exp[-2\epsilon_1^2 m] \text{ and } \mathbb{P}\{\text{bot}_S - \text{bot}_{\mathcal{D}} \geq \epsilon_1\} \leq \exp[-2\epsilon_1^2 m].$$

We will specify ϵ_1 in terms of ϵ later. Consider the following event:

$$\text{top}_{\mathcal{D}} - \text{top}_S < \epsilon_1 \quad \text{and} \quad \text{bot}_S - \text{bot}_{\mathcal{D}} < \epsilon_1.$$

By the union bound, this event is true with probability at least $1-2\exp[-2\epsilon_1^2 m]$. When the event is true, we can rearrange the equations to be a bound on $L(f)$:

$$L(f) = \frac{\text{top}_{\mathcal{D}}}{\text{bot}_{\mathcal{D}}} - \frac{\text{top}_S}{\text{bot}_S} < \frac{\text{top}_{\mathcal{D}}}{\text{bot}_{\mathcal{D}}} - \frac{\text{top}_{\mathcal{D}} - \epsilon_1}{\text{bot}_{\mathcal{D}} + \epsilon_1} =: \epsilon/2.$$

Above, we have just specified the value for ϵ_1 in terms of ϵ. Let us solve for ϵ_1:

$$\epsilon_1 = \frac{\epsilon \text{bot}_{\mathcal{D}}}{2 - \epsilon + 2\frac{\text{top}_{\mathcal{D}}}{\text{bot}_{\mathcal{D}}}} \geq \frac{\epsilon E}{4}.$$

Here, we have used $E := \text{bot}_{\mathcal{D}}$, and by definition, $\text{top}_{\mathcal{D}} \leq \text{bot}_{\mathcal{D}}$. We directly have:

$$1 - 2\exp[-2\epsilon_1^2 m] \geq 1 - 2\exp\left(-2m\left[\frac{\epsilon E}{4}\right]^2\right).$$

Therefore, from our earlier application of McDiarmid, with probability at least $1 - 2\exp\left[-\frac{m(\epsilon E)^2}{8}\right]$ the following holds: $L(f) < \epsilon/2$. □

Proof. (**of Theorem** 1) Since the D_p are a cover of $\mathcal{F}$, it is true that

$$\sup_{f\in\mathcal{F}} L(f) \geq \epsilon \iff \exists p \leq \ell_\epsilon \text{ such that } \sup_{f\in D_p} L(f) \geq \epsilon.$$

First applying the union bound, then applying Lemma 4, and then Lemma 5, we find:

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left\{\sup_{f\in\mathcal{F}} L(f) \geq \epsilon\right\} \leq \sum_{p=1}^{\ell_\epsilon} \mathbb{P}_{S\sim\mathcal{D}^m}\left\{\sup_{f\in D_p} L(f) \geq \epsilon\right\} \leq \sum_{p=1}^{\ell_\epsilon} \mathbb{P}_{S\sim\mathcal{D}^m}\left\{L(f_p) \geq \epsilon/2\right\}$$

$$\leq \sum_{p=1}^{\ell_\epsilon} 2\exp\left[-\frac{m(\epsilon E)^2}{8}\right] = \mathcal{N}\left(\mathcal{F}, \frac{\epsilon}{8\text{Lip}(\phi)}\right) 2\exp\left[-\frac{m(\epsilon E)^2}{8}\right].$$

Now, with probability at least $1 - \mathcal{N}\left(\mathcal{F}, \frac{\epsilon}{8\text{Lip}(\phi)}\right) 2\exp\left[-\frac{m(\epsilon E)^2}{8}\right]$, we have $\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_S\phi_f + \epsilon$. Let $\phi(z) = 1$ for $z \leq 0$, $\phi(z) = 0$ for $z \geq \theta$, and let $\phi(z)$ be linear in between with slope $1/\theta$. Thus, $\text{Lip}(\phi) = 1/\theta$. Since $\phi(z) \leq 1$ for $z \leq \theta$, we have:

$$\mathbb{P}_S\phi_f = \frac{\sum_{i=1}^m \sum_{k=1}^m \phi(f(\mathbf{x}_i) - f(\mathbf{x}_k))\pi(\mathbf{x}_i, \mathbf{x}_k)}{\sum_{i=1}^m \sum_{k=1}^m \pi(\mathbf{x}_i, \mathbf{x}_k)} \leq \mathbb{P}_S\{\text{margin}_f \leq \theta\}.$$

Thus, with probability at least $1 - \mathcal{N}\left(\mathcal{F}, \frac{\epsilon\theta}{8}\right) 2\exp\left[-\left(\frac{m(\epsilon E)^2}{8}\right)\right]$, we have $\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_S\{\text{margin}_f \leq \theta\} + \epsilon$. Thus, the theorem has been proved. □