Margin-based Ranking and an Equivalence between AdaBoost and RankBoost

Journal of Machine Learning Research 10 (2009) 2193-2232 Submitted 2/08; Revised 7/09; Published 10/09

Margin-based Ranking and an Equivalence between AdaBoost and RankBoost

Cynthia Rudin[*] RUDIN@MIT.EDU
MIT Sloan School of Management
Cambridge, MA 02142

Robert E. Schapire SCHAPIRE@CS.PRINCETON.EDU
Department of Computer Science
35 Olden Street
Princeton University
Princeton NJ 08540

Editor: Nicolas Vayatis

Abstract

We study boosting algorithms for learning to rank. We give a general margin-based bound for ranking based on covering numbers for the hypothesis space. Our bound suggests that algorithms that maximize the ranking margin will 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." Finally, we prove that AdaBoost and RankBoost are equally good for the problems of bipartite ranking and classification in terms of their asymptotic behavior on the training set. Under natural conditions, AdaBoost achieves an area under the ROC curve that is equally as good as RankBoost's; furthermore, RankBoost, when given a specific intercept, achieves a misclassification error that is as good as AdaBoost's. This may help to explain the empirical observations made by Cortes and Mohri, and Caruana and Niculescu-Mizil, about the excellent performance of AdaBoost as a bipartite ranking algorithm, as measured by the area under the ROC curve.

Keywords: ranking, RankBoost, generalization bounds, AdaBoost, area under the ROC curve

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 probably go to see the movie closest to the top of the list that is playing at the local theater. She does not want to waste her time and money on a movie she probably will not like.

The information she provides is as follows: for many different pairs of movies she has seen, she will tell the learning algorithm whether or not she likes the first movie better than the second one.[1] This allows her to rank whichever pairs of movies she wishes, allowing for the possibility of ties

∗. Also at Center for Computational Learning Systems, Columbia University, 475 Riverside Drive MC 7717, New York, NY 10115.

1. In practice, she could simply rate the movies, but this gives pairwise information also. The pairwise setting is strictly more general in this sense.

between movies, and the possibility that certain movies cannot necessarily be compared by her (for instance, she may not wish to directly compare cartoons with action movies). Sylvia does not need to be consistent, in the sense that she may rank $\mathbf{a} > \mathbf{b} > \mathbf{c} > \mathbf{a}$. (The loss function and algorithm will accommodate this. See Martin Gardner's amusing article (Gardner, 2001) on how nontransitivity can arise naturally in many situations.) 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, called "weak rankers" or "ranking features," who have also ranked pairs of movies. The learning algorithm must try to combine the views of the weak rankers in order to match Sylvia's preferences, and generate a recommendation list that will generalize her views. In this paper, our goal is to design and study learning algorithms for ranking problems such as this collaborative filtering task.

The ranking problem was studied in depth by Freund et al. (2003), where the RankBoost algorithm was introduced. In this setting, the ranked list is constructed using a linear combination of the weak rankers. Ideally, this combination should minimize the probability that a crucial pair is misranked, that is, the probability that the second movie in the crucial pair is ranked above the first. RankBoost aims to minimize an exponentiated version of this misranking probability.

A special case of the general ranking problem is the "bipartite" ranking problem, where there are only two classes: a positive class (good movies) and a negative class (bad movies). In this case, the misranking probability is the probability that a good movie will be ranked below a bad movie. This quantity is an affine transformation of the (negative of the) area under the Receiver-Operator-Characteristic curve (AUC).

Bipartite ranking is different from the problem of classification; if, for a given data set, the misclassification error is zero, then the misranking error must also be zero, but the converse is not necessarily true. For the ranking problem, the examples are viewed relative to each other and the decision boundary is irrelevant.

Having described the learning setting, we can now briefly summarize our three main results.

- *Generalization bound:* In Section 3, we provide a margin-based bound for ranking in the general setting described above. Our ranking margin is defined in analogy with the classification margin, and the complexity measure for the hypothesis space is a "sloppy covering number," which yields, as a corollary, a bound in terms of the L_∞ covering number. Our bound indicates that algorithms that maximize the margin will generalize well.

- *Smooth margin ranking algorithm:* We present a ranking algorithm in Section 4 designed to maximize the margin. Our algorithm is based on a "smooth margin," and we present an analysis of its convergence.

- *An equivalence between AdaBoost and RankBoost:* A remarkable property of AdaBoost is that it not only solves the classification problem, but simultaneously solves the same problem of bipartite ranking as its counterpart, RankBoost. This is proved in Section 5. One does not need to alter AdaBoost in any way for this property to hold. Conversely, the solution of RankBoost can be slightly altered to achieve a misclassification loss that is equally as good as AdaBoost's.

We now provide some background and related results.

Generalization bounds are useful for showing that an algorithm can generalize beyond its training set, in other words, that prediction is possible. More specifically, bounds indicate that a small

probability of error will most likely be achieved through a proper balance of the empirical error and the complexity of the hypothesis space. This complexity can by measured by many informative quantities; for instance, the VC dimension, which is linked in a fundamental way to classification, and the Rademacher and Gaussian complexities (Bartlett and Mendelson, 2002). The use of these quantities is tied to a kind of natural symmetry that typically exists in such problems, for instance, in the way that positive and negative examples are treated symmetrically in a classification setting. The limited bipartite case has this symmetry, but not the more general ranking problem that we have described. Prior bounds on ranking have either made approximations in order to use the VC Dimension for the general problem (as discussed by Clemençon et al., 2005, 2007, who work on statistical aspects of ranking) or focused on the bipartite case (Freund et al., 2003; Agarwal et al., 2005; Usunier et al., 2005). For our bound, we choose a covering number in the spirit of Bartlett (1998). The covering number is a general measure of the capacity of the hypothesis space; it does not lend itself naturally to classification like the VC dimension, is not limited to bipartite ranking, nor does it require symmetry in the problem. Thus, we are able to work around the lack of symmetry in this setting. In fact, a preliminary version of our work (Rudin et al., 2005) has been extended to a highly nonsymmetric setting, namely the case where the top part of the list is considered more important (Rudin, 2009). Several other recent works also consider this type of highly nonsymmetric setting for ranking (Dekel et al., 2004; Cossock and Zhang, 2008; Clemençon and Vayatis, 2007; Shalev-Shwartz and Singer, 2006; Le and Smola, 2007).

When deriving generalization bounds, it is important to consider the "separable" case, where all training instances are correctly handled by the learning algorithm so that the empirical error is zero. In the case of bipartite ranking, the separable case means that all positive instances are ranked above all negative instances, and the area under the ROC curve is exactly 1. In the separable case for classification, one important indicator of a classifier's generalization ability is the "margin." The margin has proven to be an important quantity in practice for determining an algorithm's generalization ability, for example, in the case of AdaBoost (Freund and Schapire, 1997) and support vector machines (SVMs) (Cortes and Vapnik, 1995). Although there has been some work devoted to generalization bounds for ranking as we have mentioned (Clemençon et al., 2005, 2007; Freund et al., 2003; Agarwal et al., 2005; Usunier et al., 2005), the bounds that we are aware of are not margin-based, and thus do not provide this useful type of discrimination between ranking algorithms in the separable case.

Since we are providing a general margin-based bound for ranking in Section 3, we derive algorithms which create large margins. For the classification problem, it was proved that AdaBoost does not always fully maximize the (classification) margin (Rudin et al., 2004). In fact, AdaBoost does not even necessarily make progress towards increasing the margin at every iteration. Since AdaBoost (for the classification setting) and RankBoost (for the ranking setting) were derived analogously for the two settings, RankBoost does not directly maximize the ranking margin, and it does not necessarily 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 margin for ranking at every iteration; this is the main step needed in proving convergence and convergence rates. This algorithm is analogous to the smooth margin classification algorithm "approximate coordinate ascent boosting" (Rudin et al., 2007) in its derivation, but the analogous proof that progress occurs at each iteration is much trickier; hence we present this proof here, along with a theorem stating that this algorithm converges to a maximum margin solution.

Although AdaBoost and RankBoost were derived analogously for the two settings, the parallels between AdaBoost and RankBoost are deeper than their derivations. A number of papers, including those of Cortes and Mohri (2004) and Caruana and Niculescu-Mizil (2006) have noted that in fact, 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. Or, stated another way, AdaBoost often achieves a large area under the ROC curve. In Section 5, we present a possible explanation for these experimental observations. Namely, we show that if the weak learning algorithm is capable of producing the constant classifier (the classifier whose value is always one), then remarkably, AdaBoost and RankBoost produce equally good solutions to the ranking problem in terms of loss minimization and area under the ROC curve on the training set. More generally, we define a quantity called "F-skew," an exponentiated version of the "skew" used in the expressions of Cortes and Mohri (2004, 2005) and Agarwal et al. (2005). If the F-skew vanishes, AdaBoost minimizes the exponentiated ranking loss, which is the same loss that RankBoost explicitly minimizes; thus, the two algorithms will produce equally good solutions to the exponentiated problem. Moreover, if AdaBoost's set of weak classifiers includes the constant classifier, the F-skew always vanishes. From there, it is only a small calculation to show that AdaBoost and RankBoost achieve the same asymptotic AUC value whenever it can be defined. An analogous result does not seem to hold true for support vector machines; SVMs designed to maximize the AUC only seem to yield the same AUC as the "vanilla" classification SVM in the separable case, when the AUC is exactly one (Rakotomamonjy, 2004; Brefeld and Scheffer, 2005). The main result may be useful for practitioners: if the cost of using RankBoost is prohibitive, it may be useful to consider AdaBoost to solve the ranking problem.

The converse result also holds, namely that a solution of RankBoost can be slightly modified so that the F-skew vanishes, and the asymptotic misclassification loss is equal to AdaBoost's whenever it can be defined.

We proceed from the most general to the most specific. First, in Section 3 we provide a margin-based bound for general ranking. In Sections 4.1 and 4.2 we fix the form of the hypothesis space to match that of RankBoost, that is, the space of binary functions. Here, we discuss RankBoost, AdaBoost and other coordinate-based ranking algorithms, and introduce the smooth margin ranking algorithm. In Section 5, we focus on the bipartite ranking problem, and 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 where the training error vanishes, and Sections 4.1 and 5 focus on the nonseparable case. Sections 6, 7, and 8 contain the major proofs.

A preliminary version of this work appeared in a conference paper with Cortes and Mohri (Rudin et al., 2005). Many of the results from that work have been made more general here.

2. Notation

We use notation similar to Freund et al. (2003). The training data for the supervised ranking problem consists of *instances* and their *truth function* values. The *instances*, denoted by S, are $\{\mathbf{x}_i\}_{i=1,\ldots,m}$, where $\mathbf{x}_i \in \mathcal{X}$ for all i. The set $\mathcal{X}$ is arbitrary and may be finite or infinite, usually $\mathcal{X} \subset \mathbb{R}^N$. In the case of the movie ranking problem, the $\mathbf{x}_i$'s are the movies and $\mathcal{X}$ is the set of all possible movies. We assume $\mathbf{x}_i \in \mathcal{X}$ are chosen independently and at random (iid) from a fixed but unknown probability distribution $\mathcal{D}$ on $\mathcal{X}$ (assuming implicitly that anything that needs to be measurable is measurable). The notation $\mathbf{x} \sim \mathcal{D}$ means $\mathbf{x}$ is chosen randomly according to distribution $\mathcal{D}$. The notation $S \sim \mathcal{D}^m$

means each of the m elements of the training set S are chosen independently at random 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 $\pi(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = 1$, this means that the pair $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}$ is a *crucial pair*: $\mathbf{x}^{(1)}$ should be ranked more highly than $\mathbf{x}^{(2)}$. We will consider a non-noisy case where π is deterministic, which means $\pi(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) = 0$, meaning that $\mathbf{x}^{(1)}$ should not be ranked higher than itself, and also that $\pi(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = 1$ implies $\pi(\mathbf{x}^{(2)}, \mathbf{x}^{(1)}) = 0$, meaning that if $\mathbf{x}^{(1)}$ is ranked more highly than $\mathbf{x}^{(2)}$, then $\mathbf{x}^{(2)}$ should not be ranked more highly than $\mathbf{x}^{(1)}$. It is possible to have $\pi(\mathbf{a}, \mathbf{b}) = 1$, $\pi(\mathbf{b}, \mathbf{c}) = 1$, and $\pi(\mathbf{c}, \mathbf{a}) = 1$, in which case the algorithm will always suffer some loss; we will be in the nonseparable case when this occurs. The total number of crucial training pairs can be no larger than $m(m-1)/2$ based on the rules of π, and should intuitively be of the order m^2 in order for us to perform ranking with sufficient accuracy. We assume that for each pair of training instances $\mathbf{x}_i, \mathbf{x}_k$ we receive, we also receive the value of $\pi(\mathbf{x}_i, \mathbf{x}_k)$. In a more general model, we allow the value $\pi(\mathbf{x}_i, \mathbf{x}_k)$ to be generated probabilisitically conditional on each training pair $\mathbf{x}_i, \mathbf{x}_k$. For the generalization bounds in this paper, for simplicity of presentation, we do not consider this more general model, although all of our results can be shown to hold in the more general case as well. The quantity $E := \mathbb{E}_{\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \sim \mathcal{D}}[\pi(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})]$ is the expected proportion of pairs in the database that are crucial pairs, $0 \leq E \leq 1/2$.

Back to the collaborative filtering example, to obtain the training set, Sylvia is given a random sample of movies, chosen randomly from the distribution of movies being shown in the theater. Sylvia must see these training movies and tell us all pairs of these movies such that she would rank the first above the second to determine values of the truth function π.

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 instance, we do not care if $f(\mathbf{x}^{(1)}) = .4$ and $f(\mathbf{x}^{(2)}) = .1$, only that $f(\mathbf{x}^{(1)}) > f(\mathbf{x}^{(2)})$, which we interpret to mean that $\mathbf{x}^{(1)}$ is predicted by f to be ranked higher (better) than $\mathbf{x}^{(2)}$. Also, the function f should be bounded, $f \in L_\infty(\mathcal{X})$ (or in the case where $|\mathcal{X}|$ is finite, $f \in \ell_\infty(\mathcal{X})$).

In the usual setting of boosting for classification, $|f(\mathbf{x})| \leq 1$ for all $\mathbf{x}$ and the *margin of training instance i* (with respect to classifier f) is defined by Schapire et al. (1998) to be $y_i f(\mathbf{x}_i)$, where y_i is the classification label, $y_i \in \{-1, 1\}$. The *margin of classifier* f is defined to be the minimum margin over all training instances, $\min_i y_i f(\mathbf{x}_i)$. Intuitively, the margin tells us how much the classifier 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 normalize our bounded function f so that $0 \leq f \leq 1$. The *margin of crucial pair* $\mathbf{x}_i$,$\mathbf{x}_k$ (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 defined to be the minimum margin over all crucial pairs,

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

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.

In Section 5 we will discuss the problem of bipartite ranking. Bipartite ranking is a subset of the general ranking framework we have introduced. In the bipartite ranking problem, every training

instance falls into one of two categories, the positive class Y_+ and the negative class Y_-. To transform this into the general framework, take $\pi(\mathbf{x}_i, \mathbf{x}_k) = 1$ for each pair $i \in Y_+$ and $k \in Y_-$. That is, a crucial pair exists between an element of the positive class and an element of the negative class. The class of each instance is assumed deterministic, consistent with the setup described earlier. Again, the results can be shown to hold in the case of nondeterministic class labels.

It may be tempting to think of the ranking framework as if it were just classification over the space $\mathcal{X} \times \mathcal{X}$. However, this is not the case; the examples are assumed to be drawn randomly from $\mathcal{X}$, rather than pairs of examples drawn from $\mathcal{X} \times \mathcal{X}$. Furthermore, the scoring function f has domain $\mathcal{X}$, that is, in order to produce a single ranked list, we should have $f : \mathcal{X} \to \mathbb{R}$ rather than $f : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. In the latter case, one would need an additional mechanism to reconcile the scores to produce a single ranked list. Furthermore, the bipartite ranking problem does not have the same goal as classification even though the labels are $\{-1, +1\}$. In classification, the important quantity is the misclassification error involving the sign of f, whereas for bipartite ranking, the important quantity is perhaps the area under the ROC curve, relying on differences between f values. A change in the position of one example can change the bipartite ranking loss without changing the misclassification error and vice versa.

3. A Margin-Based Bound for Ranking

Bounds in learning theory are useful for telling us which quantities (such as the margin) are involved in the learning process (see Bousquet, 2003, for discussion on this matter). In this section, we provide a margin-based bound for ranking, which gives us an intuition for separable-case ranking and yields theoretical encouragement for margin-based ranking algorithms. The quantity we hope to minimize here is the misranking probability; for two randomly chosen instances, if they are a crucial pair, we want to minimize the probability that these instances will be misranked. Formally, this misranking probability is:

$$\begin{aligned}\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} &:= \mathbb{P}_{\mathcal{D}}\{f(\bar{\mathbf{x}}) \le f(\tilde{\mathbf{x}}) \mid \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1\} = \mathbb{E}_{\bar{\mathbf{x}}, \tilde{\mathbf{x}} \sim \mathcal{D}}[\mathbf{1}_{[f(\bar{\mathbf{x}}) \le f(\tilde{\mathbf{x}})]} \mid \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1] \\ &= \frac{\mathbb{E}_{\bar{\mathbf{x}}, \tilde{\mathbf{x}} \sim \mathcal{D}}[\mathbf{1}_{[f(\bar{\mathbf{x}}) \le f(\tilde{\mathbf{x}})]} \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}})]}{\mathbb{E}_{\bar{\mathbf{x}}, \tilde{\mathbf{x}} \sim \mathcal{D}}[\pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}})]} = \frac{\mathbb{E}_{\bar{\mathbf{x}}, \tilde{\mathbf{x}} \sim \mathcal{D}}[\mathbf{1}_{[f(\bar{\mathbf{x}}) \le f(\tilde{\mathbf{x}})]} \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}})]}{E}. \end{aligned} \tag{1}$$

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

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

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

We make this terminology more general, by allowing it to include a margin of θ. For the bound we take $\theta > 0$:

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

that is, $\mathbb{P}_S\{\text{margin}_f \leq \theta\}$ is the fraction of crucial pairs in $S \times S$ with margin not 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 a "sloppy covering number" of the sort used by Schapire et al. (1998). Since such a covering number can be bounded by an L_∞ covering number, we will immediately obtain L_∞ covering number bounds as well, including a strict improvement on the one derived in the preliminary version of our work (Rudin et al., 2005). Here, we implicitly assume that $\mathcal{F} \subset L_\infty(\mathcal{X})$, $f \in \mathcal{F}$ are everywhere defined.

We next define sloppy covers and sloppy covering numbers.

Definition 1 *For $\varepsilon, \theta \geq 0$, a set $\mathcal{G}$ is a* θ-sloppy ε-cover *for $\mathcal{F}$ if for all $f \in \mathcal{F}$ and for all probability distributions $\mathcal{D}$ on $\mathcal{X}$, there exists $g \in \mathcal{G}$ such that*

$$\mathbb{P}_{\mathbf{x}\sim D}[|f(\mathbf{x}) - g(\mathbf{x})| \geq \theta] \leq \varepsilon.$$

The corresponding sloppy covering number *is the size of the smallest θ-sloppy ε-cover $\mathcal{G}$, and is written $\mathcal{N}(\mathcal{F}, \theta, \varepsilon)$.*

The *L_∞ covering number* $\mathcal{N}_\infty(\mathcal{F}, \varepsilon)$ is defined as the minimum number of (open) balls of radius ε needed to cover $\mathcal{F}$, using the L_∞ metric. Since $\|f - g\|_\infty < \theta$ implies that $\mathbb{P}_{\mathbf{x}\sim D}[|f(\mathbf{x}) - g(\mathbf{x})| \geq \theta] = 0$, we have that the sloppy covering number $\mathcal{N}(\mathcal{F}, \theta, \varepsilon)$ is never more than $\mathcal{N}_\infty(\mathcal{F}, \theta)$, and in some cases it can be exponentially smaller, such as for convex combinations of binary functions as discussed below.

Here is our main theorem, which is proved in Section 6:

Theorem 2 *(Margin-based generalization bound for ranking) For $\varepsilon > 0$, $\theta > 0$ with probability at least*

$$1 - 2\mathcal{N}\left(\mathcal{F}, \frac{\theta}{4}, \frac{\varepsilon}{8}\right) \exp\left[-\frac{m(\varepsilon E)^2}{8}\right]$$

over the random choice of the training set S, every $f \in \mathcal{F}$ satisfies:

$$\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_S\{\text{margin}_f \leq \theta\} + \varepsilon.$$

In other words, the misranking probability is upper bounded by the fraction of instances with margin below θ, plus ε; this statement is true with probability depending on m, E, θ, ε, and $\mathcal{F}$.

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 can in no way influence $\mathbb{E}_{\mathbf{x}_i, \mathbf{x}_k \sim S}[\pi(\mathbf{x}_i, \mathbf{x}_k)]$ in our setting, since we are choosing training instances randomly. E can be viewed as a constant, where recall $0 < E \leq 1/2$. If $E = 0$, it means that there is no information about the relative ranks of examples, and accordingly the bound becomes trivial. Note that in the special bipartite case, E is the proportion of positive examples multiplied by the proportion of negative examples.

In order to see that this bound encourages the margin to be made large, consider the simplified case where the empirical error term is 0, that is, $\mathbb{P}_S\{\text{margin}_f \leq \theta\} = 0$. Now, the only place where

θ appears is in the covering number. In order to make the probability of success larger, the covering number should be made as small as possible, which implies that θ should be made as large as possible.

As a special case of the theorem, we consider the standard setting where f is a (normalized) linear combination of a dictionary of step functions (or "weak rankers"). In this case, we can show the following, proved in Section 6:

Lemma 3 *(Upper bound on covering numbers for convex combinations of binary weak classifiers) For the following hypothesis space:*

$$\mathcal{F} = \left\{ f : f = \sum_j \lambda_j h_j,\ \sum_j \lambda_j = 1,\ \forall j\ \lambda_j \geq 0,\ h_j : \mathcal{X} \to \{0,1\}, h_j \in \mathcal{H} \right\},$$

we have

$$\ln \mathcal{N}(\mathcal{F}, \theta, \varepsilon) \leq \frac{\ln|\mathcal{H}|\ \ln(2/\varepsilon)}{2\theta^2}.$$

Thus, Theorem 2 implies the following corollary.

Corollary 4 *(Margin-based generalization bound for ranking, convex combination of binary weak rankers) For $\varepsilon > 0$, $\theta > 0$ with probability at least*

$$1 - 2\exp\left(\frac{\ln|\mathcal{H}|\ \ln(16/\varepsilon)}{\theta^2/8} - \frac{m(\varepsilon E)^2}{8} \right)$$

over the random choice of the training set S, every $f \in \mathcal{F}$ satisfies:

$$\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_S\{\text{margin}_f \leq \theta\} + \varepsilon.$$

In this case, we can lower bound the right hand side by $1-\delta$ for an appropriate choice of ε. In particular, Corollary 4 implies that

$$\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_S\{\text{margin}_f \leq \theta\} + \varepsilon$$

with probability at least $1-\delta$ if

$$\varepsilon = \sqrt{\frac{4}{mE^2}\left[\frac{8\ln|\mathcal{H}|}{\theta^2}\ln\left(\frac{4mE^2\theta^2}{\ln|\mathcal{H}|}\right) + 2\ln\left(\frac{2}{\delta}\right)\right]}. \tag{2}$$

This bound holds provided that θ is not too small relative to m, specifically, if

$$m\theta^2 \geq \frac{64\ln|\mathcal{H}|}{E^2}.$$

Note that the bound in (2) is only polylogarithmic in $|\mathcal{H}|$.

As we have discussed above, Theorem 2 can be trivially upper bounded using the L_∞ covering number.

Corollary 5 *(Margin-based generalization bound for ranking, L_∞ covering numbers) For $\varepsilon > 0$, $\theta > 0$ with probability at least*

$$1 - 2\mathcal{N}_\infty\left(\mathcal{F}, \frac{\theta}{4}\right)\exp\left[-\frac{m(\varepsilon E)^2}{8}\right]$$

over the random choice of the training set S, every $f \in \mathcal{F}$ satisfies:

$$\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} \leq \mathbb{P}_S\{\text{margin}_f \leq \theta\} + \varepsilon.$$

Consider the case of a finite hypothesis space $\mathcal{F}$ where every function is far apart (in L_∞) from every other function. In this case, the covering number is equal to the number of functions. This is the worst possible case, where $\mathcal{N}\left(\mathcal{F}, \frac{\theta}{4}\right) = |\mathcal{F}|$ for any value of θ. In this case, we can solve for ε directly:

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

This indicates that the error may scale as $1/\sqrt{m}$. For the ranking problem, since we are dealing with pairwise relationships, we might expect worse dependence, but this does not appear to be the case. In fact, the dependence on m is quite reasonable in comparison to bounds for the problem of classification, which does not deal with examples pairwise. This is true not only for finite hypothesis spaces (scaling as $1/\sqrt{m}$) but also when the hypotheses are convex combinations of weak rankers (scaling as $\sqrt{\ln(m)/m}$).

4. Coordinate-Based Ranking Algorithms

In the previous section we presented a uniform bound that holds for all $f \in \mathcal{F}$. In this section, 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 2. Our bound suggests that given a fixed hypothesis space $\mathcal{F}$ and a fixed number of instances m we try to maximize the margin. We will do this using coordinate ascent. Coordinate ascent/descent is similar to gradient ascent/descent except that the optimization moves along single coordinate axes rather than along the gradient. (See Burges et al., 2005, for a gradient-based ranking algorithm based on a probabilistic model.) We first derive the plain coordinate descent version of RankBoost, and show that it is different from RankBoost itself. In Section 4.2 we define the smooth ranking margin $\tilde{G}$. Then we present the "smooth margin ranking" algorithm, and prove that it makes significant progress towards increasing this smooth ranking margin at each iteration, and converges to a maximum margin solution.

4.1 Coordinate Descent and Its Variation on RankBoost's Objective

We take the hypothesis space $\mathcal{F}$ to be the class of convex combinations of weak rankers $\{h_j\}_{j=1,\ldots,n}$, where $h_j : X \to \{0,1\}$. The function f is constructed as a normalized linear combination of the h_j's:

$$f = \frac{\sum_j \lambda_j h_j}{||\boldsymbol{\lambda}||_1},$$

where $||\boldsymbol{\lambda}||_1 = \sum_j \lambda_j$, $\lambda_j \geq 0$.

We will derive and mention many different algorithms based on different objective functions; here is a summary of them:

$F(\boldsymbol{\lambda})$: For the *classification* problem, AdaBoost minimizes its objective, denoted $F(\boldsymbol{\lambda})$, by coordinate descent.

$G(\boldsymbol{\lambda})$: For *classification limited to the separable case*, the algorithms "coordinate ascent boosting" and "approximate coordinate ascent boosting" are known to maximize the margin (Rudin et al., 2007). These algorithms are based on the smooth classification margin $G(\boldsymbol{\lambda})$.

$\tilde{F}(\boldsymbol{\lambda})$: For *ranking*, "coordinate descent RankBoost" minimizes its objective, denoted $\tilde{F}(\boldsymbol{\lambda})$, by coordinate descent. RankBoost itself minimizes $\tilde{F}(\boldsymbol{\lambda})$ by a variation of coordinate descent that chooses the coordinate with knowledge of the step size.

$\tilde{G}(\boldsymbol{\lambda})$: For *ranking limited to the separable case*, "smooth margin ranking" is an approximate coordinate ascent algorithm that maximizes the ranking margin. It is based on the smooth ranking margin $\tilde{G}(\boldsymbol{\lambda})$.

The objective function for RankBoost is a sum of exponentiated margins:

$$\tilde{F}(\boldsymbol{\lambda}) := \sum_{\{i,k:[\pi(\mathbf{x}_i,\mathbf{x}_k)=1]\}} e^{-\left(\sum_j \lambda_j h_j(\mathbf{x}_i) - \sum_j \lambda_j h_j(\mathbf{x}_k)\right)} = \sum_{ik\in\mathcal{C}_p} e^{-(\mathbf{M}\boldsymbol{\lambda})_{ik}},$$

where we have rewritten in terms of a structure $\mathbf{M}$, which describes how each individual weak ranker j ranks each crucial pair $\mathbf{x}_i, \mathbf{x}_k$; this will make notation significantly easier. Define an index set that enumerates all crucial pairs $\mathcal{C}_p = \{i,k : \pi(\mathbf{x}_i,\mathbf{x}_k) = 1\}$. Formally, the elements of the two-dimensional matrix $\mathbf{M}$ are defined as follows, for index ik corresponding to crucial pair $\mathbf{x}_i, \mathbf{x}_k$:

$$M_{ik,j} := h_j(\mathbf{x}_i) - h_j(\mathbf{x}_k).$$

The first index of $\mathbf{M}$ is ik, which runs over crucial pairs, that is, elements of $\mathcal{C}_p$, and the second index j runs over weak rankers. The size of $\mathbf{M}$ is $|\mathcal{C}_p| \times n$. Since the weak rankers are binary, the entries of $\mathbf{M}$ are within $\{-1, 0, 1\}$. The notation $(\cdot)_j$ means the j^{th} index of the vector, so that the following notation is defined:

$$(\mathbf{M}\boldsymbol{\lambda})_{ik} := \sum_{j=1}^{n} M_{ik,j}\lambda_j = \sum_{j=1}^{n} \lambda_j h_j(\mathbf{x}_i) - \lambda_j h_j(\mathbf{x}_k), \text{ and } (\mathbf{d}^T\mathbf{M})_j := \sum_{ik\in\mathcal{C}_p} d_{ik} M_{ik,j},$$

for $\boldsymbol{\lambda} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^{|\mathcal{C}_p|}$.

4.1.1 COORDINATE DESCENT RANKBOOST

Let us perform standard coordinate descent on this objective function, and we will call the algorithm "coordinate descent RankBoost." We will not get the RankBoost algorithm this way; we will show how to do this in Section 4.1.2. For coordinate descent on $\tilde{F}$, at iteration t, we first choose a direction j_t in which $\tilde{F}$ is decreasing very rapidly. The direction chosen at iteration t (corresponding to the choice of weak ranker j_t) in the "optimal" case (where the best weak ranker is chosen at each iteration) is given as follows. The notation $\mathbf{e}_j$ indicates a vector of zeros with a 1 in the j^{th} entry:

$$\begin{aligned} j_t \in \operatorname*{argmax}_j \left[-\frac{\partial \tilde{F}(\boldsymbol{\lambda}_t + \alpha\mathbf{e}_j)}{\partial\alpha}\Big|_{\alpha=0} \right] &= \operatorname*{argmax}_j \sum_{ik\in\mathcal{C}_p} e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}} M_{ik,j} \\ &= \operatorname*{argmax}_j \sum_{ik\in\mathcal{C}_p} d_{t,ik} M_{ik,j} = \operatorname*{argmax}_j (\mathbf{d}_t^T\mathbf{M})_j, \end{aligned} \quad (3)$$

where the "weights" $d_{t,ik}$ are defined by:

$$d_{t,ik} := \frac{e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}}}{\tilde{F}(\boldsymbol{\lambda}_t)} = \frac{e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}}}{\sum_{\tilde{ik}\in C_p} e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{\tilde{ik}}}}.$$

From this calculation, 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; maximizing $(\mathbf{d}_t^T\mathbf{M})_j$ encourages the algorithm to choose the most accurate weak ranker with respect to the weights.

The step size our coordinate descent algorithm chooses at iteration t is α_t, where α_t satisfies the following equation for the line search along direction j_t. Define $I_{t+} := \{ik : M_{ik,j_t} = 1\}$, and similarly, $I_{t-} := \{ik : M_{ik,j_t} = -1\}$. Also define $d_{t+} := \sum_{ik\in I_+} d_{t,ik}$ and $d_{t-} := \sum_{ik\in I_-} d_{t,ik}$. The line search is:

$$\begin{aligned}
0 &= -\frac{\partial \tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t})}{\partial \alpha}\Big|_{\alpha=\alpha_t} = \sum_{ik\in C_p} e^{-(\mathbf{M}(\boldsymbol{\lambda}_t+\alpha_t\mathbf{e}_{j_t}))_{ik}} M_{ik,j_t} \\
&= \sum_{ik\in I_{t+}} e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}} e^{-\alpha_t} - \sum_{ik\in I_{t-}} e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}} e^{\alpha_t} \\
0 &= d_{t+}e^{-\alpha_t} - d_{t-}e^{\alpha_t} \\
\alpha_t &= \frac{1}{2}\ln\left(\frac{d_{t+}}{d_{t-}}\right). \qquad (4)
\end{aligned}$$

Thus, we have derived the first algorithm, coordinate descent RankBoost. Pseudocode can be found in Figure 1. In order to make the calculation for $\mathbf{d}_t$ numerically stable, we write $\mathbf{d}_t$ in terms of its update from the previous iteration.

4.1.2 RANKBOOST

Let us contrast coordinate descent RankBoost with RankBoost. They both minimize the same objective $\tilde{F}$, but they differ by the ordering of steps: for coordinate descent RankBoost, j_t is calculated first, then α_t. In contrast, RankBoost uses the formula (4) for α_t in order to calculate j_t. In other words, at each step RankBoost selects the weak ranker that yields the largest decrease in the loss function, whereas coordinate descent RankBoost selects the weak ranker of steepest slope. Let us derive RankBoost. Define the following for iteration t (eliminating the t subscript):

$$\begin{aligned}
I_{+j} &:= \{ik : M_{ik,j} = 1\}, \quad I_{-j} := \{ik : M_{ik,j} = -1\}, \quad I_{0j} := \{ik : M_{ik,j} = 0\}, \\
d_{+j} &:= \sum_{ik\in I_{+j}} d_{t,ik}, \quad d_{-j} := \sum_{ik\in I_{-j}} d_{t,ik}, \quad d_{0j} := \sum_{ik\in I_{0j}} d_{t,ik}.
\end{aligned}$$

For each j, we take a step according to (4) of size $\frac{1}{2}\ln\frac{d_{+j}}{d_{-j}}$, and choose the j_t which makes the objective function $\tilde{F}$ decrease the most. That is:

$$\begin{aligned}
j_t : &= \operatorname*{argmin}_j \tilde{F}\left(\boldsymbol{\lambda}_t + \left(\frac{1}{2}\ln\frac{d_{+j}}{d_{-j}}\right)\mathbf{e}_{j_t}\right) = \operatorname*{argmin}_j \sum_{ik\in C_p} e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}} e^{-M_{ik,j}\frac{1}{2}\ln\frac{d_{+j}}{d_{-j}}} \\
&= \operatorname*{argmin}_j \sum_{ik} d_{t,ik}\left(\frac{d_{+j}}{d_{-j}}\right)^{-\frac{1}{2}M_{ik,j}} \\
&= \operatorname*{argmin}_j \left[2(d_{+j}d_{-j})^{1/2} + d_{0j}\right]. \qquad (5)
\end{aligned}$$

1. **Input:** Matrix $\mathbf{M}$, No. of iterations t_{max}

2. **Initialize:** $\lambda_{1,j} = 0$ for $j = 1,...,n$, $d_{1,ik} = 1/m$ for all ik

3. **Loop for** $t = 1,...,t_{max}$

   (a) $j_t \in \operatorname{argmax}_j (\mathbf{d}_t^T \mathbf{M})_j$ "optimal" case choice of weak classifier

   (b) $d_{t+} = \sum_{\{ik: M_{ik,j_t}=1\}} d_{t,ik}$, $d_{t-} = \sum_{\{ik: M_{ik,j_t}=-1\}} d_{t,ik}$

   (c) $\alpha_t = \frac{1}{2} \ln \left(\frac{d_{t+}}{d_{t-}} \right)$

   (d) $d_{t+1,ik} = d_{t,ik} e^{-M_{ik,j_t} \alpha_t} /$normaliz. for each crucial pair ik in C_p

   (e) $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \alpha_t \mathbf{e}_{j_t}$, where $\mathbf{e}_{j_t}$ is 1 in position j_t and 0 elsewhere.

4. **Output:** $\boldsymbol{\lambda}_{t_{max}} / ||\boldsymbol{\lambda}_{t_{max}}||_1$

Figure 1: Pseudocode for coordinate descent RankBoost.

After we make the choice of j_t, then we can plug back into the formula for α_t, yielding $\alpha_t = \frac{1}{2} \ln \frac{d_{+j_t}}{d_{-j_t}}$. We have finished re-deriving RankBoost. As we mentioned before, the plain coordinate descent algorithm has more natural weak learning associated with it, since the weak ranker chosen tries to find the most accurate weak ranker with respect to the weighted crucial pairs; in other words, we argue (3) is a more natural weak learner than (5).

Note that for AdaBoost's objective function, choosing the weak classifier with the steepest slope (plain coordinate descent) yields the same as choosing the weak classifier with the largest decrease in the loss function: both yield AdaBoost.[2]

2. For AdaBoost, entries of the matrix $\mathbf{M}$ are $M_{ij}^{Ada} := y_i h_j(\mathbf{x}_i) \in \{-1,1\}$ since hypotheses are assumed to be $\{-1,1\}$ valued for AdaBoost. Thus $d_{0j} = 0$, and from plain coordinate descent: $j_t = \operatorname{argmax}_j d_{+j} - d_{-j} = \operatorname{argmax}_j 2d_{+j} - 1$, that is, $j_t = \operatorname{argmax}_j d_{+j}$. On the other hand, for the choice of weak classifier with the greatest decreases in the loss (same calculation as above):

$$\begin{aligned} j_t &= \operatorname{argmin}_j 2(d_{+j} d_{-j})^{1/2}, \text{ that is,} \\ j_t &= \operatorname{argmin}_j d_{+j}(1 - d_{+j}) = \operatorname{argmax}_j d_{+j}^2 - d_{+j}, \end{aligned}$$

and since $d_{+j} > 1/2$, the function $d_{+j}^2 - d_{+j}$ is monotonically increasing in d_{+j}, so $j_t = \operatorname{argmax}_j d_{+j}$. Thus, whether or not AdaBoost chooses its weak classifier with knowledge of the step size, it would choose the same weak classifier anyway.

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. In fact, it is possible to minimize $\tilde{F}$ with a positive margin that is arbitrarily small, relative to the optimal.[3] Exactly the same problem occurs for AdaBoost. It has been proven (Rudin et al., 2004) that it is possible for AdaBoost not to converge to a maximum margin solution, nor even to make progress towards increasing the margin at every iteration. Thus, since the calculations are identical for RankBoost, there are certain cases in which we can expect RankBoost not to converge to a maximum margin solution.

Theorem 6 *(RankBoost does not always converge to a maximum margin solution) There exist matrices $\boldsymbol{M}$ for which RankBoost converges to a margin that is strictly less than the maximum margin.*

Proof Since RankBoost and AdaBoost differ only in their definitions of the matrix $\mathbf{M}$, they possess exactly the same convergence properties for the same choice of $\mathbf{M}$. There is an 8×8 matrix $\mathbf{M}$ in Rudin et al. (2004) for which AdaBoost converges to a margin value of $1/3$, when the maximum margin is $3/8$. Thus, the same convergence property applies for RankBoost. It is rare in the separable case to be able to solve for the asymptotic margin that AdaBoost or RankBoost converges to; for this 8×8 example, AdaBoost's weight vectors exhibit cyclic behavior, which allowed convergence of the margin to be completely determined. ■

A more complete characterization of AdaBoost's convergence with respect to the margin (and thus RankBoost's convergence) can be found in Rudin et al. (2007).

In earlier work, we have introduced a smooth margin function, which one can maximize in order to achieve a maximum margin solution for the classification problem (Rudin et al., 2007). A coordinate ascent algorithm on this function 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 extension of the convergence proofs for this algorithm is nontrivial; our main contribution in this section is a condition under which the algorithm makes progress.

The smooth ranking function $\tilde{G}$ is defined as follows:

$$\tilde{G}(\boldsymbol{\lambda}) := \frac{-\ln \tilde{F}(\boldsymbol{\lambda})}{||\boldsymbol{\lambda}||_1}.$$

It is not hard to show (see Rudin et al., 2007) that:

$$\tilde{G}(\boldsymbol{\lambda}) < \mu(\boldsymbol{\lambda}) \leq \rho, \tag{6}$$

where the margin can be written in this notation as:

$$\mu(\boldsymbol{\lambda}) = \min_i \frac{(\mathbf{M}\boldsymbol{\lambda})_i}{\|\boldsymbol{\lambda}\|_1}$$

3. One can see this by considering any vector $\boldsymbol{\lambda}$ such that $(\mathbf{M}\boldsymbol{\lambda})_{ik}$ is positive for all crucial pairs ik. That is, we choose any $\boldsymbol{\lambda}$ that yields a positive margin. We can make the value of $\tilde{F}$ arbitrarily small by multiplying $\boldsymbol{\lambda}$ by a large positive constant; this will not affect the value of the margin because the margin is $\min_{ik \in \mathcal{C}_p} (\mathbf{M}\boldsymbol{\lambda})_{ik} / ||\boldsymbol{\lambda}||_1$, and the large constant will cancel. In this way, the objective can be arbitrarily small, while the margin is certainly not maximized. Thus, coordinate descent on $\tilde{F}$ does not necessarily have anything to do with maximizing the margin.

and the best possible margin is:

$$\rho = \min_{\{\mathbf{d}:\sum_{ik} d_{ik}=1, d_{ik}\geq 0\}} \max_j (\mathbf{d}^T \mathbf{M})_j = \max_{\{\bar{\boldsymbol{\lambda}}:\sum_j \bar{\lambda}_j=1, \bar{\lambda}_j\geq 0\}} \min_i (\mathbf{M}\bar{\boldsymbol{\lambda}})_i.$$

In other words, the smooth ranking margin is always less than the true margin, although the two quantities become closer as $||\boldsymbol{\lambda}||_1$ increases. The true margin is no greater than ρ, the min-max value of the game defined by $\mathbf{M}$ (see Freund and Schapire, 1999).

We now define the smooth margin ranking algorithm, which is approximately coordinate ascent on $\tilde{G}$. As usual, the input to the algorithm is matrix $\mathbf{M}$, determined from the training data. Also, we will only define this algorithm when $\tilde{G}(\boldsymbol{\lambda})$ is positive, so that we only use it 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. At iteration t, we have calculated $\boldsymbol{\lambda}_t$, at which point the following quantities can be calculated:

$$\begin{aligned} g_t &:= \tilde{G}(\boldsymbol{\lambda}_t) \\ \text{weights on crucial pairs } d_{t,ik} &:= e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}}/\tilde{F}(\boldsymbol{\lambda}_t) \\ \text{direction } j_t &= \operatorname*{argmax}_j (\mathbf{d}_t^T \mathbf{M})_j \\ \text{edge } r_t &:= (\mathbf{d}_t^T \mathbf{M})_{j_t}. \end{aligned}$$

The choice of j_t is the same as for coordinate descent RankBoost (also see Rudin et al., 2007). The step size α_t is chosen to obey Equation (12) below, but we need a few more definitions before we state its value, so we do not define it yet; we will first define recursive equations for $\tilde{F}$ and $\tilde{G}$. We also have $s_t = ||\boldsymbol{\lambda}_t||_1$ and $s_{t+1} = s_t + \alpha_t$, and $g_{t+1} = \tilde{G}(\boldsymbol{\lambda}_t + \alpha_t \mathbf{e}_{j_t})$, where α_t has not yet been defined.

As before, $I_{t+} := \{i,k | M_{ikj_t} = 1, \pi(\mathbf{x}_i, \mathbf{x}_k) = 1\}$, $I_{t-} := \{i,k | M_{ikj_t} = -1, \pi(\mathbf{x}_i, \mathbf{x}_k) = 1\}$, and now, $I_{t0} := \{i,k | M_{ikj_t} = 0, \pi(\mathbf{x}_i, \mathbf{x}_k) = 1\}$. Also $d_{t+} := \sum_{I_{t+}} d_{t,ik}$, $d_{-} := \sum_{I_{t-}} d_{t,ik}$, and $d_{t0} := \sum_{I_{t0}} d_{t,ik}$. Thus, by definition, we have $d_{t+} + d_{t-} + d_{t0} = 1$. Now, r_t can be written $r_t = d_{t+} - d_{t-}$. Define the factor

$$\tau_t := d_{t+} e^{-\alpha_t} + d_{t-} e^{\alpha_t} + d_{t0}, \tag{7}$$

and define its "derivative":

$$\tau_t' := \frac{\partial \tau_t (d_{t+} e^{-\alpha} + d_{t-} e^{\alpha} + d_{t0})}{\partial \alpha}\Big|_{\alpha=\alpha_t} = -d_{t+} e^{-\alpha_t} + d_{t-} e^{\alpha_t}. \tag{8}$$

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

$$\begin{aligned} \tilde{F}(\boldsymbol{\lambda}_t + \alpha \mathbf{e}_{j_t}) &= \sum_{\{i,k|\pi(\mathbf{x}_i,\mathbf{x}_k)=1\}} e^{(-\mathbf{M}\boldsymbol{\lambda}_t)_{ik}} e^{-M_{ikj_t}\alpha} \\ &= \tilde{F}(\boldsymbol{\lambda}_t)(d_{t+} e^{-\alpha} + d_{t-} e^{\alpha} + d_{t0}). \end{aligned}$$

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 $\tilde{G}$.

$$\begin{aligned}\tilde{G}(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t}) &= \frac{-\ln(\tilde{F}(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t}))}{s_t+\alpha} = \frac{-\ln(\tilde{F}(\boldsymbol{\lambda}_t))-\ln\left(d_{t+}e^{-\alpha}+d_{t-}e^{\alpha}+d_{t0}\right)}{s_t+\alpha} \\ &= g_t\frac{s_t}{s_t+\alpha}-\frac{\ln\left(d_{t+}e^{-\alpha}+d_{t-}e^{\alpha}+d_{t0}\right)}{s_t+\alpha}.\end{aligned}$$

For our algorithm, we set $\alpha=\alpha_t$ in the above expression and use the notation defined earlier:

$$\begin{aligned}g_{t+1} &= g_t\frac{s_t}{s_t+\alpha_t}-\frac{\ln\tau_t}{s_t+\alpha_t} \\ g_{t+1}-g_t &= \frac{g_ts_t-g_ts_t-g_t\alpha_t}{s_t+\alpha_t}-\frac{\ln\tau_t}{s_t+\alpha_t} = -\frac{1}{s_{t+1}}\left[g_t\alpha_t+\ln\tau_t\right]. \qquad (9)\end{aligned}$$

Now we have gathered enough notation to write the equation for α_t for smooth margin ranking. For plain coordinate ascent, the update α^* solves:

$$\begin{aligned}0 &= \frac{\partial\tilde{G}(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t})}{\partial\alpha}\Big|_{\alpha=\alpha^*} = \frac{\partial}{\partial\alpha}\left[\frac{-\ln\tilde{F}(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t})}{s_t+\alpha}\right]\Big|_{\alpha=\alpha^*} \\ &= \frac{1}{s_t+\alpha^*}\left[-\left[\frac{-\ln\tilde{F}(\boldsymbol{\lambda}_t+\alpha^*\mathbf{e}_{j_t})}{s_t+\alpha^*}\right]+\left[\frac{-\partial\tilde{F}(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t})/\partial\alpha\Big|_{\alpha=\alpha^*}}{\tilde{F}(\boldsymbol{\lambda}_t+\alpha^*\mathbf{e}_{j_t})}\right]\right] \\ &= \frac{1}{s_t+\alpha^*}\left[-\tilde{G}(\boldsymbol{\lambda}_t+\alpha^*\mathbf{e}_{j_t})+\left[\frac{-\partial\tilde{F}(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t})/\partial\alpha\Big|_{\alpha=\alpha^*}}{\tilde{F}(\boldsymbol{\lambda}_t+\alpha^*\mathbf{e}_{j_t})}\right]\right]. \qquad (10)\end{aligned}$$

We could solve this equation numerically for α^* to get a smooth margin coordinate ascent algorithm; however, we avoid this line search for α^* in smooth margin ranking. We will do an approximation that allows us to solve for α^* directly so that the algorithm is just as easy to implement as RankBoost. To get the update rule for smooth margin ranking, we set α_t to solve:

$$\begin{aligned}0 &= \frac{1}{s_t+\alpha_t}\left[-\tilde{G}(\boldsymbol{\lambda}_t)+\left[\frac{-\partial\tilde{F}(\boldsymbol{\lambda}_t+\alpha\mathbf{e}_{j_t})/\partial\alpha\Big|_{\alpha=\alpha_t}}{\tilde{F}(\boldsymbol{\lambda}_t+\alpha_t\mathbf{e}_{j_t})}\right]\right] \\ &= \frac{1}{s_t+\alpha_t}\left(-g_t+\frac{-\tau_t'\tilde{F}(\boldsymbol{\lambda}_t)}{\tau_t\tilde{F}(\boldsymbol{\lambda}_t)}\right) \\ g_t\tau_t &= -\tau_t'. \qquad (11)\end{aligned}$$

This expression can be solved analytically for α_t, but we avoid using the exact expression in our calculations whenever possible, since the solution is not that easy to work with in our analysis:

$$\alpha_t=\ln\left[\frac{-g_td_{t0}+\sqrt{g_t^2d_{t0}^2+(1+g_t)(1-g_t)4d_{t+}d_{t-}}}{(1+g_t)2d_{t-}}\right]. \qquad (12)$$

We are done defining the algorithm and in the process we have derived some useful recursive relationships. In summary:

Smooth margin ranking is the same as described in Figure 1, except that (3c) is replaced by (12), where $d_{t0} = 1 - d_{t+} - d_{t-}$ *and* $g_t = G(\boldsymbol{\lambda}_t)$.

Binary weak rankers were required to obtain an analytical solution for α_t, but if one is willing to perform a 1-dimensional linesearch (10) at each iteration, real-valued features can just as easily be used.

Now we move onto the convergence proofs, which were loosely inspired by the analysis of Zhang and Yu (2005). The following theorem gives conditions when the algorithm makes significant progress towards increasing the value of $\tilde{G}$ at iteration t. An analogous statement was an essential tool for proving convergence properties of approximate coordinate ascent boosting (Rudin et al., 2007), although the proof of the following theorem is significantly more difficult since we could not use the hyperbolic trigonometric tricks from prior work. As usual, the weak learning algorithm must always achieve an edge r_t of at least ρ for the calculation to hold, where recall $r_t = (\mathbf{d}_t^T \mathbf{M})_{j_t} = d_{t+} - d_{t-}$. At every iteration, there is always a weak ranker which achieves edge at least ρ, so this requirement is always met in the "optimal case," where we choose the best possible weak ranker at every iteration (i.e., the argmax over j). There is one more condition in order for the algorithm to make progress, namely that most of the weight should indicate the strength of the weak ranker, which implies that d_{t0} cannot take too much of the weight. Specifically, $d_{t0} < \frac{2}{3}(1-r_t)(1-r_t^2)$, which is derived from a bound on the second derivative of the step size.

Theorem 7 *(Progress according to the smooth margin) For* $0 \le g_t < r_t < 1$ *and* $0 \le d_{t0} < \frac{2}{3}(1-r_t)(1-r_t^2)$ *the algorithm makes progress at iteration* t*:*

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

The proof of this theorem is in Section 7. This theorem tells us that the value of the smooth ranking margin increases significantly when the condition on d_0 holds. This theorem is the main step in proving convergence theorems, for example:

Theorem 8 *(Convergence for smooth margin ranking) If* $d_{t0} < \frac{2}{3}(1-r_t)(1-r_t^2)$ *for all* t*, the smooth margin ranking algorithm converges to a maximum margin solution, that is,* $\lim_{t\to\infty} g_t = \rho$*. Thus the limiting margin is* ρ*, that is,* $\lim_{t\to\infty} \mu(\boldsymbol{\lambda}_t) = \rho$.

Besides Theorem 7, the only other key step in the proof of Theorem 8 is the following lemma, proved in Section 7:

Lemma 9 *(Step-size does not increase too quickly for smooth margin ranking)*

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

From here, the proof of the convergence theorem is not difficult. The two conditions found in Theorem 7 and Lemma 9 are identical to those of Lemma 5.1 and Lemma 5.2 of Rudin et al. (2007). These are the only two ingredients necessary to prove asymptotic convergence using the proof outline of Theorem 5.1 of Rudin et al. (2007); an adaptation of this proof suffices to show Theorem 8, which we now outline.

Proof *(of Theorem 8)* The values of g_t constitute a nondecreasing sequence which is uniformly bounded by 1. Thus, a limit g_∞ must exist, $g_\infty := \lim_{t\to\infty} g_t$. By (6), we know that $g_t \le \rho$ for all

t. Thus, $g_\infty \leq \rho$. Let us suppose that $g_\infty < \rho$, so that $\rho - g_\infty \neq 0$. This assumption, together with Theorem 7 and Lemma 9 can be used in the same way as in Rudin et al. (2007) to show that $\sum_t \alpha_t$ is finite, implying that:

$$\lim_{t\to\infty} \alpha_t = 0.$$

Using this fact along with (11), we find:

$$\begin{aligned} g_\infty &= \lim_{t\to\infty} g_t = \liminf_{t\to\infty} g_t = \liminf_{t\to\infty} \frac{-\tau_t'}{\tau_t} = \liminf_{t\to\infty} \frac{-(-d_{t+}e^{-\alpha_t} + d_{t-}e^{\alpha_t})}{d_{t+}e^{-\alpha_t} + d_{t-}e^{\alpha_t} + d_{t0}} \\ &= \liminf_{t\to\infty} r_t \geq \rho. \end{aligned}$$

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 Ranking Problem

In this section, we present an equivalence between AdaBoost and RankBoost in terms of their behavior on the training set. Namely, we show that under very natural conditions, AdaBoost asymptotically produces an area under the ROC curve value that is equally as good as RankBoost's. Conversely, RankBoost (but with a change in the intercept), produces a classification that is equally as good as AdaBoost's. Note that this result is designed for the non-separable case; it holds in the separable case, but the result is trivial since the area under the curve is exactly one. Also, let us be clear that the result is a theoretical proof based on the optimization of the training set only. It is not an experimental result, nor is it a probabilistic guarantee about performance on a test set (such as Theorem 2).

In the bipartite ranking problem, the focus of this section, recall that every training instance falls into one of two categories, the positive class Y_+ and the negative class Y_-. We will take $\pi(\mathbf{x}_i, \mathbf{x}_k) = 1$ for each pair $i \in Y_+$ and $k \in Y_-$ so that crucial pairs exist between elements of the positive class and elements of the negative class. Define $y_i = +1$ when $i \in Y_+$, and $y_i = -1$ otherwise. The AUC (area under the Receiver Operator Characteristic curve) is equivalent to the Mann-Whitney U statistic, and it is closely related to the fraction of misranks. Specifically,

$$1 - \mathrm{AUC}(\boldsymbol{\lambda}) = \frac{\sum_{i\in Y_+}\sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda})_{ik}\leq 0]}}{|Y_+||Y_-|} = \text{fraction of misranks.}$$

In the bipartite ranking problem, the function $\tilde{F}$ becomes an exponentiated version of the AUC, that is, since $\mathbf{1}_{[x\leq 0]} \leq e^{-x}$, we have:

$$|Y_+||Y_-|(1 - \mathrm{AUC}(\boldsymbol{\lambda})) = \sum_{i\in Y_+}\sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda})_{ik}\leq 0]} \leq \sum_{i\in Y_+}\sum_{k\in Y_-} e^{-(\mathbf{M}\boldsymbol{\lambda})_{ik}} = \tilde{F}(\boldsymbol{\lambda}). \tag{13}$$

We define the matrix $\mathbf{M}^{Ada}$, which is helpful for describing AdaBoost. $\mathbf{M}^{Ada}$ is defined elementwise by $M^{Ada}_{ij} = y_i h_j(\mathbf{x}_i)$ for $i = 1,\ldots,m$ and $j = 1,\ldots,n$. Thus, $M_{ikj} = h_j(\mathbf{x}_i) - h_j(\mathbf{x}_k) = y_i h_j(\mathbf{x}_i) + y_k h_j(\mathbf{x}_k) = M^{Ada}_{ij} + M^{Ada}_{kj}$. (To change from AdaBoost's usual $\{-1,1\}$ hypotheses to RankBoost's usual $\{0,1\}$ hypotheses, divide entries of $\mathbf{M}$ by 2.) Define the following functions:

$$F_+(\boldsymbol{\lambda}) := \sum_{i\in Y_+} e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda})_i} \text{ and } F_-(\boldsymbol{\lambda}) := \sum_{k\in Y_-} e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda})_k}.$$

The objective function for AdaBoost is $F(\boldsymbol{\lambda}) := F_+(\boldsymbol{\lambda}) + F_-(\boldsymbol{\lambda})$. The objective function for RankBoost is:

$$\begin{aligned}\tilde{F}(\boldsymbol{\lambda}) &= \sum_{i\in Y_+}\sum_{k\in Y_-}\exp\left[-\sum_j \lambda_j h_j(\mathbf{x}_i)\right]\exp\left[+\sum_j \lambda_j h_j(\mathbf{x}_k)\right] \\ &= \sum_{i\in Y_+}\sum_{k\in Y_-}\exp\left[-\sum_j \lambda_j y_i h_j(\mathbf{x}_i)\right]\exp\left[-\sum_j \lambda_j y_k h_j(\mathbf{x}_k)\right] = F_+(\boldsymbol{\lambda})F_-(\boldsymbol{\lambda}). \qquad (14)\end{aligned}$$

Thus, both objective functions involve exponents of the margins of the training instances, but with a different balance between the positive and negative instances. In both cases, the objective function favors instances to be farther away from the decision boundary—even when the instances are correctly classified and not close to the decision boundary. (This is in contrast to support vector machines which do not suffer any loss for non-support vectors. This is the main reason why an analogous result does not hold for SVMs.)

We now define a quantity called *F-skew*:

$$\text{F-skew}(\boldsymbol{\lambda}) := F_+(\boldsymbol{\lambda}) - F_-(\boldsymbol{\lambda}). \qquad (15)$$

F-skew is the exponentiated version of the "skew," which measures the imbalance between positive and negative instances. The "skew" plays an important role in the expressions of Cortes and Mohri (2004, 2005) and Agarwal et al. (2005). The F-skew measures how much greater the positive instances contribute to AdaBoost's objective than the negative instances. If the F-skew is 0, it means that the positive and negative classes are contributing equally.

The following theorem shows that whenever the F-skew vanishes, any sequence $\boldsymbol{\lambda}_t$ that optimizes AdaBoost's objective F also optimizes RankBoost's objective $\tilde{F}$, and vice versa.

Theorem 10 *(Equivalence between AdaBoost and RankBoost's objectives) Let $\{\boldsymbol{\lambda}_t\}_{t=1}^{\infty}$ be any sequence for which AdaBoost's objective is minimized,*

$$\lim_{t\to\infty} F(\boldsymbol{\lambda}_t) = \inf_{\boldsymbol{\lambda}} F(\boldsymbol{\lambda}), \qquad (16)$$

and $\lim_{t\to\infty} \text{F-skew}(\boldsymbol{\lambda}_t) = 0$. *Then RankBoost's objective is minimized,*

$$\lim_{t\to\infty} \tilde{F}(\boldsymbol{\lambda}_t) = \inf_{\boldsymbol{\lambda}} \tilde{F}(\boldsymbol{\lambda}). \qquad (17)$$

Conversely, for any sequence for which RankBoost's objective is minimized, and for which the F-skew vanishes, AdaBoost's objective is minimized as well.

The proof of the converse follows directly from

$$(F_+(\boldsymbol{\lambda}) + F_-(\boldsymbol{\lambda}))^2 - (F_+(\boldsymbol{\lambda}) - F_-(\boldsymbol{\lambda}))^2 = 4F_+(\boldsymbol{\lambda})F_-(\boldsymbol{\lambda}),$$

Equations (14) and (15), and continuity of the functions involved. The proof of the forward direction in Section 8 uses a theory of convex duality for Bregman divergences developed by Della Pietra et al. (2002) and used by Collins et al. (2002). This theory allows characterization for functions that may have minima at infinity like F and $\tilde{F}$.

Theorem 10 has very practical implications due to the following, proved in Section 8:

Corollary 11 *(AdaBoost minimizes RankBoost's objective) If the constant weak hypothesis* $h_0(\mathbf{x}) = 1$ *is included in the set of AdaBoost's weak classifiers, or equivalently, if* $\mathbf{M}^{Ada}$ *has a column* j_0 *such that* $M^{Ada}_{i,j_0} = y_i$ *for all* i*, and if the* $\{\boldsymbol{\lambda}_t\}_{t=1}^{\infty}$ *sequence obeys (16), then* $\lim_{t\to\infty} \text{F-skew}(\boldsymbol{\lambda}_t) = 0$.

This result and the previous together imply that if the constant weak hypothesis is included in the set of AdaBoost's weak classifiers, then the F-skew vanishes, and RankBoost's objective $\tilde{F}$ is minimized.

Not only does AdaBoost minimize RankBoost's exponential objective function in this case, it also achieves an equally good misranking loss. Before we state this formally as a theorem, we need to avoid a very particular nonuniqueness problem. Namely, there is some ambiguity in the definition of the ranking loss for RankBoost and AdaBoost due to the arbitrariness in the algorithms, and the discontinuity of the function $\mathbf{1}_{[z\leq 0]}$, which is used for the misranking loss $\sum_i \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda})_i \leq 0]}$. The arbitrariness in the algorithms arises from the argmax step; since argmax is a set that may contain more than one element, and since the algorithm does not specify which element in that set to choose, solutions might be different for different implementations. There are many examples where the argmax set does contain more than one element (for instance, the examples in Rudin et al., 2004). The vector $\lim_{t\to\infty} \mathbf{1}_{[\mathbf{M}\boldsymbol{\lambda}_t \leq 0]}$ may not be uniquely defined; for some i,k pair we may have $\lim_{t\to\infty} (\mathbf{M}\boldsymbol{\lambda}_t)_{ik} = 0$, and in that case, values of $\lim_{t\to\infty} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda}_t)_{ik} \leq 0]}$ may take on the values 0, 1, or the limit may not exist, depending on the algorithm. Thus, in order to write a sensible theorem, we must eliminate this pathological case. No matter which implementation we choose, this only becomes a problem if $\lim_{t\to\infty} (\mathbf{M}\boldsymbol{\lambda}_t)_{ik} = 0$, that is, there is a tie in the rankings. If there is no tie, the result is deterministic. In other words, when the pathological case is eliminated, the limiting AUC can be defined and AdaBoost asymptotically achieves the same AUC as RankBoost:

Theorem 12 *(AdaBoost and RankBoost achieve the same area under the ROC curve) Consider any two sequences* $\{\boldsymbol{\lambda}_t\}_t$ *and* $\{\boldsymbol{\lambda}'_t\}_t$ *that minimize RankBoost's objective* $\tilde{F}$*, that is,*

$$\lim_{t\to\infty} \tilde{F}(\boldsymbol{\lambda}_t) = \lim_{t\to\infty} \tilde{F}(\boldsymbol{\lambda}'_t) = \inf_{\boldsymbol{\lambda}} \tilde{F}(\boldsymbol{\lambda}).$$

Then, if each positive example has a final score distinct from each negative example, that is, $\forall ik,\ \lim_{t\to\infty} (\boldsymbol{M}\boldsymbol{\lambda}_t)_{ik} \neq 0,\ \lim_{t\to\infty} (\boldsymbol{M}\boldsymbol{\lambda}'_t)_{ik} \neq 0$*, then both sequences will asymptotically achieve the same AUC value. That is:*

$$\lim_{t\to\infty} \left[\sum_{i\in Y_+} \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda}_t)_{ik} \leq 0]} \right] = \lim_{t\to\infty} \left[\sum_{i\in Y_+} \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda}'_t)_{ik} \leq 0]} \right].$$

The proof is in Section 8. This theorem shows that, in the case where the F-skew vanishes and there are no ties, AdaBoost will generate the same area under the curve value that RankBoost does. That is, a sequence of $\boldsymbol{\lambda}'_t$'s generated by AdaBoost and a sequence of $\boldsymbol{\lambda}_t$'s generated by RankBoost will asymptotically produce the same value of the AUC.

Combining Theorem 10, Corollary 11 and Theorem 12, we can conclude the following, assuming distinct final scores: *if the constant hypothesis is included in the set of AdaBoost's weak classifiers, then AdaBoost will converge to exactly the same area under the ROC curve value as RankBoost.* Given these results, it is now understandable (but perhaps still surprising) that AdaBoost performs so well as a ranking algorithm.

This logic can be made to work in reverse, so that adding a constant hypothesis to RankBoost's output will also produce a minimizer of AdaBoost's objective. In order for this to work, we need to assign the coefficient for the constant classifier (the intercept) to force the F-skew to vanish. Changing the coefficient of the constant hypothesis does not affect RankBoost's objective, but it does affect AdaBoost's. We choose the coefficient to obey the following:

Corollary 13 *(RankBoost minimizes AdaBoost's objective) Define j_0 as the entry corresponding to the constant weak classifier. Take $\boldsymbol{\lambda}_t$ to be a minimizing sequence for RankBoost's objective, that is, $\boldsymbol{\lambda}_t$ obeys (17). Consider $\boldsymbol{\lambda}_t^{corrected}$ where:*

$$\boldsymbol{\lambda}_t^{corrected} := \boldsymbol{\lambda}_t + b_t \mathbf{e}_{j_0},$$

where $\mathbf{e}_{j_0}$ is 1 in the j_0^{th} entry corresponding to the constant weak classifier, and 0 otherwise, and where:

$$b_t = \frac{1}{2} \ln \frac{F_+(\boldsymbol{\lambda}_t)}{F_-(\boldsymbol{\lambda}_t)}.$$

Then, $\boldsymbol{\lambda}_t^{corrected}$ converges to a minimum of AdaBoost's objective, that is, $\boldsymbol{\lambda}_t^{corrected}$ obeys (16).

The proof is in Section 8. Now, we can extend to the misclassification error. The proof of the following is also in Section 8:

Theorem 14 *(AdaBoost and RankBoost achieve the same misclassification error) Consider any two sequences $\{\boldsymbol{\lambda}_t^{corrected}\}_t$ and $\{\boldsymbol{\lambda}_t'^{corrected}\}_t$, corrected as in Corollary 13, that minimize RankBoost's objective $\tilde{F}$, that is,*

$$\lim_{t\to\infty} \tilde{F}(\boldsymbol{\lambda}_t^{corrected}) = \lim_{t\to\infty} \tilde{F}(\boldsymbol{\lambda}_t'^{corrected}) = \inf_{\boldsymbol{\lambda}} \tilde{F}(\boldsymbol{\lambda}).$$

Then, if no example is on the decision boundary, that is, $\forall i,\ \lim_{t\to\infty} (\boldsymbol{M}^{Ada}\boldsymbol{\lambda}_t^{corrected})_i \neq 0$, $\forall k\ \lim_{t\to\infty} (\boldsymbol{M}^{Ada}\boldsymbol{\lambda}_t^{corrected})_k \neq 0$, and $\forall i,\ \lim_{t\to\infty} (\boldsymbol{M}^{Ada}\boldsymbol{\lambda}_t'^{corrected})_i \neq 0$, $\forall k\ \lim_{t\to\infty} (\boldsymbol{M}^{Ada}\boldsymbol{\lambda}_t'^{corrected})_k \neq 0$, then both sequences will asymptotically achieve the same misclassification loss. That is:

$$\lim_{t\to\infty} \left[\sum_{i\in Y_+} \mathbf{1}_{[(\mathbf{M}^{\mathrm{Ada}}\boldsymbol{\lambda}_t^{corrected})_i \leq 0]} + \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}^{\mathrm{Ada}}\boldsymbol{\lambda}_t^{corrected})_k \leq 0]} \right]$$
$$= \lim_{t\to\infty} \left[\sum_{i\in Y_+} \mathbf{1}_{[(\mathbf{M}^{\mathrm{Ada}}\boldsymbol{\lambda}_t'^{corrected})_i \leq 0]} + \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}^{\mathrm{Ada}}\boldsymbol{\lambda}_t'^{corrected})_k \leq 0]} \right].$$

Thus, we have shown quite a strong equivalence relationship between RankBoost and AdaBoost. Under natural conditions, AdaBoost achieves the same area under the ROC curve as RankBoost, and RankBoost can be easily made to achieve the same misclassification error as AdaBoost on the training set.

The success of an algorithm is often judged using both misclassification error and the area under the ROC curve. A practical implication of this result is that AdaBoost and RankBoost both solve the classification and ranking problems at the same time. This is true under the conditions specified, namely using a set of binary weak classifiers that includes the constant classifier, and using the

correction for RankBoost's intercept. In terms of which should be used, we have found that AdaBoost tends to converge faster for classification (and uses less memory), whereas RankBoost tends to converge faster for ranking. If the algorithm is stopped early, we suggest that if misclassification error is more important, to choose AdaBoost, and conversely, if area under the ROC curve is more important, to choose RankBoost. Asymptotically, as we have shown, they produce equally good solutions for both classification and ranking on the training set.

5.1 Connection to Multiclass/Multilabel Algorithms

The results above imply convergence properties of two algorithms for solving multiclass/ multilabel problems. Specifically, the algorithms AdaBoost.MH and AdaBoost.MR of Schapire and Singer (1999) have the same relationship to each other as AdaBoost and RankBoost.

In the multilabel setting, each training instance $\mathbf{x} \in \mathcal{X}$ may belong to multiple labels in $\mathcal{Y}$, where $\mathcal{Y}$ is a finite set of labels or classes. The total number of classes is denoted by c. Examples are ordered pairs $(\mathbf{x}, Y)$, $Y \subset \mathcal{Y}$. We use the reduction of Schapire and Singer (1999) where training example i is replaced by a set of single-labeled training examples $\{(\mathbf{x}_i, y_{i\ell})\}_{\ell=1,\ldots,c}$, where $y_{i\ell} = 1$ if $y_{i\ell} \in Y_i$ and -1 otherwise. Thus, the set of training examples are indexed by pairs i, ℓ. Within this reduction, the weak classifiers become $h_j : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.

Let us now re-index the training pairs. The training pairs i, ℓ will now be assigned a single index. Define the entries of matrix $\breve{\mathbf{M}}$ by $\breve{M}_{\breve{\imath}j} = y_{\breve{\imath}} h_j(\mathbf{x}_{\breve{\imath}}, y_{\breve{\imath}})$ for all pairs i, ℓ indexed by $\breve{\imath}$. With this notation, the objective function of AdaBoost.MH becomes:

$$F_{MH}(\boldsymbol{\lambda}) := \sum_{\breve{\imath}} \exp(-\breve{\mathbf{M}}\boldsymbol{\lambda})_{\breve{\imath}}.$$

Using similar notation, the objective function of AdaBoost.MR becomes:

$$F_{MR}(\boldsymbol{\lambda}) := \sum_{\breve{\imath} \in \{\{i,\ell\}: y_{i\ell}=1\}} \exp(-\breve{\mathbf{M}}\boldsymbol{\lambda})_{\breve{\imath}} \sum_{\breve{k} \in \{\{i,\ell\}: y_{i\ell}=-1\}} \exp(-\breve{\mathbf{M}}\boldsymbol{\lambda})_{\breve{k}}.$$

The forms of functions F_{MH} and F_{MR} are the same as those of AdaBoost and RankBoost, respectively, allowing us to directly apply all of the above results. In other words, the same equivalence relationship that we have shown for AdaBoost and RankBoost applies to AdaBoost.MH and AdaBoost.MR.

Now, we move onto the proofs.

6. Proofs from Section 3

This proof in large part follows the approach of Bartlett (1998) and Schapire et al. (1998).

For $f \in \mathcal{F}$, we will be interested in the expectation

$$\mathrm{P}_{\theta,f} := \mathbb{P}_{\overline{\mathbf{x}},\tilde{\mathbf{x}} \sim \mathcal{D}}\left[f(\overline{\mathbf{x}}) - f(\tilde{\mathbf{x}}) \le \theta \mid \pi(\overline{\mathbf{x}},\tilde{\mathbf{x}}) = 1\right] = \mathbb{E}_{\overline{\mathbf{x}},\tilde{\mathbf{x}} \sim \mathcal{D}}\left[\mathbf{1}_{[f(\overline{\mathbf{x}}) - f(\tilde{\mathbf{x}}) \le \theta]} \mid \pi(\overline{\mathbf{x}},\tilde{\mathbf{x}}) = 1\right]$$

as well as its empirical analog

$$\begin{aligned} \hat{\mathrm{P}}_{\theta,f} := \mathbb{P}_S\{\mathrm{margin}_f \le \theta\} &= \mathbb{P}_{\overline{\mathbf{x}},\tilde{\mathbf{x}} \sim S}\left[f(\overline{\mathbf{x}}) - f(\tilde{\mathbf{x}}) \le \theta \mid \pi(\overline{\mathbf{x}},\tilde{\mathbf{x}}) = 1\right] \\ &= \mathbb{E}_{\overline{\mathbf{x}},\tilde{\mathbf{x}} \sim S}\left[\mathbf{1}_{[f(\overline{\mathbf{x}}) - f(\tilde{\mathbf{x}}) \le \theta]} \mid \pi(\overline{\mathbf{x}},\tilde{\mathbf{x}}) = 1\right]. \end{aligned}$$

Note that in this notation,

$$\mathbb{P}_{\mathcal{D}}\{\text{misrank}_f\} = \mathrm{P}_{0,f}.$$

Our goal is to show that $\mathrm{P}_{0,f} \leq \hat{\mathrm{P}}_{\theta,f} + \varepsilon$ for all $f \in \mathcal{F}$ with high probability. To do so, we will first show that for every $f \in \mathcal{F}$,

$$\mathrm{P}_{0,f} - \hat{\mathrm{P}}_{\theta,f} \leq \mathrm{P}_{\theta/2,g} - \hat{\mathrm{P}}_{\theta/2,g} + \frac{\varepsilon}{2}$$

for some g in the cover $\mathcal{G}$, and then show that the difference $\mathrm{P}_{\theta/2,g} - \hat{\mathrm{P}}_{\theta/2,g}$ on the right must be small for all $g \in \mathcal{G}$, with high probability.

Lemma 15 *Let f and g be any functions in $\mathcal{F}$, and let D be any joint distribution on pairs $\bar{\mathbf{x}}, \tilde{\mathbf{x}}$. Let $0 \leq \theta_1 < \theta_2$. Then*

$$\begin{aligned}
&\mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\left[\mathbf{1}_{[f(\bar{\mathbf{x}})-f(\tilde{\mathbf{x}})\leq\theta_1]} - \mathbf{1}_{[g(\bar{\mathbf{x}})-g(\tilde{\mathbf{x}})\leq\theta_2]}\right] \\
&\quad\leq\quad \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\left\{|f(\bar{\mathbf{x}}) - g(\bar{\mathbf{x}})| \geq \frac{\theta_2-\theta_1}{2}\right\} + \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\left\{|f(\tilde{\mathbf{x}}) - g(\tilde{\mathbf{x}})| \geq \frac{\theta_2-\theta_1}{2}\right\}.
\end{aligned}$$

Proof First, note that

$$\mathbf{1}_{[y\leq\theta_1]} - \mathbf{1}_{[z\leq\theta_2]} = \begin{cases} 1 & \text{if } y \leq \theta_1 < \theta_2 < z \\ 0 & \text{otherwise} \end{cases}$$

which means that this difference can be equal to 1 only if $z-y$ is at least $\theta_2 - \theta_1$. Thus,

$$\begin{aligned}
&\mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\left[\mathbf{1}_{[f(\bar{\mathbf{x}})-f(\tilde{\mathbf{x}})\leq\theta_1]} - \mathbf{1}_{[g(\bar{\mathbf{x}})-g(\tilde{\mathbf{x}})\leq\theta_2]}\right] \\
&\quad= \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\{f(\bar{\mathbf{x}}) - f(\tilde{\mathbf{x}}) \leq \theta_1 < \theta_2 < g(\bar{\mathbf{x}}) - g(\tilde{\mathbf{x}})\} \\
&\quad\leq \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\{|(f(\bar{\mathbf{x}}) - f(\tilde{\mathbf{x}})) - (g(\bar{\mathbf{x}}) - g(\tilde{\mathbf{x}}))| \geq \theta_2 - \theta_1\} \\
&\quad\leq \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\{|f(\bar{\mathbf{x}}) - g(\bar{\mathbf{x}})| + |f(\tilde{\mathbf{x}}) - g(\tilde{\mathbf{x}})| \geq \theta_2 - \theta_1\} \\
&\quad\leq \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\left\{|f(\bar{\mathbf{x}}) - g(\bar{\mathbf{x}})| \geq \frac{\theta_2-\theta_1}{2} \vee |f(\tilde{\mathbf{x}}) - g(\tilde{\mathbf{x}})| \geq \frac{\theta_2-\theta_1}{2}\right\} \\
&\quad\leq \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\left\{|f(\bar{\mathbf{x}}) - g(\bar{\mathbf{x}})| \geq \frac{\theta_2-\theta_1}{2}\right\} + \mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim D}\left\{|f(\tilde{\mathbf{x}}) - g(\tilde{\mathbf{x}})| \geq \frac{\theta_2-\theta_1}{2}\right\}
\end{aligned}$$

by the union bound. ■

The following lemma is true for every training set S:

Lemma 16 *Let $\mathcal{G}$ be a $\theta/4$-sloppy $\varepsilon/8$-cover for $\mathcal{F}$. Then for all $f \in \mathcal{F}$, there exists $g \in \mathcal{G}$ such that*

$$P_{0,f} - \hat{P}_{\theta,f} \leq P_{\theta/2,g} - \hat{P}_{\theta/2,g} + \frac{\varepsilon}{2}.$$

Proof Let $g \in \mathcal{G}$. Lemma 15, applied to the distribution $\mathcal{D}$, conditioned on $\pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1$, implies

$$\mathrm{P}_{0,f} - \mathrm{P}_{\theta/2,g} \leq \mathbb{P}_{\mathbf{x}\sim D_1}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} + \mathbb{P}_{\mathbf{x}\sim D_2}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\}$$

where D_1 and D_2 denote the marginal distributions on $\bar{\mathbf{x}}$ and $\tilde{\mathbf{x}}$, respectively, under distribution $\bar{\mathbf{x}}, \tilde{\mathbf{x}} \sim \mathcal{D}$, conditioned on $\pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1$. In other words, for any event $\omega(\mathbf{x})$, $\mathbb{P}_{\mathbf{x}\sim D_1}\{\omega(\mathbf{x})\}$ is the same as $\mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim\mathcal{D}}\{\omega(\bar{\mathbf{x}}) \mid \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1\}$, and similarly $\mathbb{P}_{\mathbf{x}\sim D_2}\{\omega(\mathbf{x})\}$ is the same as $\mathbb{P}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim\mathcal{D}}\{\omega(\tilde{\mathbf{x}}) \mid \pi(\bar{\mathbf{x}}, \tilde{\mathbf{x}}) = 1\}$.

Likewise,

$$\hat{\mathrm{P}}_{\theta/2,g} - \hat{\mathrm{P}}_{\theta,f} \leq \mathbb{P}_{\mathbf{x}\sim S_1}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} + \mathbb{P}_{\mathbf{x}\sim S_2}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\}$$

where S_1 and S_2 are distributions defined analogously for the empirical distribution on S. Thus,

$$\begin{aligned}
\mathrm{P}_{0,f} - \mathrm{P}_{\theta/2,g} + \hat{\mathrm{P}}_{\theta/2,g} - \hat{\mathrm{P}}_{\theta,f} &\leq \mathbb{P}_{\mathbf{x}\sim D_1}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} + \mathbb{P}_{\mathbf{x}\sim D_2}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} \\
&\quad + \mathbb{P}_{\mathbf{x}\sim S_1}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} + \mathbb{P}_{\mathbf{x}\sim S_2}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} \\
&= 4\,\mathbb{P}_{\mathbf{x}\sim D^*}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} \qquad (18)
\end{aligned}$$

where D^* is the (uniform) mixture of the four distributions D_1, D_2, S_1 and S_2. Constructing D^* in this way allows us to find a g that is close to f for all four terms simultaneously, which is needed for the next step. Since $\mathcal{G}$ is a $\theta/4$-sloppy $\varepsilon/8$-cover, we can now choose g to be a function in the cover $\mathcal{G}$ such that

$$\mathbb{P}_{\mathbf{x}\sim D^*}\left\{|f(\mathbf{x}) - g(\mathbf{x})| \geq \frac{\theta}{4}\right\} \leq \frac{\varepsilon}{8}$$

which, plugging in to equation (18), proves the lemma. ■

In the proof of the theorem, we will use the g's to act as representatives (for slightly different events), so we must show that we do not lose too much by doing this.

Lemma 17 *Let $\mathcal{G}$ be a $\theta/4$-sloppy $\varepsilon/8$-cover for $\mathcal{F}$. Then*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left\{\exists f \in \mathcal{F} : P_{0,f} - \hat{P}_{\theta,f} \geq \varepsilon\right\} \leq \mathbb{P}_{S\sim\mathcal{D}^m}\left\{\exists g \in \mathcal{G} : P_{\theta/2,g} - \hat{P}_{\theta/2,g} \geq \frac{\varepsilon}{2}\right\}.$$

Proof By Lemma 16, for every training set S, for any $f \in \mathcal{F}$, there exists some $g \in \mathcal{G}$ such that

$$\mathrm{P}_{0,f} - \hat{\mathrm{P}}_{\theta,f} \leq \hat{\mathrm{P}}_{\theta/2,g} - \mathrm{P}_{\theta/2,g} + \frac{\varepsilon}{2}.$$

Thus, if there exists an $f \in \mathcal{F}$ such that $\mathrm{P}_{0,f} - \hat{\mathrm{P}}_{\theta,f} \geq \varepsilon$, then there exists a $g \in \mathcal{G}$ such that $\mathrm{P}_{\theta/2,g} - \hat{\mathrm{P}}_{\theta/2,g} \geq \frac{\varepsilon}{2}$. The statement of the lemma follows directly. ■

Now we incorporate the fact that the training set is chosen randomly. We will use a generalization of Hoeffding's inequality due to McDiarmid, as follows:

Theorem 18 ***(McDiarmid's Inequality McDiarmid 1989)*** *Let $X_1, X_2, \ldots X_m$ be independent random variables under distribution D. Let $f(\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be any real-valued function such that for all $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m; \mathbf{x}_i'$,*

$$|f(\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_m) - f(\mathbf{x}_1, \ldots, \mathbf{x}_i', \ldots \mathbf{x}_m)| \leq c_i.$$

Then for any $\varepsilon > 0$,

$$\begin{aligned}
\mathbb{P}_{X_1,X_2,\ldots,X_m\sim D}\left\{f(X_1, X_2, \ldots, X_m) - \mathbb{E}[f(X_1, X_2, \ldots, X_m)] \geq \varepsilon\right\} &\leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^m c_i^2}\right), \\
\mathbb{P}_{X_1,X_2,\ldots,X_m\sim D}\left\{\mathbb{E}[f(X_1, X_2, \ldots, X_m)] - f(X_1, X_2, \ldots, X_m) \geq \varepsilon\right\} &\leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^m c_i^2}\right).
\end{aligned}$$

Lemma 19 *For any $f \in \mathcal{F}$,*

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

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

$$\begin{aligned}
\mathrm{top}_{\mathcal{D}} &:= \mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim\mathcal{D}}[\mathbf{1}_{[f(\bar{\mathbf{x}})-f(\tilde{\mathbf{x}})\leq\theta]}\pi(\bar{\mathbf{x}},\tilde{\mathbf{x}})] \\
\mathrm{top}_S &:= \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{k=1}^{m}\mathbf{1}_{[f(\mathbf{x}_i)-f(\mathbf{x}_k)\leq\theta]}\pi(\mathbf{x}_i,\mathbf{x}_k) \\
\mathrm{bot}_{\mathcal{D}} := E &:= \mathbb{E}_{\bar{\mathbf{x}},\tilde{\mathbf{x}}\sim\mathcal{D}}[\pi(\bar{\mathbf{x}},\tilde{\mathbf{x}})] \\
\mathrm{bot}_S &:= \frac{1}{m(m-1)}\sum_{i=1}^{m}\sum_{k=1}^{m}\pi(\mathbf{x}_i,\mathbf{x}_k).
\end{aligned}$$

Since diagonal terms have $\pi(\mathbf{x}_i,\mathbf{x}_i)$ which is always 0, $\mathrm{top}_{\mathcal{D}} = \mathbb{E}_{S\sim\mathcal{D}^m}[\mathrm{top}_S]$ and similarly, $\mathrm{bot}_{\mathcal{D}} = \mathbb{E}_{S\sim\mathcal{D}^m}[\mathrm{bot}_S]$. Thus, we can bound the difference between top_S and $\mathrm{top}_{\mathcal{D}}$ using large deviation bounds, and similarly for the difference between bot_S and $\mathrm{bot}_{\mathcal{D}}$. We choose McDiarmid's Inequality to perform this task. It is not difficult to show using the rules of π that the largest possible change in top_S due to the replacement of one example is $1/m$. Similarly the largest possible change in bot_S is $1/m$. Thus, McDiarmid's inequality applied to top_S and bot_S implies that for every $\varepsilon_1 > 0$:

$$\begin{aligned}
\mathbb{P}_{S\sim\mathcal{D}^m}\{\mathrm{top}_{\mathcal{D}} - \mathrm{top}_S \geq \varepsilon_1\} &\leq \exp[-2\varepsilon_1^2 m] \\
\mathbb{P}_{S\sim\mathcal{D}^m}\{\mathrm{bot}_S - \mathrm{bot}_{\mathcal{D}} \geq \varepsilon_1\} &\leq \exp[-2\varepsilon_1^2 m].
\end{aligned}$$

Here, we use ε_1 to avoid confusion with the ε in the statement of the lemma; we will specify ε_1 in terms of ε later, but since the equations are true for any $\varepsilon_1 > 0$, we work with general ε_1 for now. Consider the following event:

$$\mathrm{top}_{\mathcal{D}} - \mathrm{top}_S < \varepsilon_1 \quad \text{and} \quad \mathrm{bot}_S - \mathrm{bot}_{\mathcal{D}} < \varepsilon_1.$$

By the union bound, this event is true with probability at least $1 - 2\exp[-2\varepsilon_1^2 m]$. When the event is true, we can rearrange the equations to be a bound on

$$\frac{\mathrm{top}_{\mathcal{D}}}{\mathrm{bot}_{\mathcal{D}}} - \frac{\mathrm{top}_S}{\mathrm{bot}_S}.$$

We do this as follows:

$$\frac{\mathrm{top}_{\mathcal{D}}}{\mathrm{bot}_{\mathcal{D}}} - \frac{\mathrm{top}_S}{\mathrm{bot}_S} < \frac{\mathrm{top}_{\mathcal{D}}}{\mathrm{bot}_{\mathcal{D}}} - \frac{\mathrm{top}_{\mathcal{D}} - \varepsilon_1}{\mathrm{bot}_{\mathcal{D}} + \varepsilon_1}. \tag{19}$$

If we now choose:

$$\varepsilon_1 = \frac{\varepsilon \mathrm{bot}_{\mathcal{D}}}{2 - \varepsilon + 2\frac{\mathrm{top}_{\mathcal{D}}}{\mathrm{bot}_{\mathcal{D}}}} \geq \frac{\varepsilon \mathrm{bot}_{\mathcal{D}}}{4} =: \frac{\varepsilon E}{4}$$

then the right hand side of (19) is equal to $\varepsilon/2$. Here, we have used $E := \mathrm{bot}_{\mathcal{D}}$, and by the definition of $\mathrm{top}_{\mathcal{D}}$ and $\mathrm{bot}_{\mathcal{D}}$, we always have $\mathrm{top}_{\mathcal{D}} \leq \mathrm{bot}_{\mathcal{D}}$. We directly have:

$$1 - 2\exp[-2\varepsilon_1^2 m] \geq 1 - 2\exp\left(-2m\left[\frac{\varepsilon E}{4}\right]^2\right).$$

Therefore, from our earlier application of McDiarmid, we find that with probability at least

$$1 - 2\exp\left[-\frac{m(\varepsilon E)^2}{8}\right]$$

the following holds:

$$\mathrm{P}_{\theta,f} - \hat{\mathrm{P}}_{\theta,f} = \frac{\mathrm{top}_{\mathcal{D}}}{\mathrm{bot}_{\mathcal{D}}} - \frac{\mathrm{top}_S}{\mathrm{bot}_S} < \varepsilon/2.$$

■

As mentioned earlier, we could have equally well have written the lemma in terms of the empirical quantity $\mathrm{bot}_{\mathcal{D}}$ rather than in terms of E. We have made this decision because the bound is useful for allowing us to determine which quantities are important to maximize in our algorithms; we cannot maximize $\mathrm{bot}_{\mathcal{D}}$ in practice because we are choosing m random instances from $\mathcal{D}$, thus we have no influence at all over the value of $\mathrm{bot}_{\mathcal{D}}$ in practice. Either way, the bound tells us that the margin should be an important quantity to consider in the design of algorithms.

Also, note that this proof implicitly used our simplifying assumption that the truth function π is deterministic. In the more general case, where the value $\pi(\mathbf{x}_i, \mathbf{x}_k)$ of each training pair $\mathbf{x}_i, \mathbf{x}_k$ is determined probabilistically, an alternative proof giving the same result can be given using Azuma's lemma.

Proof *(of Theorem 2)* Let $\mathcal{G}$ be a $\theta/4$-sloppy $\varepsilon/8$-cover of $\mathcal{F}$ of minimum size. Applying Lemma 17, the union bound, and then Lemma 19 for $\theta/2$, we find:

$$\begin{aligned}
\mathbb{P}_{S\sim\mathcal{D}^m}\left\{\exists f \in \mathcal{F} : \mathrm{P}_{0,f} - \hat{\mathrm{P}}_{\theta,f} \geq \varepsilon\right\} &\leq \mathbb{P}_{S\sim\mathcal{D}^m}\left\{\exists g \in \mathcal{G} : \mathrm{P}_{\theta/2,g} - \hat{\mathrm{P}}_{\theta/2,g} \geq \frac{\varepsilon}{2}\right\}. \\
&\leq \sum_{g\in\mathcal{G}} \mathbb{P}_{S\sim\mathcal{D}^m}\left\{\mathrm{P}_{\theta/2,g} - \hat{\mathrm{P}}_{\theta/2,g} \geq \frac{\varepsilon}{2}\right\} \\
&\leq \sum_{g\in\mathcal{G}} 2\exp\left(-\frac{m(\varepsilon E)^2}{8}\right) \\
&= \mathcal{N}\left(\mathcal{F}, \frac{\theta}{4}, \frac{\varepsilon}{8}\right) 2\exp\left[-\frac{m(\varepsilon E)^2}{8}\right].
\end{aligned}$$

Now we put everything together. With probability at least

$$1 - \mathcal{N}\left(\mathcal{F}, \frac{\theta}{4}, \frac{\varepsilon}{8}\right) 2\exp\left[-\frac{m(\varepsilon E)^2}{8}\right],$$

we have

$$\mathbb{P}_D\{\mathrm{misrank}_f\} = \mathrm{P}_{0,f} \leq \hat{\mathrm{P}}_{\theta,f} + \varepsilon = \mathbb{P}_S\{\mathrm{margin}_f \leq \theta\} + \varepsilon.$$

Thus, the theorem has been proved. ■

We now provide a proof for Lemma 3, which gives an estimate of the covering number for convex combinations of dictionary elements.

Proof *(of Lemma 3)* We are trying to estimate the covering number for $\mathcal{F}$, where

$$\mathcal{F} = \left\{ f : f = \sum_j \lambda_j h_j, \sum_j \lambda_j = 1, \forall j\, \lambda_j \geq 0, h_j : \mathcal{X} \to \{0,1\}, h_j \in \mathcal{H} \right\}.$$

Consider the following set $\mathcal{G}_N$ of all g that can be written as a simple average of N elements of $\mathcal{H}$:

$$\mathcal{G}_N = \left\{ \frac{1}{N}(g_1 + \cdots + g_N) : g_1, \ldots, g_N \in \mathcal{H} \right\}.$$

We claim that $\mathcal{G}_N$ is a θ-sloppy ε-cover when

$$N \geq \frac{\ln(2/\varepsilon)}{2\theta^2}. \tag{20}$$

To show this, let f be any function $\mathcal{F}$, and let D be any distribution. We know that $f = \sum_j \lambda_j h_j$ for some λ_j's as above, where the number of terms in the sum can be much more than N. Let us pick N dictionary elements $g_1, \ldots, g_N$ from $\mathcal{H}$ by choosing them randomly and independently with replacement according to the distribution imposed by the coefficients $\boldsymbol{\lambda}$. That is, each g_i is selected to be h_j with probability equal to λ_j. Thus, if λ_j is large, it is more likely that h_j will be chosen as one of the N chosen elements $g_1, \ldots, g_N$. Construct g as the average of those N elements.

Let $\mathbf{x} \in \mathcal{X}$ be any fixed element. Then $g(\mathbf{x})$ is an average of N Bernoulli random variables, namely, $g_1(\mathbf{x}), \ldots, g_N(\mathbf{x})$; by the manner in which each g_j was chosen, each of these Bernoulli random variables is 1 with probability exactly $f(\mathbf{x})$. Therefore, by Hoeffding's inequality,

$$\mathbb{P}_g \{ |g(\mathbf{x}) - f(\mathbf{x})| \geq \theta \} \leq 2e^{-2\theta^2 N}$$

where $\mathbb{P}_g\{\cdot\}$ denotes probability with respect to the random choice of g.

This holds for every $\mathbf{x}$. Now let $\mathbf{x}$ be random according to D. Then

$$\begin{aligned} \mathbb{E}_g \left[\mathbb{P}_{\mathbf{x} \sim D} \{ |f(\mathbf{x}) - g(\mathbf{x})| \geq \theta \} \right] &= \mathbb{E}_{\mathbf{x} \sim D} \left[\mathbb{P}_g \{ |f(\mathbf{x}) - g(\mathbf{x})| \geq \theta \} \right] \\ &\leq \mathbb{E}_{\mathbf{x} \sim D} \left[2e^{-2\theta^2 N} \right] = 2e^{-2\theta^2 N}. \end{aligned}$$

Thus, there exists $g \in \mathcal{G}_N$ such that

$$\mathbb{P}_{\mathbf{x} \sim D} \{ |f(\mathbf{x}) - g(\mathbf{x})| \geq \theta \} \leq 2e^{-2\theta^2 N}.$$

Hence, selecting N as in equation (20) ensures that $\mathcal{G}_N$ is a θ-sloppy ε-cover. The covering number $\mathcal{N}(\mathcal{F}, \theta, \varepsilon)$ is thus at most

$$|\mathcal{G}_N| \leq |\mathcal{H}|^N,$$

which is the bound given in the statement of the lemma. ■

7. Proofs from Section 4.2

Proof *(of Lemma 9)* There are two possibilities; either $\lim_{t \to \infty} s_t = \infty$ or $\lim_{t \to \infty} s_t < \infty$. We handle these cases separately, starting with the case $\lim_{t \to \infty} s_t = \infty$. From (9),

$$\begin{aligned} s_{t+1}(g_{t+1} - g_t) &= -g_t \alpha_t - \ln \tau_t \\ s_t(g_{t+1} - g_t) &= -g_t \alpha_t - \alpha_t(g_{t+1} - g_t) - \ln \tau_t \\ s_t(g_{t+1} - g_t) &= -\alpha_t g_{t+1} - \ln \tau_t \\ s_t(g_{t+1} - g_t) + \ln \tau_t + \alpha_t &= \alpha_t(1 - g_{t+1}) \geq \alpha_t(1 - \rho) \\ \frac{g_{t+1} - g_t}{1 - \rho} + \frac{\ln \tau_t + \alpha_t}{s_t(1 - \rho)} &\geq \frac{\alpha_t}{s_t} \geq \frac{\alpha_t}{s_{t+1}}. \end{aligned}$$

Since the g_t's constitute a nondecreasing sequence bounded by 1, $(g_{t+1} - g_t) \to 0$ as $t \to \infty$, so the first term on the left vanishes. The second term will vanish as long as we can bound $\ln \tau_t + \alpha_t$ by a constant, since by assumption, $s_t \to \infty$.

We define $g_{\tilde{1}}$ as the first positive value of $\tilde{G}(\boldsymbol{\lambda}_t)$; the value of $\tilde{G}$ only increases from this value. In order to bound $\ln \tau_t + \alpha_t$, we use Equation (11):

$$\begin{aligned} \ln \tau_t + \alpha_t &= \ln(-\tau_t') - \ln g_t + \alpha_t = \ln[d_{t+} e^{-\alpha_t} - d_{t-} e^{\alpha_t}] - \ln g_t + \alpha_t \\ &= \ln[d_{t+} - d_{t-} e^{2\alpha_t}] + \ln e^{-\alpha_t} - \ln g_t + \alpha_t \\ &\leq \ln d_{t+} - \ln g_t \leq \ln 1 - \ln g_{\tilde{1}} = -\ln g_{\tilde{1}} < \infty. \end{aligned}$$

Thus, the second term will vanish, and we now have the sequence α_t / s_{t+1} upper bounded by a vanishing sequence; thus, it too will vanish.

Now for the case where $\lim_{t\to\infty} s_t < \infty$. Consider

$$\begin{aligned} \sum_{t=\tilde{1}}^{T} \frac{\alpha_t}{s_{t+1}} &= \sum_{t=\tilde{1}}^{T} \frac{s_{t+1} - s_t}{s_{t+1}} = \sum_{t=\tilde{1}}^{T} \int_{s_t}^{s_{t+1}} \frac{1}{s_{t+1}} du \\ &\leq \sum_{t=\tilde{1}}^{T} \int_{s_t}^{s_{t+1}} \frac{1}{u} du = \int_{s_{\tilde{1}}}^{s_{T+1}} \frac{1}{u} du = \ln \frac{s_{T+1}}{s_{\tilde{1}}}. \end{aligned}$$

By our assumption that $\lim_{t\to\infty} s_t < \infty$, the above sequence is a bounded increasing sequence. Thus, $\sum_{t=\tilde{1}}^{\infty} \frac{\alpha_t}{s_{t+1}}$ converges. In particular,

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

■

Proof *(of Theorem 7)* The proof relies completely on an important calculus lemma, Lemma 20 below. Before we state the lemma, we make some definitions and derive some tools for later use.

We will be speaking only of iterations t and $t+1$, so when the iteration subscript has been eliminated, it refers to iteration t rather than iteration $t+1$. From now on, the basic independent variables will be r, g and d_0. Here, the ranges are $0 < r < 1$, $0 \leq g < r$, $0 \leq d_0 < \frac{2}{3}(1-r)(1-r^2)$. We change our notation to reinforce this: d_+ and d_- can be considered functions of the basic variables r and d_0 since $d_+ = (1 + r - d_0)/2$ and $d_- = (1 - r - d_0)/2$. Also define $\tau(r, g, d_0) := \tau_t$, $\tau'(r, g, d_0) = \tau_t'$, and $\alpha(r, g, d_0) := \alpha_t$, which are specified by (7), (8) and (11).

Define the following:

$$\begin{aligned} \Gamma(r, g, d_0) &:= \frac{-\ln \tau(r, g, d_0)}{\alpha(r, g, d_0)}. \\ \mathcal{B}(r, g, d_0) &:= \frac{\Gamma(r, g, d_0) - g}{r - g}. \end{aligned}$$

Now we state the important lemma we need for proving the theorem.

Lemma 20 *For* $0 < r < 1$, $0 \leq g < r$, $0 \leq d_0 < \frac{2}{3}(1-r)(1-r^2)$,

$$\mathcal{B}(r, g, d_0) > 1/2.$$

The proof is technical and has been placed in the Appendix. Using only this lemma, we can prove the theorem directly. Let us unravel the notation a bit. From the definition of $\Gamma(r,g,d_0)$ and Lemma 20:

$$\begin{aligned}\frac{-\ln\tau(r,g,d_0)}{\alpha(r,g,d_0)} = \Gamma(r,g,d_0) = g+(r-g)\mathcal{B}(r,g,d_0) &> \frac{r+g}{2}\\ -\ln\tau(r,g,d_0) &> \frac{(r+g)\alpha(r,g,d_0)}{2}.\end{aligned}$$

Using this relation at time t and incorporating the recursive equation, Equation (9),

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

■

We have proved the theorem, minus the proof of Lemma 20 which was the key step. Lemma 20 is a challenging calculus problem in three variables. For the sake of intuition, we plot $\mathcal{B}$ as a function of r and g for fixed $d_0 = 0.01$ in Figure 2. The result of Lemma 20 is apparent, namely that $\mathcal{B}$ is lower bounded by $1/2$.

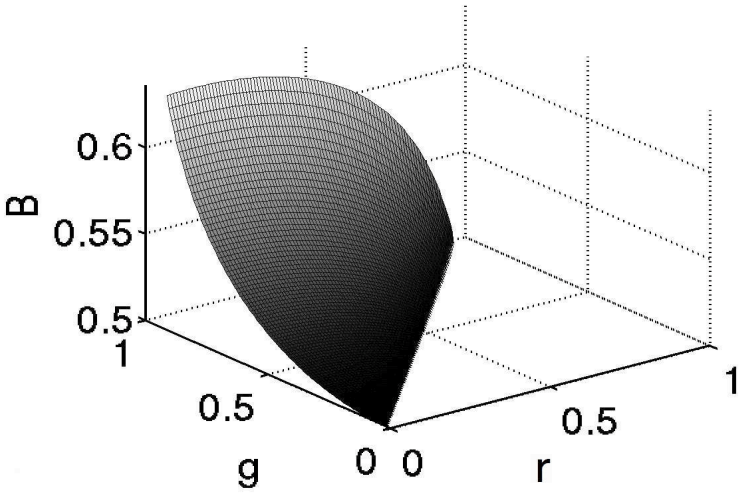

Figure 2: Surface plot of $\mathcal{B}$ as a function of r and g with $d_0 = 0.01$.

8. Proofs from Section 5

Proof *(of Theorem 10)* A proof is only necessary to handle the nonseparable case, since the statement of the theorem is trivial in the separable case. To see this, assume first that we are in the separable case, that is,

$$\lim_{t\to\infty} F_+(\boldsymbol{\lambda}_t) = \lim_{t\to\infty} F_-(\boldsymbol{\lambda}_t) = 0,$$

thus

$$\lim_{t\to\infty} \tilde{F}(\boldsymbol{\lambda}_t) = \lim_{t\to\infty} F(\boldsymbol{\lambda}_t) = 0$$

and we are done. For the rest of the proof, we handle the nonseparable case.

It is possible that the infimum of F or $\tilde{F}$ occurs at infinity, that is, F or $\tilde{F}$ may have no minimizers. Thus, it is not possible to characterize the minimizers by setting the first derivatives to zero. So, in order to more precisely describe the conditions (16) and (17), we now use a technique used by Della Pietra et al. (2002) and later used by Collins et al. (2002), in which we consider F and $\tilde{F}$ as functions of another variable, where the infimum can be achieved. Define, for a particular matrix $\bar{\mathbf{M}}$, the function

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

Define

$$\bar{\mathcal{P}} := \{\mathbf{p} | p_i \geq 0 \; \forall i, \; (\mathbf{p}^T \bar{\mathbf{M}})_j = 0 \; \forall \; j\}$$

$$\bar{\mathcal{Q}} := \{\mathbf{q} | q_i = \exp(-(\bar{\mathbf{M}}\boldsymbol{\lambda})_i) \text{ for some } \boldsymbol{\lambda}\}.$$

We may thus consider $F_{\bar{\mathbf{M}}}$ as a function of $\bar{\mathbf{q}}$, that is, $F_{\bar{\mathbf{M}}}(\bar{\mathbf{q}}) = \sum_{i=1}^{\bar{m}} \bar{q}_i$, where $\bar{\mathbf{q}} \in \bar{\mathcal{Q}}$. We know that since all $\bar{q}_i$'s are positive, the infimum of F occurs in a bounded region of $\bar{\mathbf{q}}$ space, which is just what we need.

Theorem 1 of Collins et al. (2002), which is taken directly from Della Pietra et al. (2002), implies that the following are equivalent:

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

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

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

$$\sum_{i \in Y_+} q_i^* M_{ij}^{Ada} + \sum_{k \in Y_-} q_k^* M_{kj}^{Ada} = 0 \; \forall \; j. \tag{21}$$

Define $\mathbf{q}_t$ element-wise by: $q_{t,i} := e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t)_i}$, where the $\boldsymbol{\lambda}_t$'s are a sequence that obey (16), for example, a sequence produced by AdaBoost. Thus, $\mathbf{q}_t \in \mathcal{Q}^{Ada}$ automatically. By assumption, $F(\mathbf{q}_t)$ converges to the minimum of F. Thus, since F is continuous, any limit point of the $\mathbf{q}_t$'s must minimize F as well. But because $\mathbf{q}^*$ is the unique minimizer of F, this implies that $\mathbf{q}^*$ is the one and only ℓ_p-limit point of the $\mathbf{q}_t$'s, and therefore, that the entire 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 sequence $\tilde{q}_{t,ik}$ is $\tilde{q}_{ik}^{\infty} := q_i^* q_k^*$.

What we need to show is that $\tilde{\mathbf{q}}^{\infty} = \tilde{\mathbf{q}}^*$. If we can prove this, we will have shown that $\{\boldsymbol{\lambda}_t\}_t$ converges to the minimum of RankBoost's objective function, $\tilde{F}$. We will do this by showing

that $\tilde{\mathbf{q}}^\infty \in \tilde{\mathcal{P}}$; once we accomplish this, due to the uniqueness of $\tilde{\mathbf{q}}^*$ as the intersection of $\tilde{\mathcal{P}}$ and closure$(\tilde{Q})$, we will have proved that $\tilde{\mathbf{q}}^\infty = \tilde{\mathbf{q}}^*$. So, now we proceed to show $\tilde{\mathbf{q}}^\infty \in \tilde{\mathcal{P}}$, using our assumption that the F-skew vanishes. Our assumption that the F-skew vanishes can be rewritten as:

$$\lim_{t\to\infty}\left[\sum_{i\in Y_+} q_{t,i} - \sum_{k\in Y_-} q_{t,k}\right] = 0,$$

that is, since all terms are bounded,

$$\sum_{i\in Y_+} q_i^* = \sum_{k\in Y_-} q_k^*. \tag{22}$$

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.

$$\begin{aligned}(\tilde{\mathbf{q}}^{\infty T}\mathbf{M})_j &= \sum_{i\in Y_+}\sum_{k\in Y_-} q_i^* q_k^* (M_{ij}^{Ada} + M_{kj}^{Ada}) \\ &= \left(\sum_{k\in Y_-} q_k^*\right)\left(\sum_{i\in Y_+} q_i^* M_{ij}^{Ada}\right) + \left(\sum_{i\in Y_+} q_i^*\right)\left(\sum_{k\in Y_-} q_k^* M_{kj}^{Ada}\right). \end{aligned}\tag{23}$$

Incorporating (22), which is the condition that F-skew$(\mathbf{q}^*) = 0$, (23) becomes:

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

In fact, according to (21), the bracket in this expression is zero for all j. Thus, $\tilde{\mathbf{q}}_\infty \in \tilde{\mathcal{P}}$. We have proved the forward direction of the theorem. The backwards direction, as noted earlier, follows from $(F_+ + F_-)^2 - (F_+ - F_-)^2 = 4F_+F_-$. ■

Proof *(of Corollary 11)* Recall that $\mathbf{q}^* \in \mathcal{P}^{Ada}$. Specifically writing this condition just for the constant weak classifier yields:

$$\begin{aligned} 0 &= \sum_{i\in Y_+} q_i^* M_{i0}^{Ada} + \sum_{k\in Y_-} q_k^* M_{k0}^{Ada} = \sum_{i\in Y_+} q_i^* y_i + \sum_{k\in Y_-} q_k^* y_k \\ &= \sum_{i\in Y_+} q_i^* - \sum_{k\in Y_-} q_k^* = \lim_{t\to\infty} \text{F-skew}(\boldsymbol{\lambda}_t). \end{aligned}$$

■

Proof *(of Theorem 12)* We know from the proof of Theorem 10 that since $\{\boldsymbol{\lambda}_t\}_t$ and $\{\boldsymbol{\lambda}'_t\}_t$ minimize $\tilde{F}$, we automatically have $\tilde{q}_t \to \tilde{q}^*$ and $q'_t \to \tilde{q}^*$ in ℓ_p where

$$q'_{t,ik} := e^{-(\mathbf{M}\boldsymbol{\lambda}'_t)_{ik}}.$$

Thus, we have that for all crucial pairs i,k such that $i \in Y_+$ and $k \in Y_-$:

$$\lim_{t\to\infty} e^{-(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}} = \lim_{t\to\infty} e^{-(\mathbf{M}\boldsymbol{\lambda}'_t)_{ik}} = \tilde{q}^*_{ik}.$$

For each crucial pair i,k, if $\tilde{q}^*_{ik} > 1$ then $\lim_{t\to\infty} (\mathbf{M}\boldsymbol{\lambda}_t)_{ik} < 0$, that is,

$$\lim_{t\to\infty} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}\leq 0]} = 1,$$

and conversely, if $\tilde{q}^*_{ik} < 1$ then

$$\lim_{t\to\infty} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}\leq 0]} = 0.$$

This is provided by the continuity of the function $\mathbf{1}_{[z\leq 0]}$ away from $z = 0$, and since there are no asymptotic ties in score as we have assumed, $\tilde{q}^*_{ik} \neq 1$. The same statement holds for $\boldsymbol{\lambda}'_t$. Summing over i,k pairs yields:

$$\lim_{t\to\infty} \left[\sum_{i\in Y_+} \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda}_t)_{ik}\leq 0]} \right] = \lim_{t\to\infty} \left[\sum_{i\in Y_+} \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}\boldsymbol{\lambda}'_t)_{ik}\leq 0]} \right].$$

The theorem has been proved. Note that the AUC value is obtained from this sum by the formula (13). ■

Proof *(of Corollary 13)* By Theorem 10, it is sufficient to show that the correction does not influence the value of $\tilde{F}(\boldsymbol{\lambda}_t)$ and that it makes the F-skew vanish. Consider the vector $\boldsymbol{\lambda}_t + c\mathbf{e}_{j_0}$.

$$\begin{aligned}
\tilde{F}(\boldsymbol{\lambda} + c\mathbf{e}_{j_0}) &= \sum_{i\in Y_+} \sum_{k\in Y_-} \exp\left[-\sum_j \lambda_j h_j(\mathbf{x}_i) - c \right] \exp\left[+\sum_j \lambda_j h_j(\mathbf{x}_k) + c \right] \\
&= \sum_{i\in Y_+} \sum_{k\in Y_-} \exp\left[-\sum_j \lambda_j h_j(\mathbf{x}_i) \right] \exp\left[+\sum_j \lambda_j h_j(\mathbf{x}_k) \right] = \tilde{F}(\boldsymbol{\lambda}).
\end{aligned}$$

So, changing the coefficient of the constant weak classifier will not affect the values of $\tilde{F}(\boldsymbol{\lambda})$. Now, let's compute the F-skew of the corrected sequence:

$$\begin{aligned}
\text{F-skew}(\boldsymbol{\lambda}_t^{\text{corrected}}) &= F_+(\boldsymbol{\lambda}_t + b_t\mathbf{e}_{j_0}) - F_-(\boldsymbol{\lambda}_t + b_t\mathbf{e}_{j_0}) \\
&= \sum_{i\in Y_+} e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t)_i - b_t} - \sum_{k\in Y_-} e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t)_k + b_t} \\
&= e^{-b_t}F_+(\boldsymbol{\lambda}_t) - e^{b_t}F_-(\boldsymbol{\lambda}_t) = 0
\end{aligned}$$

where this latter expression is equal to zero by our choice of b_t. Since the F-skew of the corrected sequence is always 0, the corrected sequence will minimize not only RankBoost's objective, but also AdaBoost's. ■

Proof *(of Theorem 14)* We will use a similar argument as in Theorem 12 for misclassification error rather than for ranking error. By assumption, $\boldsymbol{\lambda}_t$ is a sequence that minimizes RankBoost's objective $\tilde{F}$ and the correction forces the F-skew to be zero. Thus $\boldsymbol{\lambda}_t^{\text{corrected}}$ minimizes RankBoost's objective, and Theorem 10 implies that $\boldsymbol{\lambda}_t^{\text{corrected}}$ is also a minimizing sequence for AdaBoost's objective F. Using the same argument as in Theorem 12 substituting AdaBoost for RankBoost, we have that

$$\lim_{t\to\infty} e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t^{\text{corrected}})_i} =: q_i^*$$

exists for all i and

$$\lim_{t\to\infty} e^{-(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t^{\text{corrected}})_k} =: q_k^*$$

exists for all k. Now, we have that for each example i, if $q_i^* > 1$ then $\lim_{t\to\infty}(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t)_i < 0$, that is,

$$\lim_{t\to\infty} \mathbf{1}_{[(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t^{\text{corrected}})_i \le 0]} = 1,$$

and conversely, if $\tilde{q}_i^* < 1$ then

$$\lim_{t\to\infty} \mathbf{1}_{[(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t^{\text{corrected}})_i \le 0]} = 0.$$

The same holds for all k and for $\boldsymbol{\lambda}_t'^{\text{corrected}}$. Again, there is no asymptotic convergence to the decision boundary as we have assumed, $\tilde{q}_i^* \neq 1$, $\tilde{q}_k^* \neq 1$. The same statement holds for $\boldsymbol{\lambda}_t'^{\text{corrected}}$. Summing over i and k yields:

$$\lim_{t\to\infty}\left[\sum_{i\in Y_+} \mathbf{1}_{[(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t^{\text{corrected}})_i \le 0]} + \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t^{\text{corrected}})_k \le 0]}\right]$$
$$= \lim_{t\to\infty}\left[\sum_{i\in Y_+} \mathbf{1}_{[(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t'^{\text{corrected}})_i \le 0]} + \sum_{k\in Y_-} \mathbf{1}_{[(\mathbf{M}^{Ada}\boldsymbol{\lambda}_t'^{\text{corrected}})_k \le 0]}\right].$$

■

9. Conclusions

We have presented three main results. First, in Section 3, we presented a generalization bound for ranking. This bound incorporates a margin, allowing it to be useful in the separable case. The second main result is an algorithm, smooth margin ranking, that maximizes the ranking margin. Our third result is that under very general conditions, AdaBoost solves classification and ranking problems simultaneously, performing just as well for the ranking problem as RankBoost. Conversely, RankBoost with a change in intercept performs just as well for the classification problem as AdaBoost.

10. Open Problems and Future Work

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 have developed one such bound in Rudin (2009); In that work, we develop a specialized bound based on Theorem 2, designed to emphasize the top portion of the list.

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. In fact, it is also worth considering the empirical performance of AdaBoost to RankBoost, now that we know AdaBoost can be used for ranking.

Acknowledgments

We would like to thank the anonymous reviewers and the editor for their helpful comments; some of these comments were especially helpful in formulating Corollary 13. Thanks to Adrian Banner and Richard Sharp for their patience and assistance with earlier versions of the proof of Lemma 20. Thanks also to Corinna Cortes and Mehryar Mohri, who co-authored a preliminary conference version of this work.

This material is based upon work partially supported by the National Science Foundation under grants IIS-0325500 and CCR-0325463. CDR was supported by an NSF postdoctoral research fellowship under grant DBI-0434636 at New York University.

Appendix A. Proof of Lemma 20

We will first prove some properties of α, τ, Γ, and $\mathcal{B}$ in the following lemmas. First, we show $\alpha(r,g,d_0)$ is a nonnegative, deceasing function of g, and that $\tau(r,g,d_0)$ is an increasing function of g. We also provide a bound on the second derivative of α, which is the key step in the proof of Lemma 20.

Lemma 21 *(Properties of $\alpha(r,g,d_0)$ and $\tau(r,g,d_0)$) For fixed values of r and d_0, considering g as a variable, within the range $0 \le g < r$:*

$$\text{(i)} \quad \lim_{g \to r} \alpha(r,g,d_0) = 0,$$

$$\text{(ii)} \quad \frac{\partial \alpha(r,g,d_0)}{\partial g} = \frac{-\tau(r,g,d_0)}{g\tau'(r,g,d_0) + \tau''(r,g,d_0)} = \frac{-\tau(r,g,d_0)}{(1-g^2)\tau(r,g,d_0) - d_0} < 0,$$

$$\text{(iii)} \quad \lim_{g \to r} \frac{\partial \alpha(r,g,d_0)}{\partial g} = \frac{-1}{1-r^2-d_0} < 0,$$

$$\text{(iv)} \quad \frac{\partial \tau(r,g,d_0)}{\partial g} \ge 0,$$

$$\text{(v)} \quad \tau(r,0,1-r) = 1 - r \le d_0 + \sqrt{(1-d_0)^2 - r^2} = \tau(r,0,d_0),$$

$$\text{(vi)} \quad \frac{\partial^2 \alpha(r,g,d_0)}{\partial g^2} < 0 \text{ whenever } d_0 \le \frac{2}{3}(1-r)(1-r^2) \text{ and } g > 0.$$

Proof By definition

$$\tau(r,g,d_0) = \frac{(1+r-d_0)}{2} e^{-\alpha(r,g,d_0)} + \frac{(1-r-d_0)}{2} e^{\alpha(r,g,d_0)} + d_0,$$

$$\tau'(r,g,d_0) = -\frac{(1+r-d_0)}{2} e^{-\alpha(r,g,d_0)} + \frac{(1-r-d_0)}{2} e^{\alpha(r,g,d_0)},$$

and similarly define $\tau''(r,g,d_0) = \tau(r,g,d_0) - d_0$. Part (i) can be seen from (11), that is, $-\tau'(r,g,d_0) = g\tau(r,g,d_0)$, which simplifies to

$$\frac{(1+r-d_0)}{2} e^{-\alpha(r,g,d_0)} - \frac{(1-r-d_0)}{2} e^{\alpha(r,g,d_0)}$$

$$= g\frac{(1+r-d_0)}{2}e^{-\alpha(r,g,d_0)} + g\frac{(1-r-d_0)}{2}e^{\alpha(r,g,d_0)} + gd_0,$$

so one sets $g = r$ and verifies that $\alpha = 0$ satisfies the equation. Part (ii) is shown by taking implicit derivatives of (11) as follows:

$$\frac{\partial\alpha(r,g,d_0)}{\partial g}(g\tau'(r,g,d_0) + \tau''(r,g,d_0)) + \tau(r,g,d_0) = 0,$$

that is,

$$\frac{\partial\alpha(r,g,d_0)}{\partial g} = \frac{-\tau(r,g,d_0)}{g\tau'(r,g,d_0) + \tau''(r,g,d_0)}, \tag{24}$$

and then simplifying using (11) and the definition of $\tau''(r,g,d_0)$. For the inequality, the numerator is negative, and the denominator is (using d_+, d_- notation) $g(-d_+e^{-\alpha} + d_-e^{\alpha}) + d_+e^{-\alpha} + d_-e^{\alpha} = (1-g)d_+e^{-\alpha} + (1+g)d_-e^{\alpha} > 0$ since $g < 1$. Part (iii) is shown from (i) and (ii); for $g \to r$, we have $\alpha(r,g,d_0) \to 0$, and thus $\tau(r,g,d_0) \to 1$. The inequality comes from $1 - r^2 - d_0 > 1 - r - d_0 = 2d_- \geq 0$. To show (iv), by the chain rule,

$$\frac{\partial\tau(r,g,d_0)}{\partial g} = \tau'(r,g,d_0)\frac{\partial\alpha(r,g,d_0)}{\partial g}.$$

Since $\tau(r,g,d_0) > 0$ and $\tau'(r,g,d_0) = -g\tau(r,g,d_0)$, we know $\tau'(r,g,d_0) \leq 0$. Additionally, from (ii), $\frac{\partial\alpha}{\partial g} < 0$. Thus (iv) is proved. For (v), we know that when $g = 0$, $\tau'(r,g,d_0) = -g\tau(r,g,d_0)$ means $\tau'(r,0,d_0) = 0$. Using the definition for $\tau'(r,g,d_0)$, we find that $e^{\alpha(r,0,d_0)} = \left(\frac{1+r-d_0}{1-r-d_0}\right)^{1/2}$. Substituting this into the definition of τ yields the equality conditions in (v). The inequality comes from the fact that the right hand side, $d_0 + \sqrt{(1-d_0)^2 - r^2}$, is monotonically decreasing in d_0. For (vi), a derivative of (24) yields:

$$(g\tau'(r,g,d_0) + \tau''(r,g,d_0))\frac{\partial^2\alpha(r,g,d_0)}{\partial g^2}$$

$$= -\left(\frac{\partial\alpha(r,g,d_0)}{\partial g}\right)\left[\left(\frac{\partial\alpha(r,g,d_0)}{\partial g}\right)(g\tau''(r,g,d_0) + \tau'''(r,g,d_0)) + 2\tau'(r,g,d_0)\right],$$

where $\tau'''(r,g,d_0) = \tau'(r,g,d_0)$. The left expression (using d_+, d_- notation) is $g\tau'(r,g,d_0) + \tau''(r,g,d_0) = d_+(1-g)e^{-\alpha} + d_-(1+g)e^{\alpha} > 0$ since $g < 1$. Since (ii) shows that $\partial\alpha/\partial g < 0$, we are left to show that the bracketed expression on the right is negative in order for the second derivative of α to be negative. Consider that quantity:

$$\left(\frac{\partial\alpha(r,g,d_0)}{\partial g}\right)(g\tau''(r,g,d_0) + \tau'''(r,g,d_0)) + 2\tau'(r,g,d_0)$$

$$= \tau'(r,g,d_0)\left[\frac{\partial\alpha(r,g,d_0)}{\partial g}\left(\frac{g\tau''(r,g,d_0) + \tau'(r,g,d_0)}{\tau'(r,g,d_0)}\right) + 2\right]$$

and substituting $\tau'(r,g,d_0) = -g\tau(r,g,d_0)$ and $\tau''(r,g,d_0) = \tau(r,g,d_0) - d_0$,

$$\begin{aligned}
&= \tau'(r,g,d_0)\left[\frac{\partial\alpha(r,g,d_0)}{\partial g}\left(\frac{g\tau(r,g,d_0) - gd_0 - g\tau(r,g,d_0)}{-g\tau(r,g,d_0)}\right) + 2\right] \\
&= \tau'(r,g,d_0)\left[\frac{\partial\alpha(r,g,d_0)}{\partial g}\left(\frac{d_0}{\tau(r,g,d_0)}\right) + 2\right] \text{ and from (ii),} \\
&= \tau'(r,g,d_0)\left[\frac{-d_0}{(1-g^2)\tau(r,g,d_0) - d_0} + 2\right]. \qquad (25)
\end{aligned}$$

Since $-\tau'(r,g,d_0) = g\tau(r,g,d_0)$, we know $\tau'(r,g,d_0) < 0$ when $g > 0$. Let us show that the bracketed expression of (25) is positive. Using our assumption on d_0, also $1 - r^2 < 1 - g^2$, (v), and (iv),

$$\begin{aligned}
d_0 &< (1-r^2)(1-r)\frac{2}{3} < (1-g^2)(1-r)\frac{2}{3} = (1-g^2)\tau(r,0,1-r)\frac{2}{3} \\
&\leq (1-g^2)\tau(r,0,d_0)\frac{2}{3} \leq (1-g^2)\tau(r,g,d_0)\frac{2}{3}.
\end{aligned}$$

Rearranging this yields

$$\frac{d_0}{[(1-g^2)\tau(r,g,d_0) - d_0]} < 2.$$

The proof is finished. ■

In order to build up to Lemma 20, we need some properties of $\Gamma(r,g,d_0)$ and B.

Lemma 22 *(Properties of $\Gamma(r,g,d_0)$) For every fixed value of r and d_0, considering g as a variable, within the range $0 \leq g < r$:*

$$\begin{aligned}
&\text{(i)} && \lim_{g\to r}\Gamma(r,g,d_0) = r \\
&\text{(ii)} && \Gamma(r,g,d_0) > g \\
&\text{(iii)} && \frac{\partial\Gamma(r,g,d_0)}{\partial g} > 0 \\
&\text{(iv)} && \Gamma(r,g,d_0) < r.
\end{aligned}$$

Proof The proof of (i) uses L'Hôpital's rule, which we have permission to use from Lemma 21 (i) since $\lim_{g\to r}\alpha(r,g,d_0) = 0$.

$$\lim_{g\to r}\Gamma(r,g,d_0) = \lim_{g\to r}\frac{-\ln\tau(r,g,d_0)}{\alpha(r,g,d_0)} = \lim_{g\to r}\frac{-\frac{\tau'(r,g,d_0)}{\tau(r,g,d_0)}\frac{\partial\alpha(r,g,d_0)}{\partial g}}{\frac{\partial\alpha(r,g,d_0)}{\partial g}} = \lim_{g\to r} g = r.$$

Here we have used that $\lim_{g\to r}\frac{\partial\alpha(r,g,d_0)}{\partial g}$ is finite from Lemma 21 (ii), and applied (11), that is, $-\tau'(r,g,d_0) = g\tau(r,g,d_0)$.

For the proofs of (ii) and (iii) we consider the derivative of $\Gamma(r,g,d_0)$ with respect to g. Recall that $\tau'(r,g,d_0)$ is given by the formula (8).

$$\begin{aligned}
\frac{\partial\Gamma(r,g,d_0)}{\partial g} &= \left[\frac{-\tau'(r,g,d_0)}{\tau(r,g,d_0)} + \frac{\ln\tau(r,g,d_0)}{\alpha(r,g,d_0)}\right]\frac{1}{\alpha(r,g,d_0)}\frac{\partial\alpha(r,g,d_0)}{\partial g} \\
&= (\Gamma(r,g,d_0) - g)\left(-\frac{\partial\alpha(r,g,d_0)}{\partial g}\right)\frac{1}{\alpha(r,g,d_0)}. \qquad (26)
\end{aligned}$$

For the last line above we used (11) and the definition of $\Gamma(r,g,d_0)$. Since $\alpha(r,g,d_0)$ is a positive, decreasing function of g from Lemma 21 (ii), we know $-\partial\alpha(r,g,d_0)/\partial g$ and $1/\alpha(r,g,d_0)$ are positive. Thus,

$$\text{sign}\left(\frac{\partial\Gamma(r,g,d_0)}{\partial g}\right) = \text{sign}(\Gamma(r,g,d_0)-g). \tag{27}$$

We show next that $\Gamma(r,0,d_0) > 0$. From (11), we know $0 = -\tau'(r,0,d_0)$, which by definition of $\tau'(r,g,d_0)$ in (8) gives $\alpha(r,0,d_0) = \frac{1}{2}\ln(d_+/d_-) > 0$. Now,

$$\Gamma(r,0,d_0) = \frac{-\ln\tau(r,0,d_0)}{\alpha(r,0,d_0)} = \frac{1}{\alpha(r,0,d_0)}\left(-\ln\left(2(d_-d_+)^{1/2}+d_0\right)\right).$$

We also have $2(d_-d_+)^{1/2}+d_0 < d_-+d_++d_0 = 1$, so we are done showing that $\Gamma(r,0,d_0) > 0$.

Now we proceed by contradiction. Assume that there is some value of $\bar{g}$, where $0 \le \bar{g} < r$, for which $\Gamma(r,\bar{g},d_0) \le \bar{g}$. That is, assume the functions $\Gamma(r,g,d_0)$ and $f(g) = g$ cross. In that case, the derivative $\partial\Gamma(r,g,d_0)/\partial g$ would have a nonpositive sign at $g = \bar{g}$ by (27), and the function $\Gamma(r,g,d_0)$ would be a nonincreasing function for $\bar{g} < g < r$. That is, since $\Gamma(r,g,d_0)$ would have a nonpositive slope at $\bar{g}$, it cannot increase to cross the line $f(g) = g$ in order to reverse the sign of the slope. However, this is a contradiction, since the function must indeed increase; it must reach the limiting value r as $g \to r$, as we showed in (i). Hence, $\Gamma(r,g,d_0) > g$ for all g such that $0 \le g < r$, proving (ii), and thus by (27), $\partial\Gamma(r,g,d_0)/\partial g > 0$ for all g such that $0 \le g < r$, proving (iii).

The proof of (iv) is again by contradiction. Fix arbitrary values of r and d_0. Assume $\Gamma(r,\bar{g},d_0) \ge r$ for some $\bar{g} < r$. Since the function $\Gamma(r,g,d_0)$ is an increasing function of g, $\Gamma(r,g,d_0)$ must be larger than r and strictly increasing for $g > \bar{g}$. Yet by (i), $\Gamma(r,g,d_0) \to r$ as $g \to r$ for each fixed pair of r and d_0. This is a contradiction, since $\Gamma(r,g,d_0)$ cannot decrease to meet this limit. ■

Lemma 23

(i) $\quad 0 < \mathcal{B}(r,g,d_0) < 1$

(ii) $\quad \lim_{g\to r}\mathcal{B}(r,g,d_0) = \frac{1}{2}$ *for fixed r and* d_0.

Proof From Lemma 22 (ii), $\Gamma(r,g,d_0)-g$ is positive, and by assumption $g < r$. Thus, $\mathcal{B}(r,g,d_0) > 0$. Also, from Lemma 22 (iv), $\Gamma(r,g,d_0) < r$. Thus, $\mathcal{B}(r,g,d_0) < 1$. Thus (i) is proved. The proof of (ii) uses L'Hôpital's rule twice (which we may use by Lemma 21 (i)) also (11), and the fact that derivatives of $\alpha(r,g,d_0)$ with respect to g are finite.

$$\begin{aligned}
\lim_{g\to r}\mathcal{B}(r,g,d_0) &= \lim_{g\to r}\frac{\frac{-\ln\tau(r,g,d_0)}{\alpha(r,g,d_0)}-g}{r-g} = \lim_{g\to r}\frac{-\ln\tau(r,g,d_0)-g\alpha(r,g,d_0)}{\alpha(r,g,d_0)(r-g)} \\
&= \lim_{g\to r}\frac{-\frac{\tau'(r,g,d_0)}{\tau(r,g,d_0)}\frac{\partial\alpha(r,g,d_0)}{\partial g}-g\frac{\partial\alpha(r,g,d_0)}{\partial g}-\alpha(r,g,d_0)}{-\alpha(r,g,d_0)+(r-g)\frac{\partial\alpha(r,g,d_0)}{\partial g}} \\
&= \lim_{g\to r}\frac{-\alpha(r,g,d_0)}{-\alpha(r,g,d_0)+(r-g)\frac{\partial\alpha(r,g,d_0)}{\partial g}} \\
&= \lim_{g\to r}\frac{-\frac{\partial\alpha(r,g,d_0)}{\partial g}}{-2\frac{\partial\alpha(r,g,d_0)}{\partial g}+(r-g)\frac{\partial^2\alpha(r,g,d_0)}{\partial g^2}} = \frac{1}{2}.
\end{aligned}$$

■

There is one thing left in order to prove Lemma 20. This is where the key step appears, that is, our bound on the second derivative of α.

Lemma 24

$$\frac{(r-g)}{\alpha(r,g,d_0)}\left(-\frac{\partial\alpha(r,g,d_0)}{\partial g}\right) < 1.$$

Proof Define

$$\phi(r,g,d_0) := (r-g)\left(-\frac{\partial\alpha(r,g,d_0)}{\partial g}\right) - \alpha(r,g,d_0).$$

In order to prove the lemma, we need only to show that $\phi(r,g,d_0)$ is always negative. We will show that $\partial\phi(r,g,d_0)/\partial g$ is positive. Thus, the largest possible value of $\phi(r,g,d_0)$ occurs when g is at its maximum, namely, when $g = r$. If $g = r$, then $\phi(r,g,d_0) = 0$. Thus, $\phi(r,g,d_0)$ is everywhere negative and the lemma is proved. Now we have only to prove $\partial\phi(r,g,d_0)/\partial g$ is positive. Again, we take derivatives:

$$\frac{\partial\phi(r,g,d_0)}{\partial g} = (r-g)\left(-\frac{\partial^2\alpha(r,g,d_0)}{\partial g^2}\right),$$

and since $r-g$ is always positive, and since we have taken efforts to ensure α's second derivative is negative (except at the irrelevant endpoint $g = 0$) in Lemma 21 (vi), we are done. ■

We finally prove Lemma 20.

Proof *(of Lemma 20)* We consider $\partial\mathcal{B}(r,g,d_0)/\partial g$ for each fixed pair of r and d_0 values and derive a differential equation for $\mathcal{B}$. We will prove that the derivative is always nonnegative. Then we will use Lemma 23 to show that $\mathcal{B}(r,g,d_0)$ is nonnegative. Here is the differential equation:

$$\begin{aligned}
\frac{\partial\mathcal{B}(r,g,d_0)}{\partial g} &= \frac{1}{r-g}\left[\frac{\partial\Gamma(r,g,d_0)}{\partial g} - 1 + \frac{\Gamma(r,g,d_0)-g}{r-g}\right] \\
&= \frac{1}{r-g}\left[\frac{\partial\Gamma(r,g,d_0)}{\partial g} - 1 + \mathcal{B}(r,g,d_0)\right] \\
&= \frac{1}{r-g}\left[(\Gamma(r,g,d_0)-g)\left(-\frac{\partial\alpha(r,g,d_0)}{\partial g}\frac{1}{\alpha(r,g,d_0)}\right) - 1 + \mathcal{B}(r,g,d_0)\right] \\
&= \frac{1}{r-g}\left[\mathcal{B}(r,g,d_0)(r-g)\left(-\frac{\partial\alpha(r,g,d_0)}{\partial g}\frac{1}{\alpha(r,g,d_0)}\right) - 1 + \mathcal{B}(r,g,d_0)\right] \\
&= \frac{\mathcal{B}(r,g,d_0)}{r-g}\left[(r-g)\left(-\frac{\partial\alpha(r,g,d_0)}{\partial g}\frac{1}{\alpha(r,g,d_0)}\right) - \left(\frac{1}{\mathcal{B}(r,g,d_0)} - 1\right)\right]. \qquad (28)
\end{aligned}$$

Here we have incorporated the differential equation for $\Gamma(r,g,d_0)$ from (26). Again, we will prove by contradiction. Assume that for some values of r and g, where $g < r$, we have $\mathcal{B}(r,g,d_0) \leq 1/2$. That is, assume $\left(\frac{1}{\mathcal{B}(r,g,d_0)} - 1\right) \geq 1$. In that case, the bracket in Equation (28) is negative, by Lemma 24. Since $0 < \mathcal{B}(r,g,d_0) < 1$ from Lemma 23, and $g < r$ by assumption, the factor $\mathcal{B}(r,g,d_0)/(r-g)$ of Equation (28) is positive and the bracket is negative, thus $\frac{\partial\mathcal{B}(r,g,d_0)}{\partial g} < 0$, so $\mathcal{B}(r,g,d_0)$ is a decreasing function. Hence, for each fixed r and d_0, $\mathcal{B}(r,g,d_0)$ decreases from a value

which is less than or equal to $1/2$. Recall from Lemma 23 that $\lim_{g\to r} \mathcal{B}(r,g,d_0) = 1/2$, and thus this limit can never be attained. Contradiction. Thus, for all values of r, d_0 and g within $0 < r < 1$, $0 \le g < r$, $0 \le d_0 < \frac{2}{3}(1-r)(1-r^2)$, we must have $\mathcal{B}(r,g,d_0) > 1/2$. We have proved the lemma. ■

References

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.

Peter L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, March 1998.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

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

Ulf Brefeld and Tobias Scheffer. AUC maximizing support vector learning. In *Proceedings of the ICML 2005 Workshop on ROC Analysis in Machine Learning*, 2005.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the Twenty-second International Conference on Machine Learning*, pages 89–96, 2005.

Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the Twenty-third International Conference on Machine Learning*, 2006.

Stéphan Clemençon and Nicolas Vayatis. Ranking the best instances. *Journal of Machine Learning Research*, 8:2671–2699, Dec 2007.

Stéphan Clemençon, Gabor Lugosi, and Nicolas Vayatis. Ranking and scoring using empirical risk minimization. In *Proceedings of the Eighteenth Annual Conference on Learning Theory*, 2005.

Stéphan Clemençon, Gabor Lugosi, and Nicolas Vayatis. Ranking and empirical minimization of U-statistics. *The Annals of Statistics*, 36(2):844–874, 2007.

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

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

Corinna Cortes and Mehryar Mohri. Confidence intervals for the area under the ROC curve. In *Advances in Neural Information Processing Systems 17*, 2005.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.

David Cossock and Tong Zhang. Statistical analysis of bayes optimal subset ranking. *IEEE Trans. Info. Theory*, 54:4140–5154, 2008.

Ofer Dekel, Christopher Manning, and Yoram Singer. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems 16*, 2004.

Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Duality and auxiliary functions for Bregman distances. Technical Report CMU-CS-01-109R, School of Computer Science, Carnegie Mellon University, 2002.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

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

Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

Martin Gardner. *The Colossal Book of Mathematics*, chapter on More Nontransitive Paradoxes, pages 297–311. Norton, 2001.

Quoc Le and Alex Smola. Direct optimization of ranking measures. arXiv:0704.3359v1, November 2007.

Colin McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics 1989*, pages 148–188. Cambridge University Press, 1989.

Alain Rakotomamonjy. Optimizing AUC with support vector machine (SVM). In *Proceedings of European Conference on Artificial Intelligence Workshop on ROC Curve and AI, Valencia, Spain*, 2004.

Cynthia Rudin. The P-Norm Push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, October 2009.

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.

Cynthia Rudin, Corinna Cortes, Mehryar Mohri, and Robert E. Schapire. Margin-based ranking meets boosting in the middle. In *Proceedings of the Eighteenth Annual Conference on Learning Theory*, pages 63–78, 2005.

Cynthia Rudin, Robert E. Schapire, and Ingrid Daubechies. Analysis of boosting algorithms using the smooth margin function. *The Annals of Statistics*, 35(6):2723–2768, 2007.

Robert E. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, December 1999.

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.

Shai Shalev-Shwartz and Yoram Singer. Efficient learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research*, 7:1567–1599, December 2006.

Nicolas Usunier, Massih-Reza Amini, and Patrick Gallinari. A data-dependent generalisation error bound for the AUC. In *Proceedings of the ICML 2005 Workshop on ROC Analysis in Machine Learning*, 2005.

Tong Zhang and Bin Yu. Boosting with early stopping - convergence and consistency. *The Annals of Statistics*, 33(4):1538–1579, 2005.