A STUDY OF THE BIPARTITE RANKING PROBLEM IN MACHINE LEARNING

BY

SHIVANI AGARWAL

B.Sc., University of Delhi, 1998
B.A., University of Cambridge, 2000
M.S., University of Illinois at Urbana-Champaign, 2002

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2005

Urbana, Illinois

Abstract

The problem of ranking, in which the goal is to learn a real-valued ranking function that induces a ranking or ordering over an instance space, has recently gained attention in machine learning. A particular setting of interest is the bipartite ranking problem, in which instances come from two categories, positive and negative; the learner is given examples of instances labeled as positive or negative, and the goal is to learn from these examples a ranking function that ranks future positive instances higher than negative ones. This thesis makes four important contributions to the understanding of the bipartite ranking problem.

First, we derive large deviation and uniform convergence bounds for the bipartite ranking error, a quantity used to measure the quality of a bipartite ranking function. The large deviation bound serves to bound the expected error of a ranking function in terms of its empirical error on an independent test sample; the uniform convergence bound serves to bound the expected error of a learned ranking function in terms of its empirical error on the training sample from which it is learned. The uniform convergence bound is expressed in terms of a new set of combinatorial parameters that we term the *bipartite rank-shatter coefficients*.

Second, we define a model of learnability for bipartite ranking functions, and derive a number of results in this model. In particular, we derive both upper and lower bounds on the sample complexity of learning ranking functions in this model. The upper bound is expressed in terms of the bipartite rank-shatter coefficients. The lower bound is expressed in terms of another new combinatorial parameter that we term the *rank dimension*.

Third, we derive generalization bounds for bipartite ranking algorithms based on the notion of algorithmic stability. Unlike bounds based on uniform convergence, these bounds can be applied also to algorithms that search function classes of unbounded complexity. In particular, we are able to apply our results to obtain generalization bounds for kernel-based ranking algorithms, to which bounds based on uniform convergence are often not applicable.

Finally, we demonstrate a practical application of bipartite ranking to a problem in bioinformatics, namely the problem of identifying genes related to a given disease based on microarray data. Our studies on leukemia and colon cancer data sets show very promising results, including the identification of some exciting candidate genes as potential targets for drug development.

Publication Notes

The results in Chapter 2 were presented at the *18th Annual Conference on Neural Information Processing Systems* in December 2004 [2]; the results in Chapter 3 were presented at the *10th International Workshop on Artificial Intelligence and Statistics* in January 2005 [3]. A longer paper combining the results of both chapters appeared in the *Journal of Machine Learning Research* [1]. The results in Chapters 4 and 5 are to be be presented at the *18th Annual Conference on Learning Theory* in June 2005 [5, 4]. A manuscript based on the results in Chapter 6 is in preparation for submission.

To my parents
for letting me pursue my dreams
for so long
so far away from home
&
To my husband
for giving me
new dreams to pursue

Acknowledgments

Thanks go first to my advisor, Dan Roth, for initiating me into research, for providing constant support and encouragement, and for giving me the freedom to pursue my interests. I am also grateful to him for arranging most of my financial support, and to Eyal Amir for arranging my support toward the end of my program.

I would like to thank the members of my doctoral committee – Eyal Amir, Gerald DeJong, Sariel Har-Peled, Dan Roth, and Nikolaos Sahinidis – for taking out time to talk to me about my research, for reading various versions of my thesis, and for providing valuable feedback.

I owe special thanks to Sariel Har-Peled, who has been an invaluable source of inspiration, support and advice. It was with him that I received my first teaching opportunity, and since then I have turned to him for all kinds of help. I also owe special thanks to Christopher Bishop, with whom I interned at Microsoft Research, Cambridge. Although my work with him was on a different topic and does not appear in this thesis, I learned a lot from working with him, and he too has been a constant source of support and encouragement.

Much of the research described in this thesis is a result of collaborative efforts. Chapter 2 is joint work with Thore Graepel, Ralf Herbrich, and Dan Roth; Chapter 3 with Sariel Har-Peled and Dan Roth; Chapter 4 with Dan Roth; Chapter 5 with Partha Niyogi; and Chapter 6 with Shiladitya Sengupta. I am grateful to each of my collaborators for all I have learned from working with them. I am also grateful to the anonymous reviewers of various conference and journal papers that I have written with my collaborators; the criticisms and suggestions provided by their reviews have directly improved the quality of this thesis.

I have been fortunate to have the love and affection of a most wonderful family, especially my parents, who have always encouraged me to follow my dreams, and my husband Shiladitya, whose love, support and care have helped me through many difficult times. This thesis is dedicated to them.

Table of Contents

Chapter 1

Introduction

During the last three decades, considerable progress has been made in the understanding of binary classification (learning of binary-valued functions) and regression (learning of real-valued functions), both classical problems in machine learning. Although numerous questions remain to be answered, there is a well-developed theory in place for these problems, and practical successes have been demonstrated in a variety of applications.

Recently, a new learning problem, namely that of *ranking*, has begun to gain attention [21, 48, 26, 85, 37]. In ranking, one learns a real-valued function that assigns scores to objects, but the scores themselves do not matter; instead, what is important is the relative ranking of objects induced by those scores. This problem is distinct from both classification and regression, and it is natural to ask whether a strong understanding can be developed for this problem. This thesis attempts to develop such an understanding for a particular setting of the ranking problem known as the *bipartite* ranking problem.

1.1 The Binary Classification Problem

In the binary classification problem, there is an instance space $\mathcal{X}$ from which instances are drawn, and there is a set of two class labels taken without loss of generality to be $\{-1, 1\}$. The learner is given a finite sequence of labeled training examples $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \{-1, 1\})^m$, and the goal is to learn from these examples a binary-valued classification function $h : \mathcal{X} \to \{-1, 1\}$ that predicts accurately the class labels for new instances.

The following loss function is useful in measuring the quality of a binary classification function:

Definition 1.1 (Binary classification loss) *The* binary classification loss *(also called* zero-one loss*), denoted by* ℓ_{class}*, is the function* $\ell_{\text{class}} : \{-1, 1\}^{\mathcal{X}} \times \mathcal{X} \times \{-1, 1\} \to \mathbb{R}^+ \cup \{0\}$ *defined*

by

$$\ell_{\text{class}}(h, x, y) \quad = \quad \mathbf{I}_{\{h(x) \neq y\}}$$

for all $h : \mathcal{X} \to \{-1, 1\}$ *and* $(x, y) \in \mathcal{X} \times \{-1, 1\}$*, where* $\mathbf{I}_{\{\cdot\}}$ *denotes the indicator variable whose value is one if its argument is true and zero otherwise.*

It is generally assumed that all examples (x, y) (both training examples and future, unseen examples) are drawn randomly and independently according to some (unknown) distribution $\mathcal{D}$ on $\mathcal{X} \times \{-1, 1\}$. The quality of a classification function is then measured by its *expected binary classification error* with respect to $\mathcal{D}$, defined as follows:

Definition 1.2 (Expected binary classification error) *Let* $h : \mathcal{X} \to \{-1, 1\}$ *be a classification function on* $\mathcal{X}$*, and let* $\mathcal{D}$ *be a distribution on* $\mathcal{X} \times \{-1, 1\}$*. The* expected binary classification error *(or simply* expected classification error *or* expected error*) of* h *with respect to* $\mathcal{D}$*, denoted by* $L_{\mathcal{D}}(h)$*, is defined as*

$$L_{\mathcal{D}}(h) \quad = \quad \mathbf{E}_{(x,y)\sim\mathcal{D}}\Big\{\ell_{\text{class}}(h, x, y)\Big\}\,.$$

The expected classification error $L_{\mathcal{D}}(h)$ is the probability that an example drawn randomly according to $\mathcal{D}$ will be misclassified by h. In practice, since the distribution $\mathcal{D}$ is not known, the expected error of a classification function cannot be computed exactly. Instead, it must be estimated using a finite data sample. A widely used estimate is the following:

Definition 1.3 (Empirical binary classification error) *Let* $h : \mathcal{X} \to \{-1, 1\}$ *be a classification function on* $\mathcal{X}$*, and let* $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \{-1, 1\})^m$*. The* empirical binary classification error *(or simply* empirical classification error *or* empirical error*) of* h *with respect to* S*, denoted by* $\hat{L}_S(h)$*, is defined as*

$$\hat{L}_S(h) \quad = \quad \frac{1}{m}\sum_{i=1}^{m} \ell_{\text{class}}(h, x_i, y_i)\,.$$

It is easily seen that the empirical error of a classification function h is an unbiased estimator of the expected error of h, *i.e.*, that $\mathbf{E}_{S\sim\mathcal{D}^m}\{\hat{L}_S(h)\} = L_{\mathcal{D}}(h)$. When the examples in S are drawn randomly and independently according to $\mathcal{D}$, the sequence S constitutes a random sample. Much work in learning theory research has concentrated on developing bounds on the probability that an error estimate obtained from such a random sample will have a large deviation from the expected error. While the expected error of a classification function may not

be exactly computable, such bounds allow one to compute confidence intervals within which the expected value is likely to be contained with high probability.

1.2 The Regression Problem

The regression problem is similar to the binary classification problem; there is an instance space $\mathcal{X}$, and the goal is to learn to predict labels of instances drawn from this space. The difference is that the labels in regression are real-valued rather than binary. Specifically, the learner is given a finite sequence of training examples $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathbb{R})^m$, and the goal is to learn from these examples a real-valued regression function[1] $f : \mathcal{X} \to \mathbb{R}$ that predicts accurately labels of new instances.

The following two loss functions are commonly used in measuring the quality of a regression function:

Definition 1.4 (Absolute loss) *The* absolute loss*, denoted by* ℓ_{abs}*, is the function* $\ell_{\text{abs}} : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathbb{R} \to \mathbb{R}^+ \cup \{0\}$ *defined by*

$$\ell_{\text{abs}}(f, x, y) \quad = \quad |f(x) - y|$$

for all $f : \mathcal{X} \to \mathbb{R}$ *and* $(x, y) \in \mathcal{X} \times \mathbb{R}$*.*

Definition 1.5 (Squared loss) *The* squared loss*, denoted by* ℓ_{sq}*, is the function* $\ell_{\text{sq}} : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathbb{R} \to \mathbb{R}_+ \cup \{0\}$ *defined by*

$$\ell_{\text{sq}}(f, x, y) \quad = \quad (f(x) - y)^2$$

for all $f : \mathcal{X} \to \mathbb{R}$ *and* $(x, y) \in \mathcal{X} \times \mathbb{R}$*.*

As in classification, it is generally assumed that all examples (x, y) (both training examples and future, unseen examples) are drawn randomly and independently according to some (unknown) distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{R}$. The quality of a regression function is then measured by its *expected absolute error* or *expected squared error* with respect to $\mathcal{D}$, defined as follows:

Definition 1.6 (Expected absolute/squared error) *Let* $f : \mathcal{X} \to \mathbb{R}$ *be a regression function on* $\mathcal{X}$*, and let* $\mathcal{D}$ *be a distribution on* $\mathcal{X} \times \mathbb{R}$*. The* expected absolute error *of* f *with respect to*

[1] The term 'regression function' is often used in the binary classification literature to refer to the function $\eta(x) = \mathbf{P}\{y = 1|x\}$, which gives, for each instance $x \in \mathcal{X}$, the conditional probability of a positive label given x. In this thesis, we use the term to refer simply to a real-valued function in the context of the regression problem.

$\mathcal{D}$, denoted by $L^{\text{abs}}_{\mathcal{D}}(f)$, is defined as

$$L^{\text{abs}}_{\mathcal{D}}(f) \;=\; \mathbf{E}_{(x,y)\sim\mathcal{D}}\Big\{\ell_{\text{abs}}(f,x,y)\Big\}\,.$$

Similarly, the expected squared error *of f with respect to $\mathcal{D}$, denoted by $L^{\text{sq}}_{\mathcal{D}}(f)$, is defined as*

$$L^{\text{sq}}_{\mathcal{D}}(f) \;=\; \mathbf{E}_{(x,y)\sim\mathcal{D}}\Big\{\ell_{\text{sq}}(f,x,y)\Big\}\,.$$

As in classification, since the distribution $\mathcal{D}$ is not known, the expected absolute or squared error cannot be computed exactly; instead, empirical estimates must be used:

Definition 1.7 (Empirical absolute/squared error) *Let $f : \mathcal{X}\to\mathbb{R}$ be a regression function on $\mathcal{X}$, and let $S = ((x_1,y_1),\ldots,(x_m,y_m)) \in (\mathcal{X}\times\mathbb{R})^m$. The* empirical absolute error *of f with respect to S, denoted by $\hat{L}^{\text{abs}}_S(f)$, is defined as*

$$\hat{L}^{\text{abs}}_S(f) \;=\; \frac{1}{m}\sum_{i=1}^{m}\ell_{\text{abs}}(f,x_i,y_i)\,.$$

Similarly, the empirical squared error *of f with respect to S, denoted by $\hat{L}^{\text{sq}}_S(f)$, is defined as*

$$\hat{L}^{\text{sq}}_S(f) \;=\; \frac{1}{m}\sum_{i=1}^{m}\ell_{\text{sq}}(f,x_i,y_i)\,.$$

It is easily seen that for any $f : \mathcal{X}\to\mathbb{R}$, $\mathbf{E}_{S\sim\mathcal{D}^m}\{\hat{L}^{\text{abs}}_S(f)\} = L^{\text{abs}}_{\mathcal{D}}(f)$ and $\mathbf{E}_{S\sim\mathcal{D}^m}\{\hat{L}^{\text{sq}}_S(f)\} = L^{\text{sq}}_{\mathcal{D}}(f)$. As in the case of classification, there has been much work in learning theory research on deriving probabilistic bounds on the expected absolute or squared error of a regression function using empirical estimates obtained from a finite sample.

1.3 The Bipartite Ranking Problem

The goal in ranking is to learn a ranking or ordering over an instance space $\mathcal{X}$ from a finite number of examples of order relationships among instances in this space. In the general ranking problem, the learner is given training examples in the form of ordered pairs of instances $(x,x') \in \mathcal{X}\times\mathcal{X}$ labeled with a ranking preference $r\in\mathbb{R}$, with the interpretation that x is to be ranked higher than x' if $r>0$ and lower than x' if $r<0$ ($r=0$ indicates no ranking preference between the two instances); the penalty for mis-ordering such a pair is proportional to $|r|$. Given a finite number of such examples, the goal is to learn a real-valued ranking function $f : \mathcal{X}\to\mathbb{R}$ that ranks accurately future instances; f is considered to rank an instance $x\in\mathcal{X}$ higher than an instance $x'\in\mathcal{X}$ if $f(x) > f(x')$, and lower than x' if $f(x) < f(x')$.

In the bipartite ranking problem [37], instances come from two categories, positive and negative. The learner is given a training sample $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ consisting of a sequence of positive training examples $S_+ = (x_1^+, \ldots, x_m^+) \in \mathcal{X}^m$ and a sequence of negative training examples $S_- = (x_1^-, \ldots, x_n^-) \in \mathcal{X}^n$, and the goal is to learn from this sample a ranking function $f : \mathcal{X} \to \mathbb{R}$ that ranks future positive instances higher than negative ones, *i.e.*, that assigns higher values to positive instances than to negative ones. This form of ranking problem arises, for example, in information retrieval, where one is interested in retrieving documents from some database that are relevant to a given topic. In this case, the training examples consist of documents labeled as relevant (positive) or irrelevant (negative), and the goal is to produce a list of documents that contains relevant documents at the top and irrelevant ones at the bottom; in other words, one wants a ranking of the documents in which relevant documents are ranked higher than irrelevant documents.

In the framework of the general ranking problem described above, a training sample $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ in the bipartite ranking problem can be viewed as consisting of mn ordered pairs of the form (x_i^+, x_j^-), for $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$, with the ranking preference for each pair being simply $r = 1$. If the training sample in a ranking problem is represented as a graph in which instances form vertices and an edge is drawn between each pair of vertices for which a non-zero ranking preference is given, the graph for a sample of the above form turns out to be bipartite. Freund et al. [37] therefore refer to this form of training sample as 'bipartite feedback', and it is for this reason that the corresponding ranking problem is termed the bipartite ranking problem.

The following loss function, defined on pairs of instances, is useful in measuring the quality of a bipartite ranking function:

Definition 1.8 (Bipartite ranking loss) *The* bipartite ranking loss*, denoted by* ℓ_{rank}*, is the function* $\ell_{\text{rank}} : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ *defined by*

$$\ell_{\text{rank}}(f, x, x') \quad = \quad \mathbf{I}_{\{f(x) < f(x')\}} + \frac{1}{2}\mathbf{I}_{\{f(x) = f(x')\}}$$

for all $f : \mathcal{X} \to \mathbb{R}$ *and* $(x, x') \in \mathcal{X} \times \mathcal{X}$*.*

The bipartite ranking loss $\ell_{\text{rank}}(f, x, x')$ of a ranking function $f : \mathcal{X} \to \mathbb{R}$ on a pair of instances $(x, x') \in \mathcal{X} \times \mathcal{X}$ is one if $f(x) < f(x')$, zero if $f(x) > f(x')$, and half if $f(x) = f(x')$.

We assume that positive instances are drawn randomly and independently according to some (unknown) distribution $\mathcal{D}_+$ on the instance space $\mathcal{X}$, and that negative instances are drawn randomly and independently according to some (unknown) distribution $\mathcal{D}_-$ on $\mathcal{X}$. The quality of a ranking function is then measured by its *expected bipartite ranking error* with respect to $\mathcal{D}_+$ and $\mathcal{D}_-$, defined as follows:

Definition 1.9 (Expected bipartite ranking error) *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$, and let $\mathcal{D}_+, \mathcal{D}_-$ be distributions on $\mathcal{X}$. The* expected bipartite ranking error *(or simply* expected ranking error *or* expected error*) of f with respect to $\mathcal{D}_+$ and $\mathcal{D}_-$, denoted by $R_{\mathcal{D}_+,\mathcal{D}_-}(f)$, is defined as*

$$R_{\mathcal{D}_+,\mathcal{D}_-}(f) \;=\; \mathbf{E}_{x^+ \sim \mathcal{D}_+, x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x^+, x^-) \right\}.$$

The expected ranking error $R_{\mathcal{D}_+,\mathcal{D}_-}(f)$ is the probability that a positive instance drawn randomly according to $\mathcal{D}_+$ is ranked lower by f than a negative instance drawn randomly according to $\mathcal{D}_-$, assuming that ties are broken uniformly at random.[2] In practice, since the distributions $\mathcal{D}_+$ and $\mathcal{D}_-$ are unknown, the expected error of a ranking function cannot be computed exactly. Instead, it must be estimated from an empirically observable quantity such as the following:

Definition 1.10 (Empirical bipartite ranking error) *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$, and let $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$. The* empirical bipartite ranking error *(or simply* empirical ranking error *or* empirical error*) of f with respect to (S_+, S_-), denoted by $\hat{R}_{S_+,S_-}(f)$, is defined as*

$$\hat{R}_{S_+,S_-}(f) \;=\; \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell_{\text{rank}}(f, x_i^+, x_j^-)\,.$$

The empirical ranking error $\hat{R}_{S_+,S_-}(f)$ is the fraction of positive-negative pairs in (S_+, S_-) that are ranked incorrectly by f, assuming again that ties are broken uniformly at random. The following simple lemma shows that the empirical error of a ranking function f is an unbiased estimator of the expected error of f.

Lemma 1.1 *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$, let $m, n \in \mathbb{N}$, and let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on $\mathcal{X}$. Then*

$$\mathbf{E}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \hat{R}_{S_+,S_-}(f) \right\} \;=\; R_{\mathcal{D}_+,\mathcal{D}_-}(f)\,.$$

Proof By linearity of expectation, we have

$$\begin{aligned}
\mathbf{E}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \hat{R}_{S_+,S_-}(f) \right\} &= \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{E}_{x_i^+ \sim \mathcal{D}_+, x_j^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x_i^+, x_j^-) \right\} \\
&= \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} R_{\mathcal{D}_+,\mathcal{D}_-}(f) \\
&= R_{\mathcal{D}_+,\mathcal{D}_-}(f)\,.
\end{aligned}$$

□

[2] Freund et al. [37] define a slightly simpler form of ranking error that does not account for ties.

We note that a quantity relating to receiver operating characteristic (ROC) curves, namely the area under the ROC curve (AUC), has been used to measure the empirical performance of a ranking function [23, 89, 1]. This quantity is simply equal to one minus the empirical ranking error defined above.[3]

Although the bipartite ranking problem shares similarities with the binary classification problem, it should be noted that, due to the use of different loss functions, the two problems are in fact distinct. In particular, a binary-valued function $h : \mathcal{X} \to \{-1, 1\}$ that is obtained by thresholding a real-valued function $f : \mathcal{X} \to \mathbb{R}$ may have good classification performance even though the corresponding real-valued function f may have poor ranking performance. Indeed, it is possible for binary-valued functions obtained by thresholding different real-valued functions to have the same classification errors, while the ranking errors of the real-valued functions may differ significantly. For example, consider the following two rankings on a sample consisting of 4 positive and 4 negative examples:

In both cases, the error of the best classification function that can be obtained by applying a threshold is 2/8. However, the ranking error of f_1 is 4/16, whereas that of f_2 is 8/16. For a detailed analysis of this distinction, see [23].

1.4 Summary of Results

As in the case of classification and regression, an important question in ranking concerns the derivation of probabilistic bounds on the expected error of a ranking function using empirical estimates obtained from a finite sample. In Chapter 2, we study large deviation properties of the bipartite ranking error; in particular, we derive a distribution-free large deviation bound which serves to bound the expected error of a ranking function in terms of its empirical error on an independent test sample. A comparison of our result with a corresponding large deviation result for the classification error suggests that, in the distribution-free setting, the test sample size required to obtain an ϵ-accurate estimate of the expected error of a ranking function with δ-confidence is larger than that required to obtain an ϵ-accurate estimate of the expected error of a classification function with the same confidence.

In Chapter 3, we study uniform convergence properties of the bipartite ranking error; in particular, we derive a distribution-free uniform convergence bound which serves to bound the expected error of a learned ranking function in terms of its empirical error on the training

[3] As in the case of the ranking error definition used in [37], the AUC definition of [23] does not account for ties; this is easily remedied.

sample from which it is learned. Our uniform convergence bound is expressed in terms of a new set of combinatorial parameters that we term the *bipartite rank-shatter coefficients*; these play the same role in our result as do the standard VC-dimension related shatter coefficients (also known as the growth function) in uniform convergence results for the classification error. A comparison of our result with a recent uniform convergence result for the ranking error derived by Freund et al. [37] shows that the bound provided by our result can be considerably tighter.

In Chapter 4, we define a model of learnability for bipartite ranking functions, analogous to existing models of learnability for classification and regression functions. We derive both an upper bound on the sample complexity of learning ranking functions in this model, which leads to a sufficient condition for the learnability of a class of ranking functions, and a lower bound on the sample complexity, which leads to a necessary condition for learnability. The upper bound, which makes use of the uniform convergence result of Chapter 3, is expressed in terms of the bipartite rank-shatter coefficients. The lower bound is expressed in terms of another new combinatorial parameter that we term the *rank dimension*. We also investigate questions of the computational complexity of learning ranking functions.

In Chapter 5, we study generalization properties of bipartite ranking algorithms using the notion of algorithmic stability; in particular, we derive generalization bounds for bipartite ranking algorithms that have good stability properties. We show that kernel-based ranking algorithms that perform regularization in a reproducing kernel Hilbert space have such stability properties, and therefore our bounds can be applied to these algorithms; this is in contrast with bounds based on uniform convergence, which require the function class searched by an algorithm to have bounded complexity and in many cases cannot be applied to kernel-based algorithms.

Finally, in Chapter 6, we demonstrate a practical application of bipartite ranking to bioinformatics. In particular, we show how the question of identifying genes related to a given disease based on microarray data can be formulated naturally as a bipartite ranking problem. Our experiments with this approach on leukemia and colon cancer data sets show very promising results, including the identification of some exciting candidate genes as potential targets for drug development.

Since the material in the thesis builds upon a diverse set of technical developments, we review previous work and develop necessary background as relevant in each chapter.

Chapter 2

A Large Deviation Bound for the Bipartite Ranking Error

2.1 Introduction

As discussed in Chapter 1, much work in learning theory research has focused on deriving probabilistic bounds on the expected error of a classification or regression function using empirical estimates obtained from a finite sample. We are interested in the question of deriving such bounds for the expected error of a ranking function. The question has two parts, both of which are important for machine learning practice. First, what can be said about the expected error of a ranking function based on its empirical error on an independent test sample? Second, what can be said about the expected error of a learned ranking function based on its empirical error on the training sample from which it is learned? We address the first question in this chapter; the second question is addressed in Chapter 3.

We are interested in bounding the probability that the empirical error of a ranking function f with respect to a (random) test sample (S_+, S_-) will have a large deviation from its expected error. In other words, we are interested in bounding probabilities of the form

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\}$$

for given $\epsilon > 0$. Our large deviation bound for the ranking error (Section 2.2) is derived using McDiarmid's inequality [73]; the approach is conceptually similar to that of Hill et al. [49], who considered large deviation properties of a quantity called the average precision. We compare our bound with bounds that can be obtained from the classical statistical literature (Section 2.3), as well as with a large deviation bound for the classification error (Section 2.4). A simple application of the union bound allows the large deviation bound to be extended to learned ranking functions chosen from finite function classes (Section 2.5).

2.2 Large Deviation Bound

Our main tool in deriving a large deviation bound for the ranking error will be the following powerful concentration inequality of McDiarmid [73], which bounds the deviation of any function of a sample for which a single change in the sample has limited effect:

Theorem 2.1 (McDiarmid, 1989) *Let $X_1, \ldots, X_N$ be independent random variables with X_k taking values in a set A_k for each k. Let $\phi : (A_1 \times \cdots \times A_N) \to \mathbb{R}$ be such that*

$$\sup_{x_i \in A_i, x_k' \in A_k} \left| \phi(x_1, \ldots, x_N) - \phi(x_1, \ldots, x_{k-1}, x_k', x_{k+1}, \ldots, x_N) \right| \quad \leq \quad c_k \, .$$

Then for any $\epsilon > 0$,

$$\mathbf{P}\left\{ \left| \phi(X_1, \ldots, X_N) - \mathbf{E}\Big\{ \phi(X_1, \ldots, X_N) \Big\} \right| \geq \epsilon \right\} \quad \leq \quad 2e^{-2\epsilon^2 / \sum_{k=1}^N c_k^2} \, .$$

Note that when $X_1, \ldots, X_N$ are independent bounded random variables with $X_k \in [a_k, b_k]$ with probability one, and $\phi(X_1, \ldots, X_N) = \sum_{k=1}^N X_k$, McDiarmid's inequality (with $c_k = b_k - a_k$) reduces to Hoeffding's inequality.

We shall show that for any ranking function $f : \mathcal{X} \to \mathbb{R}$, changing a single element in a sample $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ has limited effect on the empirical error $\hat{R}_{S_+,S_-}(f)$ of f, so that McDiarmid's inequality can be applied. For any $i \in \{1, \ldots, m\}$ and $z \in \mathcal{X}$, we use $S_+^{i,z}$ to denote the sequence obtained from S_+ by replacing x_i^+ with z; similarly, for any $j \in \{1, \ldots, n\}$ and $z \in \mathcal{X}$, we use $S_-^{j,z}$ to denote the sequence obtained from S_- by replacing x_j^- with z.

Lemma 2.1 *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$ and let $m, n \in \mathbb{N}$. Let $\phi : \mathcal{X}^m \times \mathcal{X}^n \to \mathbb{R}$ be defined as*

$$\phi(S_+, S_-) \quad = \quad \hat{R}_{S_+,S_-}(f) \, .$$

Then for all $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, $z \in \mathcal{X}$, $k \in \{1, \ldots, m\}$ and $l \in \{1, \ldots, n\}$,

$$\begin{aligned} \left| \phi(S_+, S_-) - \phi(S_+^{k,z}, S_-) \right| \quad &\leq \quad 1/m \, , \\ \left| \phi(S_+, S_-) - \phi(S_+, S_-^{l,z}) \right| \quad &\leq \quad 1/n \, . \end{aligned}$$

Proof Let $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, $z \in \mathcal{X}$, and let $k \in \{1, \ldots, m\}$. We have

$$\begin{aligned} &\left| \phi(S_+, S_-) - \phi(S_+^{k,z}, S_-) \right| \\ &= \quad \left| \hat{R}_{S_+,S_-}(f) - \hat{R}_{S_+^{k,z},S_-}(f) \right| \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{mn}\left|\sum_{i=1}^{m}\sum_{j=1}^{n}\ell_{\mathrm{rank}}(f,x_i^+,x_j^-) - \left(\sum_{i\neq k}\sum_{j=1}^{n}\ell_{\mathrm{rank}}(f,x_i^+,x_j^-) + \sum_{j=1}^{n}\ell_{\mathrm{rank}}(f,z,x_j^-)\right)\right| \\
&= \frac{1}{mn}\left|\sum_{j=1}^{n}\left(\ell_{\mathrm{rank}}(f,x_k^+,x_j^-) - \ell_{\mathrm{rank}}(f,z,x_j^-)\right)\right| \\
&\leq \frac{1}{mn}\sum_{j=1}^{n}\left|\ell_{\mathrm{rank}}(f,x_k^+,x_j^-) - \ell_{\mathrm{rank}}(f,z,x_j^-)\right| \\
&\leq \frac{1}{mn}n\,,
\end{aligned}
$$

since $0 \leq \ell_{\mathrm{rank}}(f,x,x') \leq 1$ for all $x,x' \in \mathcal{X}$. Similarly, it can be shown that for each $l \in \{1,\ldots,n\}$,

$$
\left|\phi(S_+,S_-) - \phi(S_+,S_-^{l,z})\right| \leq 1/n\,.
$$

This proves the lemma. □

The following is the main result of this chapter.

Theorem 2.2 *Let $f : \mathcal{X}\to\mathbb{R}$ be a fixed ranking function on $\mathcal{X}$, let $m,n \in \mathbb{N}$, and let $\mathcal{D}_+,\mathcal{D}_-$ be any distributions on $\mathcal{X}$. Then for any $\epsilon > 0$,*

$$
\mathbf{P}_{S_+\sim\mathcal{D}_+^m,S_-\sim\mathcal{D}_-^n}\left\{\left|\hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f)\right| \geq \epsilon\right\} \leq 2e^{-2mn\epsilon^2/(m+n)}\,.
$$

Proof By Lemma 2.1, we can apply McDiarmid's inequality (Theorem 2.1) to get

$$
\begin{aligned}
&\mathbf{P}_{S_+\sim\mathcal{D}_+^m,S_-\sim\mathcal{D}_-^n}\left\{\left|\hat{R}_{S_+,S_-}(f) - \mathbf{E}_{S_+\sim\mathcal{D}_+^m,S_-\sim\mathcal{D}_-^n}\left\{\hat{R}_{S_+,S_-}(f)\right\}\right| \geq \epsilon\right\} \\
&\qquad \leq 2e^{-2\epsilon^2/(m(\frac{1}{m})^2+n(\frac{1}{n})^2)} \\
&\qquad = 2e^{-2mn\epsilon^2/(m+n)}\,.
\end{aligned}
$$

The result follows from Lemma 1.1. □

From Theorem 2.2, we can derive a confidence interval interpretation of the bound that gives, for any $0 < \delta \leq 1$, a confidence interval based on the empirical error of a ranking function (on a random test sample) which is likely to contain the expected error with probability at least $1-\delta$. More specifically, we have:

Corollary 2.1 *Let $f : \mathcal{X}\to\mathbb{R}$ be a fixed ranking function on $\mathcal{X}$, let $m,n \in \mathbb{N}$, and let $\mathcal{D}_+,\mathcal{D}_-$*

be any distributions on $\mathcal{X}$. *Then for any* $0 < \delta \leq 1$,

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \sqrt{\frac{(m+n)\ln\left(\frac{2}{\delta}\right)}{2mn}} \right\} \quad \leq \quad \delta \,.$$

Proof This follows directly from Theorem 2.2 by setting $2e^{-2mn\epsilon^2/(m+n)} = \delta$ and solving for ϵ. □

We note that a different approach for deriving confidence intervals for the ranking error has recently been taken by Cortes and Mohri [24]; in particular, their confidence intervals for the ranking error[1] are constructed from confidence intervals for the classification error.

For any given proportion of positive and negative instances, Theorem 2.2 also allows us to obtain an expression for a test sample size that is sufficient to obtain, for given $0 < \epsilon, \delta \leq 1$, an ϵ-accurate estimate of the ranking error with δ-confidence:

Corollary 2.2 *Let* $f : \mathcal{X} \to \mathbb{R}$ *be a fixed ranking function on* $\mathcal{X}$, *let* $\mathcal{D}_+, \mathcal{D}_-$ *be any distributions on* $\mathcal{X}$, *and let* $0 < \epsilon, \delta \leq 1$. *Let* $\rho \in (0,1) \cap \mathbb{Q}$, *and let* $M \in \mathbb{N}$ *be such that* $m = \rho M \in \mathbb{N}$, $n = (1-\rho)M \in \mathbb{N}$. *If*

$$M \quad \geq \quad \frac{1}{2\rho(1-\rho)\epsilon^2} \ln\left(\frac{2}{\delta}\right) \,,$$

then

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \quad \leq \quad \delta \,.$$

Proof From Theorem 2.2, we have

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \quad \leq \quad 2e^{-2\rho(1-\rho)M\epsilon^2} \,.$$

The result follows by setting $2e^{-2\rho(1-\rho)M\epsilon^2} \leq \delta$ and solving for M. □

2.3 Comparison with Bounds from Statistical Literature

The empirical ranking error (defined in Chapter 1, Definition 1.10) has the form of the Wilcoxon-Mann-Whitney statistic, which has been studied extensively in the statistical literature. In

[1] Cortes and Mohri [24] actually derive confidence intervals for the AUC; as mentioned in Chapter 1, this is equal to one minus the empirical ranking error, and therefore their results imply confidence intervals for the ranking error.

particular, Lehmann [66] derives an exact expression for the variance of the Wilcoxon-Mann-Whitney statistic that can be used to obtain large deviation bounds for the ranking error. Below we compare the large deviation bound we have derived above with these bounds that can be obtained from the statistical literature. We note that the expression derived by Lehmann is for a simpler form of the Wilcoxon-Mann-Whitney statistic that does not account for ties; therefore, in this section we assume the empirical and expected ranking error are defined without the terms that account for ties (this corresponds to taking ℓ_{rank} to be $\ell_{\text{rank}}(f, x, x') = \mathbf{I}_{\{f(x)<f(x')\}}$; the large deviation result we have derived above applies also in this setting).

The variance of the empirical ranking error of a fixed ranking function $f : \mathcal{X} \to \mathbb{R}$ is given by the following expression [66]:

$$\begin{aligned} \sigma_R^2 &= \mathbf{Var}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \hat{R}_{S_+,S_-}(f) \right\} \\ &= \frac{R(f)(1-R(f)) + (m-1)(p_1 - (1-R(f))^2) + (n-1)(p_2 - (1-R(f))^2)}{mn}, \end{aligned} \tag{2.1}$$

where we have used

$$R(f) \equiv R_{\mathcal{D}_+,\mathcal{D}_-}(f)$$

to keep the expression concise, and where

$$p_1 = \mathbf{P}_{x_1^+, x_2^+ \sim \mathcal{D}_+, x_1^- \sim \mathcal{D}_-} \left\{ \{f(x_1^+) > f(x_1^-)\} \cap \{f(x_2^+) > f(x_1^-)\} \right\} \tag{2.2}$$

$$p_2 = \mathbf{P}_{x_1^+ \sim \mathcal{D}_+, x_1^-, x_2^- \sim \mathcal{D}_-} \left\{ \{f(x_1^+) > f(x_1^-)\} \cap \{f(x_1^+) > f(x_2^-)\} \right\}. \tag{2.3}$$

Next we recall the following classical inequality:

Theorem 2.3 (Chebyshev's inequality) *Let X be a random variable. Then for any $\epsilon > 0$,*

$$\mathbf{P}\left\{ |X - \mathbf{E}\{X\}| \geq \epsilon \right\} \leq \frac{\mathbf{Var}\{X\}}{\epsilon^2}.$$

The expression for the variance σ_R^2 of the ranking error can be used with Chebyshev's inequality to give the following bound: for any $\epsilon > 0$,

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \leq \frac{\sigma_R^2}{\epsilon^2}. \tag{2.4}$$

This leads to the following confidence interval: for any $0 < \delta \leq 1$,

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \frac{\sigma_R}{\sqrt{\delta}} \right\} \leq \delta. \tag{2.5}$$

It has been established that the Wilcoxon-Mann-Whitney statistic follows an asymptotically normal distribution. Therefore, when the sample size $M = m + n$ is large, one can use a normal approximation to obtain a tighter bound:

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \quad \leq \quad 2(1 - \Phi(\epsilon/\sigma_R)) \,, \tag{2.6}$$

where $\Phi(\cdot)$ denotes the standard normal cumulative distribution function given by $\Phi(u) = \int_0^u e^{-z^2/2}\, dz/\sqrt{2\pi}$. The resulting confidence interval is given by

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \sigma_R \Phi^{-1}(1 - \delta/2) \right\} \quad \leq \quad \delta \,. \tag{2.7}$$

The quantities p_1 and p_2 that appear in the expression for σ_R^2 in Eq. (2.1) depend on the underlying distributions $\mathcal{D}_+$ and $\mathcal{D}_-$; for example, Hanley and McNeil [43] derive expressions for p_1 and p_2 in the case when the scores $f(x^+)$ assigned to positive instances x^+ and the scores $f(x^-)$ assigned to negative instances x^- both follow negative exponential distributions. Distribution-independent bounds can be obtained by using the fact that the variance σ_R^2 is at most [24, 27, 14]

$$\sigma_{\max}^2 \quad = \quad \frac{R(f)(1 - R(f))}{\min(m, n)} \quad \leq \quad \frac{1}{4 \min(m, n)} \,. \tag{2.8}$$

A comparison of the resulting bounds with the large deviation bound we have derived above using McDiarmid's inequality is shown in Figure 2.1. The McDiarmid bound is tighter than the bound obtained using Chebyshev's inequality. It is looser than the bound obtained using the normal approximation; however, since the normal approximation is valid only for large M, for smaller values of M the McDiarmid bound is safer.

Of course, it should be noted that this comparison holds only in the distribution-free setting. In practice, depending on the underlying distribution, the actual variance of the ranking error may be much smaller than $\sigma_{\max}^2$; indeed, in the best case, the variance could be as small as

$$\sigma_{\min}^2 \quad = \quad \frac{R(f)(1 - R(f))}{mn} \quad \leq \quad \frac{1}{4mn} \,. \tag{2.9}$$

Therefore, one may be able to obtain tighter confidence intervals with Eqs. (2.5) and (2.7) by estimating the actual variance of the ranking error. For example, one may attempt to estimate the quantities p_1, p_2 and $R(f)$ that appear in the expression in Eq. (2.1) directly from the data, or one may use resampling methods such as the bootstrap [33], in which the variance is estimated from the sample variance observed over a number of bootstrap samples obtained from the data. The confidence intervals obtained using such estimates are only approximate (*i.e.*, the $1 - \delta$ confidence is not guaranteed), but they can often be useful in practice.

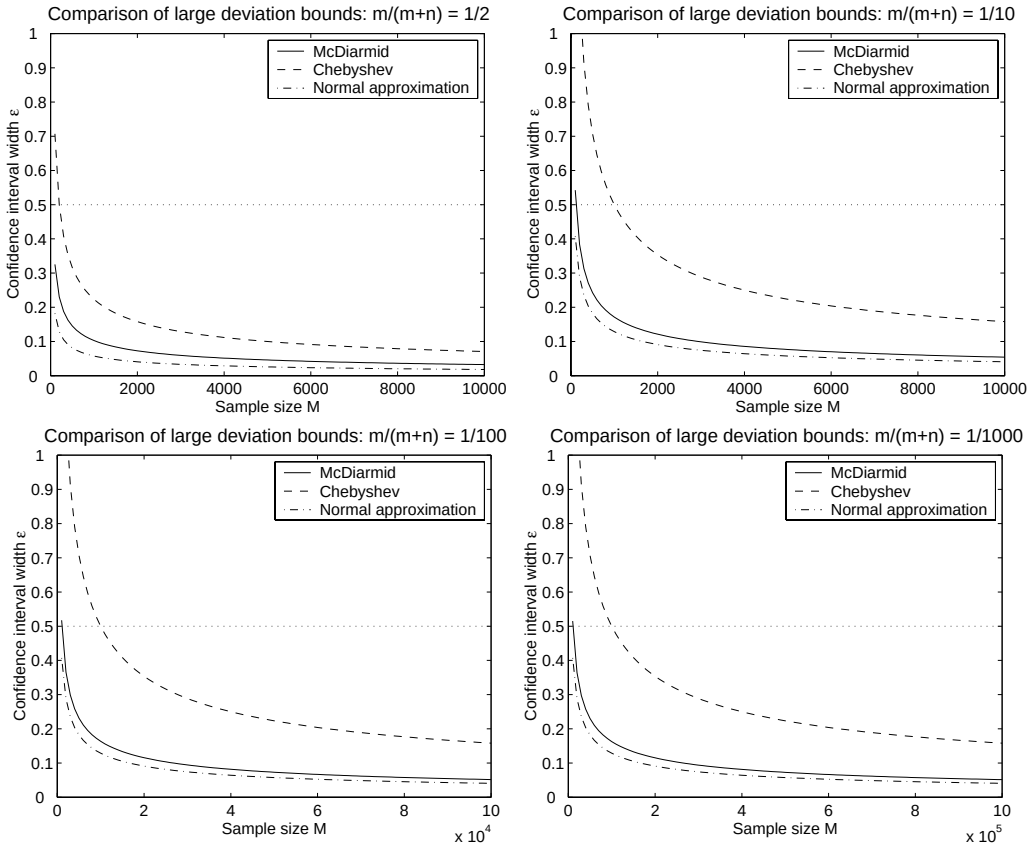

Figure 2.1: A comparison of our large deviation bound, derived using McDiarmid's inequality, with large deviation bounds obtainable from the statistical literature (see Section 2.3). The plots are for $\delta = 0.01$ and show how the confidence interval size ϵ given by the different bounds varies with the sample size $M = m + n$, for various values of $m/(m + n)$.

2.4 Comparison with Large Deviation Bound for Classification Error

Our use of McDiarmid's inequality in deriving the large deviation bound for the error of a ranking function is analogous to the use of Hoeffding's inequality in deriving a similar large deviation bound for the error of a classification function. The need for the more general inequality of McDiarmid in our derivation arises from the fact that the empirical ranking error, unlike the empirical classification error, cannot be expressed as a sum of independent random variables.

Using the notation of Chapter 1, the large deviation bound for the classification error obtained via Hoeffding's inequality (see, for example, [31, Chapter 8]) states that for a fixed classification function $h : \mathcal{X} \to \{-1, 1\}$, for any $M \in \mathbb{N}$, any distribution $\mathcal{D}$ on $\mathcal{X} \times \{-1, 1\}$ and

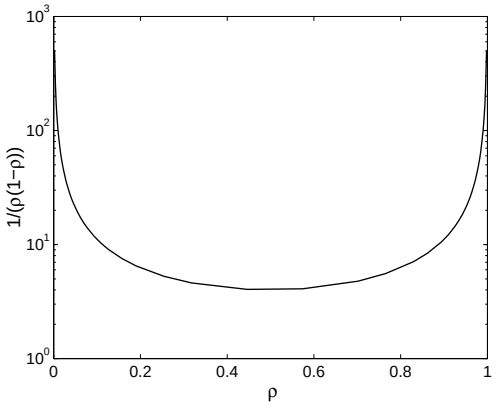

Figure 2.2: The test sample size bound for the ranking error, for a given proportion of positive examples ρ, is larger than the corresponding test sample size bound for the classification error by a factor of $1/(\rho(1-\rho))$ (see Section 2.4).

any $\epsilon > 0$,

$$\mathbf{P}_{S\sim\mathcal{D}^M}\left\{\left|\hat{L}_S(h) - L_{\mathcal{D}}(h)\right| \geq \epsilon\right\} \quad \leq \quad 2e^{-2M\epsilon^2}\,.$$

This leads to the following test sample size bound: given $0 < \epsilon, \delta \leq 1$, if

$$M \quad \geq \quad \frac{1}{2\epsilon^2}\ln\left(\frac{2}{\delta}\right)\,,$$

then

$$\mathbf{P}_{S\sim\mathcal{D}^M}\left\{\left|\hat{L}_S(h) - L_{\mathcal{D}}(h)\right| \geq \epsilon\right\} \quad \leq \quad \delta\,.$$

Comparing to the result of Corollary 2.2, we see that for a given proportion of positive examples ρ, the test sample size sufficient to obtain an ϵ-accurate estimate of the expected error of a ranking function with δ-confidence is $1/(\rho(1-\rho))$ times larger than the corresponding test sample size sufficient to obtain a similar estimate of the expected error of a classification function. For $\rho = 1/2$, this means a sample size larger by a factor of 4; as the proportion of positive examples ρ departs from $1/2$, the factor grows larger (see Figure 2.2).

Again, it should be noted that the above conclusion holds only in the distribution-free setting. Indeed, the variance σ_L^2 of the empirical classification error (which follows a binomial distribution) is given by

$$\sigma_L^2 \quad = \quad \mathbf{Var}_{S\sim\mathcal{D}^M}\left\{\hat{L}_S(h)\right\} \quad = \quad \frac{L_{\mathcal{D}}(h)(1-L_{\mathcal{D}}(h))}{M} \quad \leq \quad \frac{1}{4M}\,. \tag{2.10}$$

Comparing to Eqs. (2.8) and (2.9), we see that although this is smaller than the worst-case

variance of the ranking error, in the best case, the variance of the ranking error can be considerably smaller, leading to a tighter bound for the ranking error and therefore a smaller sufficient test sample size.

2.5 Bound for Learned Ranking Functions Chosen from Finite Function Classes

The large deviation result derived in Theorem 2.2 bounds the expected error of a ranking function in terms of its empirical error on an independent test sample. A simple application of the union bound allows the result to be extended to bound the expected error of a learned ranking function in terms of its empirical error on the training sample from which it is learned, in the case when the learned ranking function is chosen from a finite function class. More specifically, we have:

Theorem 2.4 *Let $\mathcal{F}$ be a finite class of real-valued functions on $\mathcal{X}$ and let $f_{S_+,S_-} \in \mathcal{F}$ denote the ranking function chosen by a learning algorithm based on a training sample (S_+, S_-). Let $m, n \in \mathbb{N}$ and let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on $\mathcal{X}$. Then for any $\epsilon > 0$,*

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f_{S_+,S_-}) \right| \geq \epsilon \right\} \quad \leq \quad 2|\mathcal{F}| e^{-2mn\epsilon^2/(m+n)} .$$

Proof For any $\epsilon > 0$, we have

$$\begin{aligned}
&\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f_{S_+,S_-}) \right| \geq \epsilon \right\} \\
&\quad \leq \quad \mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \max_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \\
&\quad \leq \quad \mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \bigcup_{f \in \mathcal{F}} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \right\} \\
&\quad \leq \quad \sum_{f \in \mathcal{F}} \mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \qquad \text{(by the union bound)} \\
&\quad \leq \quad 2|\mathcal{F}| e^{-2mn\epsilon^2/(m+n)} \qquad \text{(by Theorem 2.2)} .
\end{aligned}$$

□

As before, we can derive from Theorem 2.4 expressions for a confidence interval and sufficient training sample size; we give these below without proof.

Corollary 2.3 *Let $\mathcal{F}$ be a finite class of real-valued functions on $\mathcal{X}$ and let $f_{S_+,S_-} \in \mathcal{F}$ denote the ranking function chosen by a learning algorithm based on a training sample (S_+, S_-). Let*

$m, n \in \mathbb{N}$ and let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on $\mathcal{X}$. Then for any $0 < \delta \leq 1$,

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f_{S_+,S_-}) \right| \geq \sqrt{\frac{(m+n)\left(\ln|\mathcal{F}| + \ln\left(\frac{2}{\delta}\right)\right)}{2mn}} \right\} \quad \leq \quad \delta\,.$$

Corollary 2.4 *Let $\mathcal{F}$ be a finite class of real-valued functions on $\mathcal{X}$ and let $f_{S_+,S_-} \in \mathcal{F}$ denote the ranking function chosen by a learning algorithm based on a training sample (S_+, S_-). Let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on $\mathcal{X}$, and let $0 < \epsilon, \delta \leq 1$. Let $\rho \in (0,1) \cap \mathbb{Q}$, and let $M \in \mathbb{N}$ be such that $m = \rho M \in \mathbb{N}$, $n = (1-\rho)M \in \mathbb{N}$. If*

$$M \quad \geq \quad \frac{1}{2\rho(1-\rho)\epsilon^2}\left(\ln|\mathcal{F}| + \ln\left(\frac{2}{\delta}\right)\right),$$

then

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \left| \hat{R}_{S_+,S_-}(f_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f_{S_+,S_-}) \right| \geq \epsilon \right\} \quad \leq \quad \delta\,.$$

2.6 Conclusions and Open Questions

We have derived a distribution-free large deviation bound for the bipartite ranking error. Our result parallels the classical large deviation result for the classification error obtained via Hoeffding's inequality. Since the empirical ranking error cannot be expressed as a sum of independent random variables, a more powerful inequality of McDiarmid was required. A comparison with the corresponding large deviation result for the classification error suggests that, in the distribution-free setting, the test sample size required to obtain an ϵ-accurate estimate of the expected accuracy of a ranking function with δ-confidence is larger than the test sample size required to obtain a similar estimate of the expected error of a classification function.

It remains an open question whether tighter large deviation bounds can be derived using other proof techniques. A possible route for deriving an alternative large deviation bound for the ranking error could be via the theory of U-statistics; the empirical ranking error can be expressed as a two-sample U-statistic, and therefore it may be possible to apply specialized results from U-statistic theory (see, for example, [28]).

A simple application of the union bound allowed the large deviation bound to be extended to learned ranking functions chosen from finite function classes. The general case, when the learned ranking function may be chosen from a possibly infinite function class, is the subject of the next chapter.

Chapter 3

A Uniform Convergence Bound for the Bipartite Ranking Error

3.1 Introduction

In this chapter we are interested in bounding the probability that the empirical error of a learned ranking function f_{S_+,S_-} with respect to the (random) training sample (S_+, S_-) from which it is learned will have a large deviation from its expected error, when the function f_{S_+,S_-} is chosen from a possibly infinite function class $\mathcal{F}$. The standard approach for obtaining such bounds is via uniform convergence results. In particular, we have for any $\epsilon > 0$,

$$\begin{aligned}
&\mathbf{P}_{S_+\sim\mathcal{D}_+^m, S_-\sim\mathcal{D}_-^n}\left\{\left|\hat{R}_{S_+,S_-}(f_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f_{S_+,S_-})\right| \geq \epsilon\right\} \\
&\qquad \leq \quad \mathbf{P}_{S_+\sim\mathcal{D}_+^m, S_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f)\right| \geq \epsilon\right\} .
\end{aligned}$$

Therefore, to bound probabilities of the form on the left hand side above, it is sufficient to derive a uniform convergence result that bounds probabilities of the form on the right hand side. Our uniform convergence bound for the ranking error (Section 3.3) is expressed in terms of a new set of combinatorial parameters that we term the bipartite rank-shatter coefficients (Sections 3.2 and 3.4). A comparison of our bound with a recent uniform convergence bound of Freund et al. [37], which is expressed directly in terms of the standard shatter coefficients (growth function) studied in classification, shows that our bound can be considerably tighter (Section 3.5). We also provide an empirical assessment of the correctness of the functional shape of our bound (Section 3.6).

3.2 Bipartite Rank-Shatter Coefficients

We define first the notion of a bipartite rank matrix; this is used in our definition of bipartite rank-shatter coefficients.

Definition 3.1 (Bipartite rank matrix) *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$, let $m, n \in \mathbb{N}$, and let $\underline{x} = (x_1, \ldots, x_m) \in \mathcal{X}^m$, $\underline{x}' = (x'_1, \ldots, x'_n) \in \mathcal{X}^n$. Define the* bipartite rank matrix *of f with respect to $(\underline{x}, \underline{x}')$, denoted by $\mathbf{B}_f(\underline{x}, \underline{x}')$, to be the matrix in $\{0, \frac{1}{2}, 1\}^{m \times n}$ whose (i, j)-th element is given by*

$$\begin{aligned} \left[\mathbf{B}_f(\underline{x}, \underline{x}')\right]_{ij} &= \mathbf{I}_{\{f(x_i) < f(x'_j)\}} + \frac{1}{2}\mathbf{I}_{\{f(x_i) = f(x'_j)\}} \\ &= \ell_{\text{rank}}(f, x_i, x'_j) \end{aligned}$$

for all $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$.

Definition 3.2 (Bipartite rank-shatter coefficient) *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$, and let $m, n \in \mathbb{N}$. Define the* (m, n)-th bipartite rank-shatter coefficient *of $\mathcal{F}$, denoted by $r(\mathcal{F}, m, n)$, as follows:*

$$r(\mathcal{F}, m, n) = \max_{\underline{x} \in \mathcal{X}^m, \underline{x}' \in \mathcal{X}^n} \left| \left\{ \mathbf{B}_f(\underline{x}, \underline{x}') \mid f \in \mathcal{F} \right\} \right| .$$

Clearly, for finite $\mathcal{F}$, we have $r(\mathcal{F}, m, n) \leq |\mathcal{F}|$ for all m, n. In general, $r(\mathcal{F}, m, n) \leq 3^{mn}$ for all m, n. In fact, not all 3^{mn} matrices in $\{0, \frac{1}{2}, 1\}^{m \times n}$ can be realized as bipartite rank matrices. Therefore, we have

$$r(\mathcal{F}, m, n) \leq \psi(m, n) ,$$

where $\psi(m, n)$ is the number of matrices in $\{0, \frac{1}{2}, 1\}^{m \times n}$ that can be realized as a bipartite rank matrix. The number $\psi(m, n)$ can be characterized in the following ways:

Theorem 3.1 *Let $\psi(m, n)$ be the number of matrices in $\{0, \frac{1}{2}, 1\}^{m \times n}$ that can be realized as a bipartite rank matrix $\mathbf{B}_f(\underline{x}, \underline{x}')$ for some $f : \mathcal{X} \to \mathbb{R}$, $\underline{x} \in \mathcal{X}^m$, $\underline{x}' \in \mathcal{X}^n$. Then*

1. *$\psi(m, n)$ is equal to the number of complete mixed acyclic (m, n)-bipartite graphs (where a mixed graph is one which may contain both directed and undirected edges, and where we define a cycle in such a graph as a cycle that contains at least one directed edge and in which all directed edges have the same directionality along the cycle).*

2. *$\psi(m, n)$ is equal to the number of matrices in $\{0, \frac{1}{2}, 1\}^{m \times n}$ that do not contain a sub-matrix of any of the forms shown in Table 3.1.*

Table 3.1: Sub-matrices that cannot appear in a bipartite rank matrix.

$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$
$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{2} \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 1 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 1 \\ 1 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{2} \\ 1 & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 1 & \frac{1}{2} \end{bmatrix}$

Proof *Part 1.* Let $\mathcal{G}(m,n)$ denote the set of all complete mixed (m,n)-bipartite graphs. Clearly, $|\mathcal{G}(m,n)| = 3^{mn}$, since there are mn edges and three possibilities for each edge. Let $V = \{v_1, \ldots, v_m\}$, $V' = \{v'_1, \ldots, v'_n\}$ be sets of m and n vertices respectively, and for any matrix $\mathbf{B} = [b_{ij}] \in \{0, \frac{1}{2}, 1\}^{m \times n}$, let $E(\mathbf{B})$ denote the set of edges between V and V' given by $E(\mathbf{B}) = \{(v_i \leftarrow v'_j) \mid b_{ij} = 1\} \cup \{(v_i \rightarrow v'_j) \mid b_{ij} = 0\} \cup \{(v_i - v'_j) \mid b_{ij} = \frac{1}{2}\}$. Define the mapping $G : \{0, \frac{1}{2}, 1\}^{m \times n} \rightarrow \mathcal{G}(m,n)$ as follows:

$$G(\mathbf{B}) = (V \cup V', E(\mathbf{B}))\,.$$

Then clearly, G is a bijection that puts the sets $\{0, \frac{1}{2}, 1\}^{m \times n}$ and $\mathcal{G}(m,n)$ into one-to-one correspondence. We show that a matrix $\mathbf{B} \in \{0, \frac{1}{2}, 1\}^{m \times n}$ can be realized as a bipartite rank matrix if and only if the corresponding bipartite graph $G(\mathbf{B}) \in \mathcal{G}(m,n)$ is acyclic.

First suppose $\mathbf{B} = \mathbf{B}_f(\underline{x}, \underline{x}')$ for some $f : \mathcal{X} \rightarrow \mathbb{R}$, $\underline{x} \in \mathcal{X}^m$, $\underline{x}' \in \mathcal{X}^n$, and let if possible $G(\mathbf{B})$ contain a cycle, say

$$(v_{i_1} \leftarrow v'_{j_1} - v_{i_2} - v'_{j_2} - \ \ldots\ - v_{i_k} - v'_{j_k} - v_{i_1})\,.$$

Then, from the definition of a bipartite rank matrix, we get

$$f(x_{i_1}) < f(x'_{j_1}) = f(x_{i_2}) = f(x'_{j_2}) = \ldots = f(x_{i_k}) = f(x'_{j_k}) = f(x_{i_1})\,,$$

which is a contradiction.

To prove the other direction, let $\mathbf{B} \in \{0, \frac{1}{2}, 1\}^{m \times n}$ be such that $G(\mathbf{B})$ is acyclic. Let $G'(\mathbf{B})$ denote the directed graph obtained by collapsing together vertices in $G(\mathbf{B})$ that are connected by an undirected edge. Then it is easily verified that $G'(\mathbf{B})$ does not contain any directed cycles, and therefore there exists a complete order on the vertices of $G'(\mathbf{B})$ that is consistent with the partial order defined by the edges of $G'(\mathbf{B})$ (topological sorting; see, for example, [22], Section 22.4). This implies a unique order on the vertices of $G(\mathbf{B})$ (in which vertices connected by undirected edges are assigned the same position in the ordering). For any $\underline{x} \in \mathcal{X}^m$, $\underline{x}' \in \mathcal{X}^n$, identifying $\underline{x}, \underline{x}'$ with the vertex sets V, V' of $G(\mathbf{B})$ therefore gives a unique

order on $x_1, \ldots, x_m, x'_1, \ldots, x'_n$. It can be verified that defining $f : \mathcal{X} \to \mathbb{R}$ such that it respects this order then gives $\mathbf{B} = \mathbf{B}_f(\underline{x}, \underline{x}')$.

Part 2. Consider again the bijection $G : \{0, \frac{1}{2}, 1\}^{m \times n} \to \mathcal{G}(m, n)$ defined in Part 1 above. We show that a matrix $\mathbf{B} \in \{0, \frac{1}{2}, 1\}^{m \times n}$ does not contain a sub-matrix of any of the forms shown in Table 3.1 if and only if the corresponding bipartite graph $G(\mathbf{B}) \in \mathcal{G}(m, n)$ is acyclic; the desired result then follows by Part 1 of the theorem.

We first note that the condition that $\mathbf{B} \in \{0, \frac{1}{2}, 1\}^{m \times n}$ not contain a sub-matrix of any of the forms shown in Table 3.1 is equivalent to the condition that the corresponding mixed (m, n)-bipartite graph $G(\mathbf{B}) \in \mathcal{G}(m, n)$ not contain any 4-cycles.

Now, to prove the first direction, let $\mathbf{B} \in \{0, \frac{1}{2}, 1\}^{m \times n}$ not contain a sub-matrix of any of the forms shown in Table 3.1. As noted above, this means $G(\mathbf{B})$ does not contain any 4-cycles. Let, if possible, $G(\mathbf{B})$ contain a cycle of length $2k$, say

$$(v_{i_1} \leftarrow v'_{j_1} \text{ — } v_{i_2} \text{ — } v'_{j_2} \text{ — } \ldots \text{ — } v_{i_k} \text{ — } v'_{j_k} \text{ — } v_{i_1}) .$$

Now consider v_{i_1}, v'_{j_2}. Since $G(\mathbf{B})$ is a complete bipartite graph, there must be an edge between these vertices. If $G(\mathbf{B})$ contained the edge $(v_{i_1} \to v'_{j_2})$, it would contain the 4-cycle

$$(v_{i_1} \leftarrow v'_{j_1} \text{ — } v_{i_2} \text{ — } v'_{j_2} \leftarrow v_{i_1}) ,$$

which would be a contradiction. Similarly, if $G(\mathbf{B})$ contained the edge $(v_{i_1} \text{ — } v'_{j_2})$, it would contain the 4-cycle

$$(v_{i_1} \leftarrow v'_{j_1} \text{ — } v_{i_2} \text{ — } v'_{j_2} \text{ — } v_{i_1}) ,$$

which would again be a contradiction. Therefore, $G(\mathbf{B})$ must contain the edge $(v_{i_1} \leftarrow v'_{j_2})$. However, this means $G(\mathbf{B})$ must contain a $2(k-1)$-cycle, namely,

$$(v_{i_1} \leftarrow v'_{j_2} \text{ — } v_{i_3} \text{ — } v'_{j_3} \text{ — } \ldots \text{ — } v_{i_k} \text{ — } v'_{j_k} \text{ — } v_{i_1}) .$$

By a recursive argument, we eventually get that $G(\mathbf{B})$ must contain a 4-cycle, which is a contradiction.

To prove the other direction, let $\mathbf{B} \in \{0, \frac{1}{2}, 1\}^{m \times n}$ be such that $G(\mathbf{B})$ is acyclic. Then it follows trivially that $G(\mathbf{B})$ does not contain a 4-cycle, and therefore, by the above observation, $\mathbf{B}$ does not contain a sub-matrix of any of the forms shown in Table 3.1. □

We discuss further properties of the bipartite rank-shatter coefficients in Section 3.4; we first present below our uniform convergence result in terms of these coefficients.

3.3 Uniform Convergence Bound

The following is the main result of this chapter.

Theorem 3.2 *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$, let $m, n \in \mathbb{N}$, and let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on $\mathcal{X}$. Then for any $\epsilon > 0$,*

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \quad \leq \quad 4 \cdot r(\mathcal{F}, 2m, 2n) \cdot e^{-mn\epsilon^2/8(m+n)} .$$

The proof requires the following result of Devroye [30], which bounds the variance of any function of a sample that satisfies the conditions of McDiarmid's inequality (Theorem 2.1), *i.e.*, for which a single change in the sample has limited effect:

Theorem 3.3 (Devroye, 1991) *Let $X_1, \ldots, X_N$ be independent random variables with X_k taking values in a set A_k for each k. Let $\phi : (A_1 \times \cdots \times A_N) \to \mathbb{R}$ be such that*

$$\sup_{x_i \in A_i, x_k' \in A_k} \left| \phi(x_1, \ldots, x_N) - \phi(x_1, \ldots, x_{k-1}, x_k', x_{k+1}, \ldots, x_N) \right| \quad \leq \quad c_k .$$

Then

$$\mathbf{Var}\Big\{ \phi(X_1, \ldots, X_N) \Big\} \quad \leq \quad \frac{1}{4} \sum_{k=1}^{N} c_k^2 .$$

Proof (of Theorem 3.2)
The proof is adapted from proofs of uniform convergence for the classification error (see, for example, [12, 31]). It consists of four steps.

Step 1. Symmetrization by a ghost sample.

Define $\tilde{S}_+ = (\tilde{x}_1^+, \ldots, \tilde{x}_m^+) \in \mathcal{X}^m$ such that for each $i \in \{1, \ldots, m\}$, $x_i^+, \tilde{x}_i^+$ are independent and identically distributed. Similary, define $\tilde{S}_- = (\tilde{x}_1^-, \ldots, \tilde{x}_n^-) \in \mathcal{X}^n$ such that for each $j \in \{1, \ldots, n\}$, $x_j^-, \tilde{x}_j^-$ are independent and identically distributed. Then for any $\epsilon > 0$ satisfying $mn\epsilon^2/(m+n) \geq 2$, we have

$$\begin{aligned} &\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \epsilon \right\} \\ &\leq \quad 2\mathbf{P}_{S_+, \tilde{S}_+ \sim \mathcal{D}_+^m, S_-, \tilde{S}_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - \hat{R}_{\tilde{S}_+,\tilde{S}_-}(f) \right| \geq \frac{\epsilon}{2} \right\} . \end{aligned}$$

To see this, let $f^*_{S_+,S_-} \in \mathcal{F}$ be a function for which $\left|\hat{R}_{S_+,S_-}(f^*_{S_+.S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})\right| \geq \epsilon$ if such a function exists, and let $f^*_{S_+,S_-}$ be a fixed function in $\mathcal{F}$ otherwise. Then

$$\begin{aligned}
&\mathbf{P}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m,S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_{S_+,S_-}(f) - \hat{R}_{\tilde{S}_+,\tilde{S}_-}(f)\right| \geq \frac{\epsilon}{2}\right\} \\
&\quad\geq\quad \mathbf{P}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m,S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\left|\hat{R}_{S_+,S_-}(f^*_{S_+,S_-}) - \hat{R}_{\tilde{S}_+,\tilde{S}_-}(f^*_{S_+,S_-})\right| \geq \frac{\epsilon}{2}\right\} \\
&\quad\geq\quad \mathbf{P}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m,S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\left\{\left|\hat{R}_{S_+,S_-}(f^*_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})\right| \geq \epsilon\right\}\bigcap\right. \\
&\qquad\qquad\qquad\left.\left\{\left|\hat{R}_{\tilde{S}_+,\tilde{S}_-}(f^*_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})\right| \leq \frac{\epsilon}{2}\right\}\right\} \\
&\quad=\quad \mathbf{E}_{S_+,S_-}\left\{\mathbf{I}_{\{|\hat{R}_{S_+,S_-}(f^*_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})|\geq\epsilon\}} \times\right. \\
&\qquad\qquad\left.\mathbf{P}_{\tilde{S}_+,\tilde{S}_-|S_+,S_-}\left\{\left|\hat{R}_{\tilde{S}_+,\tilde{S}_-}(f^*_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})\right| \leq \frac{\epsilon}{2}\right\}\right\}. \qquad (3.1)
\end{aligned}$$

The conditional probability inside can be bounded using Lemma 1.1 and Chebyshev's inequality (Theorem 2.3):

$$\begin{aligned}
&\mathbf{P}_{\tilde{S}_+,\tilde{S}_-|S_+.S_-}\left\{\left|\hat{R}_{\tilde{S}_+,\tilde{S}_-}(f^*_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})\right| \leq \frac{\epsilon}{2}\right\} \\
&\qquad\geq\quad 1 - \frac{\mathbf{Var}_{\tilde{S}_+,\tilde{S}_-|S_+,S_-}\left\{\hat{R}_{\tilde{S}_+,\tilde{S}_-}(f^*_{S_+,S_-})\right\}}{\epsilon^2/4}.
\end{aligned}$$

Now, by Lemma 2.1 and Theorem 3.3, we have

$$\mathbf{Var}_{\tilde{S}_+,\tilde{S}_-|S_+,S_-}\left\{\hat{R}_{\tilde{S}_+,\tilde{S}_-}(f^*_{S_+,S_-})\right\} \;\leq\; \frac{1}{4}\left(\sum_{i=1}^m\left(\frac{1}{m}\right)^2 + \sum_{j=1}^n\left(\frac{1}{n}\right)^2\right) \;=\; \frac{m+n}{4mn}.$$

This gives

$$\mathbf{P}_{\tilde{S}_+,\tilde{S}_-|S_+,S_-}\left\{\left|\hat{R}_{\tilde{S}_+,\tilde{S}_-}(f^*_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})\right| \leq \frac{\epsilon}{2}\right\} \;\geq\; 1 - \frac{m+n}{mn\epsilon^2} \;\geq\; \frac{1}{2},$$

whenever $mn\epsilon^2/(m+n) \geq 2$. Thus, from Eq. (3.1) and the definition of $f^*_{S_+,S_-}$, we have

$$\begin{aligned}
&\mathbf{P}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m,S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_{S_+,S_-}(f) - \hat{R}_{\tilde{S}_+,\tilde{S}_-}(f)\right| \geq \frac{\epsilon}{2}\right\} \\
&\quad\geq\quad \frac{1}{2}\mathbf{E}_{S_+\sim\mathcal{D}_+^m,S_-\sim\mathcal{D}_-^n}\left\{\mathbf{I}_{\{|\hat{R}_{S_+,S_-}(f^*_{S_+.S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})|\geq\epsilon\}}\right\} \\
&\quad=\quad \frac{1}{2}\mathbf{P}_{S_+\sim\mathcal{D}_+^m,S_-\sim\mathcal{D}_-^n}\left\{\left|\hat{R}_{S_+,S_-}(f^*_{S_+,S_-}) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*_{S_+,S_-})\right| \geq \epsilon\right\}
\end{aligned}$$

$$\geq \quad \frac{1}{2}\mathbf{P}_{S_+\sim\mathcal{D}_+^m, S_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f)\right| \geq \epsilon\right\}.$$

Step 2. Permutations.

Let $\Gamma_{m,n}$ be the set of all permutations of $\{x_1^+,\ldots,x_m^+,x_1^-,\ldots,x_n^-,\tilde{x}_1^+,\ldots,\tilde{x}_m^+,\tilde{x}_1^-,\ldots,\tilde{x}_n^-\}$ that swap x_i^+ with $\tilde{x}_i^+$ and x_j^- with $\tilde{x}_j^-$, for all i in some subset of $\{1,\ldots,m\}$ and all j in some subset of $\{1,\ldots,n\}$. In other words, for all $\sigma\in\Gamma_{m,n}$ and $i\in\{1,\ldots,m\}$, either $\sigma(x_i^+)=x_i^+$, in which case $\sigma(\tilde{x}_i^+)=\tilde{x}_i^+$, or $\sigma(x_i^+)=\tilde{x}_i^+$, in which case $\sigma(\tilde{x}_i^+)=x_i^+$. Similarly, for all $\sigma\in\Gamma_{m,n}$ and $j\in\{1,\ldots,n\}$, either $\sigma(x_j^-)=x_j^-$, in which case $\sigma(\tilde{x}_j^-)=\tilde{x}_j^-$, or $\sigma(x_j^-)=\tilde{x}_j^-$, in which case $\sigma(\tilde{x}_j^-)=x_j^-$. Denote

$$\sigma(S_+) \;=\; (\sigma(x_1^+),\ldots,\sigma(x_m^+)), \qquad \sigma(S_-) \;=\; (\sigma(x_1^-),\ldots,\sigma(x_n^-))$$

$$\sigma(\tilde{S}_+) \;=\; (\sigma(\tilde{x}_1^+),\ldots,\sigma(\tilde{x}_m^+)), \qquad \sigma(\tilde{S}_-) \;=\; (\sigma(\tilde{x}_1^-),\ldots,\sigma(\tilde{x}_n^-)).$$

Now, define

$$\beta_f(S_+,S_-,\tilde{S}_+,\tilde{S}_-) \;\equiv\; \frac{1}{mn}\sum_{i=1}^m\sum_{j=1}^n\left(\ell_{\mathrm{rank}}(f,x_i^+,x_j^-) - \ell_{\mathrm{rank}}(f,\tilde{x}_i^+,\tilde{x}_j^-)\right).$$

Then clearly, since $x_i^+,\tilde{x}_i^+$ and $x_j^-,\tilde{x}_j^-$ are independent and identically distributed for each i,j, for any $\sigma\in\Gamma_{m,n}$ we have that the distribution of

$$\sup_{f\in\mathcal{F}}\left|\beta_f\left(S_+,S_-,\tilde{S}_+,\tilde{S}_-\right)\right|$$

is the same as the distribution of

$$\sup_{f\in\mathcal{F}}\left|\beta_f\Big(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\Big)\right|.$$

Therefore, using $\mathcal{U}(D)$ to denote the uniform distribution over a discrete set D, we have the following:

$$\begin{aligned}
&\mathbf{P}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m, S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_{S_+,S_-}(f) - \hat{R}_{\tilde{S}_+,\tilde{S}_-}(f)\right| \geq \frac{\epsilon}{2}\right\}\\
&\quad= \quad \mathbf{P}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m, S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\beta_f\left(S_+,S_-,\tilde{S}_+,\tilde{S}_-\right)\right| \geq \frac{\epsilon}{2}\right\}\\
&\quad= \quad \frac{1}{|\Gamma_{m,n}|}\sum_{\sigma\in\Gamma_{m,n}}\mathbf{P}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m, S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right| \geq \frac{\epsilon}{2}\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{|\Gamma_{m,n}|}\sum_{\sigma\in\Gamma_{m,n}} \mathbf{E}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m,S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\mathbf{I}_{\left\{\sup_{f\in\mathcal{F}}\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right|\geq\frac{\epsilon}{2}\right\}}\right\}\\
&= \mathbf{E}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m,S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\frac{1}{|\Gamma_{m,n}|}\sum_{\sigma\in\Gamma_{m,n}}\mathbf{I}_{\left\{\sup_{f\in\mathcal{F}}\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right|\geq\frac{\epsilon}{2}\right\}}\right\}\\
&= \mathbf{E}_{S_+,\tilde{S}_+\sim\mathcal{D}_+^m,S_-,\tilde{S}_-\sim\mathcal{D}_-^n}\left\{\mathbf{P}_{\sigma\sim\mathcal{U}(\Gamma_{m,n})}\left\{\sup_{f\in\mathcal{F}}\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right|\geq\frac{\epsilon}{2}\right\}\right\}\\
&\leq \max_{S_+,\tilde{S}_+\in\mathcal{X}^m,S_-,\tilde{S}_-\in\mathcal{X}^n}\mathbf{P}_{\sigma\sim\mathcal{U}(\Gamma_{m,n})}\left\{\sup_{f\in\mathcal{F}}\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right|\geq\frac{\epsilon}{2}\right\}.
\end{aligned}
$$

Step 3. Reduction to a finite class.

We wish to bound the quantity on the right hand side above. From the definition of bipartite rank matrices (Definition 3.1), it follows that for any $S_+,\tilde{S}_+\in\mathcal{X}^m$, $S_-,\tilde{S}_-\in\mathcal{X}^n$, as f ranges over $\mathcal{F}$, the number of different random variables

$$
\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}^+),\sigma(\tilde{S}^-)\right)\right|
$$

is at most the number of different bipartite rank matrices $\mathbf{B}_f\Big((S_+,\tilde{S}_+),(S_-,\tilde{S}_-)\Big)$ that can be realized by functions in $\mathcal{F}$. This number, by definition, cannot exceed $r(\mathcal{F},2m,2n)$ (see the definition of bipartite rank-shatter coefficients, Definition 3.2). Therefore, the supremum in the above probability is a maximum of at most $r(\mathcal{F},2m,2n)$ random variables. Thus, by the union bound, we get for any $S_+,\tilde{S}_+\in\mathcal{X}^m$, $S_-,\tilde{S}_-\in\mathcal{X}^n$,

$$
\begin{aligned}
&\mathbf{P}_{\sigma\sim\mathcal{U}(\Gamma_{m,n})}\left\{\sup_{f\in\mathcal{F}}\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right|\geq\frac{\epsilon}{2}\right\}\\
&\quad\leq\ r(\mathcal{F},2m,2n)\cdot\sup_{f\in\mathcal{F}}\mathbf{P}_{\sigma\sim\mathcal{U}(\Gamma_{m,n})}\left\{\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right|\geq\frac{\epsilon}{2}\right\}.
\end{aligned}
$$

Step 4. McDiarmid's inequality.

Notice that for any $S_+,\tilde{S}_+\in\mathcal{X}^m$, $S_-,\tilde{S}_-\in\mathcal{X}^n$, we can write

$$
\begin{aligned}
&\mathbf{P}_{\sigma\sim\mathcal{U}(\Gamma_{m,n})}\left\{\left|\beta_f\left(\sigma(S_+),\sigma(S_-),\sigma(\tilde{S}_+),\sigma(\tilde{S}_-)\right)\right|\geq\frac{\epsilon}{2}\right\}\\
&\quad=\ \mathbf{P}_{W_+\sim\mathcal{U}\left(\prod_{i=1}^m\{x_i^+,\tilde{x}_i^+\}\right),W_-\sim\mathcal{U}\left(\prod_{j=1}^n\{x_j^-,\tilde{x}_j^-\}\right)}\left\{\left|\beta_f\left(W_+,W_-,\tilde{W}_+,\tilde{W}_-\right)\right|\geq\frac{\epsilon}{2}\right\},
\end{aligned}
$$

where $W_+=(w_1^+,\ldots,w_m^+)$, $W_-=(w_1^-,\ldots,w_n^-)$, $\tilde{W}_+=(\tilde{w}_1^+,\ldots,\tilde{w}_m^+)$, $\tilde{W}_-=(\tilde{w}_1^-,\ldots,\tilde{w}_n^-)$, and

$$
\tilde{w}_i^+=\begin{cases}\tilde{x}_i^+, & \text{if } w_i^+=x_i^+\\ x_i^+, & \text{if } w_i^+=\tilde{x}_i^+\end{cases},\qquad \tilde{w}_j^-=\begin{cases}\tilde{x}_j^-, & \text{if } w_j^-=x_j^-\\ x_j^-, & \text{if } w_j^-=\tilde{x}_j^-\end{cases}.
$$

Now, for any $f \in \mathcal{F}$,

$$\begin{aligned}
&\mathbf{E}_{W_+ \sim \mathcal{U}\left(\prod_{i=1}^m \{x_i^+, \tilde{x}_i^+\}\right), W_- \sim \mathcal{U}\left(\prod_{j=1}^n \{x_j^-, \tilde{x}_j^-\}\right)} \left\{ \beta_f \left(W_+, W_-, \tilde{W}_+, \tilde{W}_- \right) \right\} \\
&= \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \mathbf{E}_{w_i^+ \sim \mathcal{U}\left(\{x_i^+, \tilde{x}_i^+\}\right), w_j^- \sim \mathcal{U}\left(\{x_j^-, \tilde{x}_j^-\}\right)} \left\{ \ell_{\text{rank}}(f, w_i^+, w_j^-) - \ell_{\text{rank}}(f, \tilde{w}_i^+, \tilde{w}_j^-) \right\} \\
&= \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \frac{1}{4} \Big\{ \left(\ell_{\text{rank}}(f, x_i^+, x_j^-) - \ell_{\text{rank}}(f, \tilde{x}_i^+, \tilde{x}_j^-) \right) + \left(\ell_{\text{rank}}(f, x_i^+, \tilde{x}_j^-) - \ell_{\text{rank}}(f, \tilde{x}_i^+, x_j^-) \right) + \\
&\qquad \left(\ell_{\text{rank}}(f, \tilde{x}_i^+, x_j^-) - \ell_{\text{rank}}(f, x_i^+, \tilde{x}_j^-) \right) + \left(\ell_{\text{rank}}(f, \tilde{x}_i^+, \tilde{x}_j^-) - \ell_{\text{rank}}(f, x_i^+, x_j^-) \right) \Big\} \\
&= 0 \,.
\end{aligned}$$

Also, it can be verified that for any $f \in \mathcal{F}$, a change in the value of a single random variable w_i^+ can bring a change of at most $2/m$ in the value of

$$\beta_f \left(W_+, W_-, \tilde{W}_+, \tilde{W}_- \right) ,$$

and that a change in the value of a single random variable w_j^- can bring a change of at most $2/n$. Therefore, by McDiarmid's inequality (Theorem 2.1), it follows that for any $f \in \mathcal{F}$,

$$\begin{aligned}
&\mathbf{P}_{W_+ \sim \mathcal{U}\left(\prod_{i=1}^m \{x_i^+, \tilde{x}_i^+\}\right), W_- \sim \mathcal{U}\left(\prod_{j=1}^n \{x_j^-, \tilde{x}_j^-\}\right)} \left\{ \left| \beta_f \left(W_+, W_-, \tilde{W}_+, \tilde{W}_- \right) \right| \geq \frac{\epsilon}{2} \right\} \\
&\leq 2e^{-2\epsilon^2/4(m(\frac{2}{m})^2 + n(\frac{2}{n})^2)} \\
&= 2e^{-mn\epsilon^2/8(m+n)} \,.
\end{aligned}$$

Putting everything together, we get that

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+, S_-}(f) - R_{\mathcal{D}_+, \mathcal{D}_-}(f) \right| \geq \epsilon \right\} \leq 4 \cdot r(\mathcal{F}, 2m, 2n) \cdot e^{-mn\epsilon^2/8(m+n)} ,$$

for $mn\epsilon^2/(m+n) \geq 2$. In the other case, *i.e.*, for $mn\epsilon^2/(m+n) < 2$, the bound is greater than one and therefore holds trivially. □

From Theorem 3.2, we can derive a confidence interval interpretation of the bound as follows:

Corollary 3.1 *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$, let $m, n \in \mathbb{N}$, and let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on $\mathcal{X}$. Then for any $0 < \delta \leq 1$,*

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+, S_-}(f) - R_{\mathcal{D}_+, \mathcal{D}_-}(f) \right| \geq \sqrt{\frac{8(m+n)\left(\ln r(\mathcal{F}, 2m, 2n) + \ln\left(\frac{4}{\delta}\right) \right)}{mn}} \right\} \leq \delta \,.$$

Proof This follows directly from Theorem 3.2 by setting $4 \cdot r(\mathcal{F}, 2m, 2n) \cdot e^{-mn\epsilon^2/8(m+n)} = \delta$ and solving for ϵ. □

3.4 Properties of Bipartite Rank-Shatter Coefficients

As discussed in Section 3.2, we have $r(\mathcal{F}, m, n) \leq \psi(m, n)$, where $\psi(m, n)$ is the number of matrices in $\{0, \frac{1}{2}, 1\}^{m \times n}$ that can be realized as a bipartite rank matrix. The number $\psi(m, n)$ is strictly smaller than 3^{mn}; indeed, $\psi(m, n) = O(e^{(m+n)(\ln(m+n)+1)})$. (To see this, note that the number of distinct bipartite rank matrices of size $m \times n$ is bounded above by the total number of permutations of $(m+n)$ objects, allowing for objects to be placed at the same position. This number is equal to $(m+n)!\, 2^{(m+n-1)} = O(e^{(m+n)(\ln(m+n)+1)})$.) Nevertheless, $\psi(m, n)$ is still very large; in particular, $\psi(m, n) \geq 3^{\max(m,n)}$. (To see this, note that choosing any column vector in $\{0, \frac{1}{2}, 1\}^m$ and replicating it along the n columns or choosing any row vector in $\{0, \frac{1}{2}, 1\}^n$ and replicating it along the m rows results in a matrix that does not contain a sub-matrix of any of the forms shown in Table 3.1. The conclusion then follows from Theorem 3.1 (Part 2).)

For the bound of Theorem 3.2 to be meaningful, one needs an upper bound on $r(\mathcal{F}, m, n)$ that is at least slightly smaller than $e^{mn/8(m+n)}$. Below we provide one method for deriving upper bounds on $r(\mathcal{F}, m, n)$; we extend slightly the standard VC-dimension related shatter coefficients studied in binary classification to $\{-1, 0, 1\}$-valued function classes, and then derive an upper bound on the bipartite rank-shatter coefficients $r(\mathcal{F}, m, n)$ of a class of ranking functions $\mathcal{F}$ in terms of the shatter coefficients of a class of $\{-1, 0, 1\}$-valued functions derived from $\mathcal{F}$.

Definition 3.3 (Shatter coefficient) *Let $\mathcal{H}$ be a class of $\{-1, 0, 1\}$-valued functions on $\mathcal{X}$ and let $N \in \mathbb{N}$. Define the N*th shatter coefficient *of $\mathcal{H}$, denoted by $s(\mathcal{H}, N)$, as follows:*

$$s(\mathcal{H}, N) \quad = \quad \max_{\underline{x} \in \mathcal{X}^N} \left| \left\{ (h(x_1), \ldots, h(x_N)) \;\mid h \in \mathcal{H} \right\} \right|.$$

Clearly, $s(\mathcal{H}, N) \leq 3^N$ for all N. Next we define a series of $\{-1, 0, 1\}$-valued function classes derived from a given ranking function class. Only the second function class is used in this section; the other two are needed in Section 3.5. Note that we take, for $u \in \mathbb{R}$,

$$\text{sign}(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{if } u = 0 \\ -1 & \text{if } u < 0 \end{cases}.$$

Definition 3.4 (Function classes) *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$. Define*

the following classes of $\{-1,0,1\}$*-valued functions derived from* $\mathcal{F}$*:*

$$1.\quad \bar{\mathcal{F}} = \{\bar{f}:\mathcal{X}\to\{-1,0,1\} \mid \bar{f}(x)=\operatorname{sign}(f(x)) \text{ for some } f\in\mathcal{F}\} \tag{3.2}$$

$$2.\quad \tilde{\mathcal{F}} = \{\tilde{f}:\mathcal{X}\times\mathcal{X}\to\{-1,0,1\} \mid \tilde{f}(x,x')=\operatorname{sign}(f(x)-f(x')) \text{ for some } f\in\mathcal{F}\} \tag{3.3}$$

$$3.\quad \check{\mathcal{F}} = \{\check{f}_z:\mathcal{X}\to\{-1,0,1\} \mid \check{f}_z(x)=\operatorname{sign}(f(x)-f(z)) \text{ for some } f\in\mathcal{F}, z\in\mathcal{X}\} \tag{3.4}$$

The following result gives an upper bound on the bipartite rank-shatter coefficients of a class of ranking functions $\mathcal{F}$ in terms of the shatter coefficients of $\tilde{\mathcal{F}}$:

Theorem 3.4 *Let* $\mathcal{F}$ *be a class of real-valued functions on* $\mathcal{X}$*, and let* $\tilde{\mathcal{F}}$ *be the class of* $\{-1,0,1\}$*-valued functions on* $\mathcal{X}\times\mathcal{X}$ *defined by Eq. (3.3). Then for all* $m,n\in\mathbb{N}$*,*

$$r(\mathcal{F},m,n) \;\leq\; s(\tilde{\mathcal{F}},mn)\,.$$

Proof For any $m,n\in\mathbb{N}$, we have[1]

$$\begin{aligned}
r(\mathcal{F},m,n) &= \max_{\underline{x}\in\mathcal{X}^m,\underline{x}'\in\mathcal{X}^n}\left|\left\{\left[\mathbf{I}_{\{f(x_i)<f(x'_j)\}}+\frac{1}{2}\mathbf{I}_{\{f(x_i)=f(x'_j)\}}\right] \;\middle|\; f\in\mathcal{F}\right\}\right| \\
&= \max_{\underline{x}\in\mathcal{X}^m,\underline{x}'\in\mathcal{X}^n}\left|\left\{\left[\mathbf{I}_{\{\tilde{f}(x_i,x'_j)=-1\}}+\frac{1}{2}\mathbf{I}_{\{\tilde{f}(x_i,x'_j)=0\}}\right] \;\middle|\; \tilde{f}\in\tilde{\mathcal{F}}\right\}\right| \\
&= \max_{\underline{x}\in\mathcal{X}^m,\underline{x}'\in\mathcal{X}^n}\left|\left\{\left[\tilde{f}(x_i,x'_j)\right] \;\middle|\; \tilde{f}\in\tilde{\mathcal{F}}\right\}\right| \\
&\leq \max_{\mathbf{X},\mathbf{X}'\in\mathcal{X}^{m\times n}}\left|\left\{\left[\tilde{f}(x_{ij},x'_{ij})\right] \;\middle|\; \tilde{f}\in\tilde{\mathcal{F}}\right\}\right| \\
&= \max_{\underline{x},\underline{x}'\in\mathcal{X}^{mn}}\left|\left\{\left(\tilde{f}(x_1,x'_1),\ldots,\tilde{f}(x_{mn},x'_{mn})\right) \;\middle|\; \tilde{f}\in\tilde{\mathcal{F}}\right\}\right| \\
&= s(\tilde{\mathcal{F}},mn)\,.
\end{aligned}$$

□

Below we make use of the above result to derive polynomial upper bounds on the bipartite rank-shatter coefficients for linear and higher-order polynomial ranking functions. We note that the same method can be used to establish similar upper bounds for other algebraically well-behaved function classes.

Lemma 3.1 *For* $d\in\mathbb{N}$*, let* $\mathcal{F}_{\text{lin}(d)}$ *denote the class of linear ranking functions on* $\mathbb{R}^d$*:*

$$\mathcal{F}_{\text{lin}(d)} \;=\; \{f:\mathbb{R}^d\to\mathbb{R} \mid f(\mathbf{x})=\mathbf{w}\cdot\mathbf{x}+b \text{ for some } \mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}\}\,.$$

[1] We use the notation $[a_{ij}]$ to denote a matrix whose $(i,j)^{\text{th}}$ element is a_{ij}. The dimensions of such a matrix should be clear from context.

Then for all $N \in \mathbb{N}$,

$$s(\tilde{\mathcal{F}}_{\mathrm{lin}(d)}, N) \;\leq\; \left(\frac{2eN}{d}\right)^d .$$

Proof We have,

$$\tilde{\mathcal{F}}_{\mathrm{lin}(d)} \;=\; \{\tilde{f} : \mathbb{R}^d \times \mathbb{R}^d \to \{-1, 0, 1\} \mid \tilde{f}(\mathbf{x}, \mathbf{x}') = \mathrm{sign}(\mathbf{w} \cdot (\mathbf{x} - \mathbf{x}')) \text{ for some } \mathbf{w} \in \mathbb{R}^d\} .$$

Let $(\mathbf{x}_1, \mathbf{x}'_1), \ldots, (\mathbf{x}_N, \mathbf{x}'_N)$ be any N points in $\mathbb{R}^d \times \mathbb{R}^d$, and consider the 'dual' weight space corresponding to $\mathbf{w} \in \mathbb{R}^d$. Each point $(\mathbf{x}_i, \mathbf{x}'_i)$ defines a hyperplane $(\mathbf{x}_i - \mathbf{x}'_i)$ in this space; the N points thus give rise to an arrangement of N hyperplanes in $\mathbb{R}^d$. It is easily seen that the number of sign patterns $(\tilde{f}(\mathbf{x}_1, \mathbf{x}'_1), \ldots, \tilde{f}(\mathbf{x}_N, \mathbf{x}'_N))$ that can be realized by functions $\tilde{f} \in \tilde{\mathcal{F}}_{\mathrm{lin}(d)}$ is equal to the total number of faces of this arrangement [72], which is at most [18]

$$\sum_{k=0}^{d} \sum_{i=d-k}^{d} \binom{i}{d-k} \binom{N}{i} \;=\; \sum_{i=0}^{d} 2^i \binom{N}{i} \;\leq\; \left(\frac{2eN}{d}\right)^d .$$

Since the N points were arbitrary, the result follows. □

Theorem 3.5 *For $d \in \mathbb{N}$, let $\mathcal{F}_{\mathrm{lin}(d)}$ denote the class of linear ranking functions on $\mathbb{R}^d$ (defined in Lemma 3.1 above). Then for all $m, n \in \mathbb{N}$,*

$$r(\mathcal{F}_{\mathrm{lin}(d)}, m, n) \;\leq\; \left(\frac{2emn}{d}\right)^d .$$

Proof This follows immediately from Lemma 3.1 and Theorem 3.4. □

Lemma 3.2 *For $d, q \in \mathbb{N}$, let $\mathcal{F}_{\mathrm{poly}(d,q)}$ denote the class of polynomial ranking functions on $\mathbb{R}^d$ with degree less than or equal to q. Then for all $N \in \mathbb{N}$,*

$$s(\tilde{\mathcal{F}}_{\mathrm{poly}(d,q)}, N) \;\leq\; \left(\frac{2eN}{C(d,q)}\right)^{C(d,q)} ,$$

where

$$C(d, q) \;=\; \sum_{i=1}^{q} \left(\binom{d}{i} \sum_{j=1}^{q} \binom{j-1}{i-1} \right). \tag{3.5}$$

Proof We have,

$$\tilde{\mathcal{F}}_{\mathrm{poly}(d,q)} \;=\; \{\tilde{f} : \mathbb{R}^d \times \mathbb{R}^d \to \{-1, 0, 1\} \mid \tilde{f}(\mathbf{x}, \mathbf{x}') = \mathrm{sign}(f(\mathbf{x}) - f(\mathbf{x}')) \text{ for some } f \in \mathcal{F}_{\mathrm{poly}(d,q)}\} .$$

Let $(\mathbf{x}_1, \mathbf{x}'_1), \ldots, (\mathbf{x}_N, \mathbf{x}'_N)$ be any N points in $\mathbb{R}^d \times \mathbb{R}^d$. For any $f \in \mathcal{F}_{\text{poly}(d,q)}$, $(f(\mathbf{x}) - f(\mathbf{x}'))$ is a linear combination of $C(d,q)$ basis functions of the form $(g_k(\mathbf{x}) - g_k(\mathbf{x}'))$, $1 \leq k \leq C(d,q)$, each $g_k(\mathbf{x})$ being a product of 1 to q components of $\mathbf{x}$. Denote $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_{C(d,q)}(\mathbf{x})) \in \mathbb{R}^{C(d,q)}$. Then each point $(\mathbf{x}_i, \mathbf{x}'_i)$ defines a hyperplane $(\mathbf{g}(\mathbf{x}_i) - \mathbf{g}(\mathbf{x}'_i))$ in $\mathbb{R}^{C(d,q)}$; the N points thus give rise to an arrangement of N hyperplanes in $\mathbb{R}^{C(d,q)}$. It is easily seen that the number of sign patterns $(\tilde{f}(\mathbf{x}_1, \mathbf{x}'_1), \ldots, \tilde{f}(\mathbf{x}_N, \mathbf{x}'_N))$ that can be realized by functions $\tilde{f} \in \tilde{\mathcal{F}}_{\text{poly}(d,q)}$ is equal to the total number of faces of this arrangement [72], which is at most [18]

$$\left(\frac{2eN}{C(d,q)}\right)^{C(d,q)}.$$

Since the N points were arbitrary, the result follows. □

Theorem 3.6 *For $d, q \in \mathbb{N}$, let $\mathcal{F}_{\text{poly}(d,q)}$ denote the class of polynomial ranking functions on $\mathbb{R}^d$ with degree less than or equal to q. Then for all $m, n \in \mathbb{N}$,*

$$r(\mathcal{F}_{\text{poly}(d,q)}, m, n) \;\leq\; \left(\frac{2emn}{C(d,q)}\right)^{C(d,q)},$$

where $C(d,q)$ is as defined in Eq. (3.5).

Proof This follows immediately from Lemma 3.2 and Theorem 3.4. □

3.5 Comparison with Uniform Convergence Bound of Freund et al.

Freund et al. [37] recently derived a uniform convergence bound for the bipartite ranking error[2]. Although the result of Freund et al. is given only for function classes considered by their RankBoost algorithm, their technique is generally applicable. We state their result below, using our notation, for the general case (*i.e.*, function classes not restricted to those considered by RankBoost), and then offer a comparison of our bound with theirs. As in [37], the result is given in the form of a confidence interval.[3]

[2] As mentioned in Chapter 1, the ranking error defined by Freund et al. does not account for ties.

[3] The result of Freund et al. was stated in terms of the VC dimension, but the basic result can be stated in terms of shatter coefficients. Due to our ranking error definition which accounts for ties, the standard shatter coefficients are replaced here with the extended shatter coefficients defined above for $\{-1, 0, 1\}$-valued function classes.

Theorem 3.7 (Generalization of [37, Theorem 3]) *Let $\mathcal{F}$ be a class of real-valued functions on $\mathcal{X}$, let $m, n \in \mathbb{N}$, and let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on $\mathcal{X}$. Then for any $0 < \delta \leq 1$,*

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq 2\sqrt{\frac{\ln s(\check{\mathcal{F}}, 2m) + \ln\left(\frac{12}{\delta}\right)}{m}} + 2\sqrt{\frac{\ln s(\check{\mathcal{F}}, 2n) + \ln\left(\frac{12}{\delta}\right)}{n}} \right\} \leq \delta ,$$

where $\check{\mathcal{F}}$ is the class of $\{-1, 0, 1\}$-valued functions on $\mathcal{X}$ defined by Eq. (3.4).

The proof follows that of Freund et al. [37]; we give details for completeness. We shall need to extend the notion of classification error to $\{-1, 0, 1\}$-valued functions. With some overloading of notation, let the classification loss of a function $h : \mathcal{X} \to \{-1, 0, 1\}$ on $(x, y) \in \mathcal{X} \times \{-1, 1\}$ be defined as

$$\ell_{\text{class}}(h, x, y) = \mathbf{I}_{\{h(\mathbf{x}) \neq 0\}} \mathbf{I}_{\{h(\mathbf{x}) \neq y\}} + \frac{1}{2} \mathbf{I}_{\{h(\mathbf{x}) = 0\}} . \tag{3.6}$$

Let the expected classification error of a function $h : \mathcal{X} \to \{-1, 0, 1\}$ with respect to a distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, 1\}$ be defined as

$$L_{\mathcal{D}}(h) = \mathbf{E}_{(x,y) \sim \mathcal{D}} \left\{ \ell_{class}(h, x, y) \right\} . \tag{3.7}$$

Similarly, let the empirical classification error of a function $h : \mathcal{X} \to \{-1, 0, 1\}$ with respect to a sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \{-1, 1\})^m$ be defined as

$$\hat{L}_S(h) = \frac{1}{m} \sum_{i=1}^{m} \left\{ \ell_{class}(h, x_i, y_i) \right\} . \tag{3.8}$$

Then, following the proof of a similar result given in [101] for binary-valued functions, it can be shown that if $\mathcal{H}$ is a class of $\{-1, 0, 1\}$-valued functions on $\mathcal{X}$ and $M \in \mathbb{N}$, then for any $\epsilon > 0$,

$$\mathbf{P}_{S \sim \mathcal{D}^M} \left\{ \sup_{h \in \mathcal{H}} \left| \hat{L}_S(h) - L_{\mathcal{D}}(h) \right| \geq \epsilon \right\} \leq 6 s(\mathcal{H}, 2M) e^{-M\epsilon^2/4} . \tag{3.9}$$

Proof (of Theorem 3.7)
For any $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, we have for all $f \in \mathcal{F}$,

$$\left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right|$$

$$
\begin{aligned}
&= \left| \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell_{\text{rank}}(f, x_i^+, x_j^-) - \mathbf{E}_{x^+ \sim \mathcal{D}_+, x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x^+, x^-) \right\} \right| \\
&= \left| \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell_{\text{rank}}(f, x_i^+, x_j^-) - \frac{1}{m} \sum_{i=1}^{m} \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x_i^+, x^-) \right\} \right. \\
&\qquad \left. + \frac{1}{m} \sum_{i=1}^{m} \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x_i^+, x^-) \right\} - \mathbf{E}_{x^+ \sim \mathcal{D}_+, x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x^+, x^-) \right\} \right| \\
&= \left| \frac{1}{m} \sum_{i=1}^{m} \left(\frac{1}{n} \sum_{j=1}^{n} \ell_{\text{rank}}(f, x_i^+, x_j^-) - \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x_i^+, x^-) \right\} \right) \right. \\
&\qquad \left. + \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{rank}}(f, x_i^+, x^-) - \mathbf{E}_{x^+ \sim \mathcal{D}_+} \left\{ \ell_{\text{rank}}(f, x^+, x^-) \right\} \right\} \right| \\
&\leq \frac{1}{m} \sum_{i=1}^{m} \left| \frac{1}{n} \sum_{j=1}^{n} \ell_{\text{rank}}(f, x_i^+, x_j^-) - \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f, x_i^+, x^-) \right\} \right| \\
&\qquad + \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \left| \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{rank}}(f, x_i^+, x^-) - \mathbf{E}_{x^+ \sim \mathcal{D}_+} \left\{ \ell_{\text{rank}}(f, x^+, x^-) \right\} \right| \right\} \\
&\leq \sup_{f' \in \mathcal{F}, z \in \mathcal{X}} \left| \frac{1}{n} \sum_{j=1}^{n} \ell_{\text{rank}}(f', z, x_j^-) - \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{rank}}(f', z, x^-) \right\} \right| \\
&\qquad + \sup_{f' \in \mathcal{F}, z \in \mathcal{X}} \left| \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{rank}}(f', x_i^+, z) - \mathbf{E}_{x^+ \sim \mathcal{D}_+} \left\{ \ell_{\text{rank}}(f', x^+, z) \right\} \right| \\
&= \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \frac{1}{n} \sum_{j=1}^{n} \ell_{\text{class}}(\check{f}_z, x_j^-, -1) - \mathbf{E}_{x^- \sim \mathcal{D}_-} \left\{ \ell_{\text{class}}(\check{f}_z, x^-, -1) \right\} \right| \\
&\qquad + \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \frac{1}{m} \sum_{i=1}^{m} \ell_{\text{class}}(\check{f}_z, x_i^+, 1) - \mathbf{E}_{x^+ \sim \mathcal{D}_+} \left\{ \ell_{\text{class}}(\check{f}_z, x^+, 1) \right\} \right| \\
&= \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \hat{L}_{S_-^*}(\check{f}_z) - L_{\mathcal{D}_-^*}(\check{f}_z) \right| + \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \hat{L}_{S_+^*}(\check{f}_z) - L_{\mathcal{D}_+^*}(\check{f}_z) \right| ,
\end{aligned}
$$

where $S_+^* = ((x_1^+, 1), \ldots, (x_m^+, 1))$, $S_-^* = ((x_1^-, -1), \ldots, (x_n^-, -1))$, $\mathcal{D}_+^*$ is the distribution over $\mathcal{X} \times \{-1, 1\}$ that is the product of $\mathcal{D}_+$ on $\mathcal{X}$ with a distribution on $\{-1, 1\}$ that has all its mass on 1, and $\mathcal{D}_-^*$ is the distribution over $\mathcal{X} \times \{-1, 1\}$ that is the product of $\mathcal{D}_-$ on $\mathcal{X}$ with a distribution on $\{-1, 1\}$ that has all its mass on -1. This gives for all $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$,

$$
\sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+, S_-}(f) - R_{\mathcal{D}_+, \mathcal{D}_-}(f) \right|
$$

$$\leq \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \hat{L}_{S_-^*}(\check{f}_z) - L_{\mathcal{D}_-^*}(\check{f}_z) \right| + \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \hat{L}_{S_+^*}(\check{f}_z) - L_{\mathcal{D}_+^*}(\check{f}_z) \right| . \tag{3.10}$$

Now, from the confidence interval interpretation of the result given in Eq. (3.9), we have

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m} \left\{ \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \hat{L}_{S_+^*}(\check{f}_z) - L_{\mathcal{D}_+^*}(\check{f}_z) \right| \geq 2\sqrt{\frac{\ln s(\check{\mathcal{F}}, 2m) + \ln\left(\frac{12}{\delta}\right)}{m}} \right\} \leq \frac{\delta}{2}, \tag{3.11}$$

$$\mathbf{P}_{S_- \sim \mathcal{D}_-^n} \left\{ \sup_{\check{f}_z \in \check{\mathcal{F}}} \left| \hat{L}_{S_-^*}(\check{f}_z) - L_{\mathcal{D}_-^*}(\check{f}_z) \right| \geq 2\sqrt{\frac{\ln s(\check{\mathcal{F}}, 2n) + \ln\left(\frac{12}{\delta}\right)}{n}} \right\} \leq \frac{\delta}{2}. \tag{3.12}$$

Combining Eqs. (3.10–3.12) gives the desired result. □

We now compare the uniform convergence bound derived in Section 3.3 with that of Freund et al. for a simple function class for which the quantities involved in both bounds (namely, $r(\mathcal{F}, 2m, 2n)$ and $s(\check{\mathcal{F}}, 2m), s(\check{\mathcal{F}}, 2n)$) can be characterized exactly. Specifically, consider the function class $\mathcal{F}_{\text{lin}(1)}$ of linear ranking functions on $\mathbb{R}$, given by

$$\mathcal{F}_{\text{lin}(1)} = \{f : \mathbb{R} \to \mathbb{R} \mid f(x) = wx + b \text{ for some } w \in \mathbb{R}, b \in \mathbb{R}\}.$$

Although $\mathcal{F}_{\text{lin}(1)}$ is an infinite function class, it is easy to verify that $r(\mathcal{F}_{\text{lin}(1)}, m, n) = 3$ for all $m, n \in \mathbb{N}$. (To see this, note that for any set of $m+n$ distinct points in $\mathbb{R}$, one can obtain exactly three different ranking behaviours with functions in $\mathcal{F}_{\text{lin}(1)}$: one by setting $w > 0$, another by setting $w < 0$, and the third by setting $w = 0$.) On the other hand, $s(\check{\mathcal{F}}_{\text{lin}(1)}, N) = 4N + 1$ for all $N \geq 2$, since $\check{\mathcal{F}}_{\text{lin}(1)} = \bar{\mathcal{F}}_{\text{lin}(1)}$ (see Eq. (3.2)) and, as is easily verified, the number of sign patterns on $N \geq 2$ distinct points in $\mathbb{R}$ that can be realized by functions in $\bar{\mathcal{F}}_{\text{lin}(1)}$ is $4N + 1$. We thus get from our result (Corollary 3.1) that

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}_{\text{lin}(1)}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq \sqrt{\frac{8(m+n)\left(\ln 3 + \ln\left(\frac{4}{\delta}\right)\right)}{mn}} \right\} \leq \delta,$$

and from the result of Freund et al. (Theorem 3.7) that

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}_{\text{lin}(1)}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| \geq 2\sqrt{\frac{\ln(8m+1) + \ln\left(\frac{12}{\delta}\right)}{m}} + 2\sqrt{\frac{\ln(8n+1) + \ln\left(\frac{12}{\delta}\right)}{n}} \right\} \leq \delta.$$

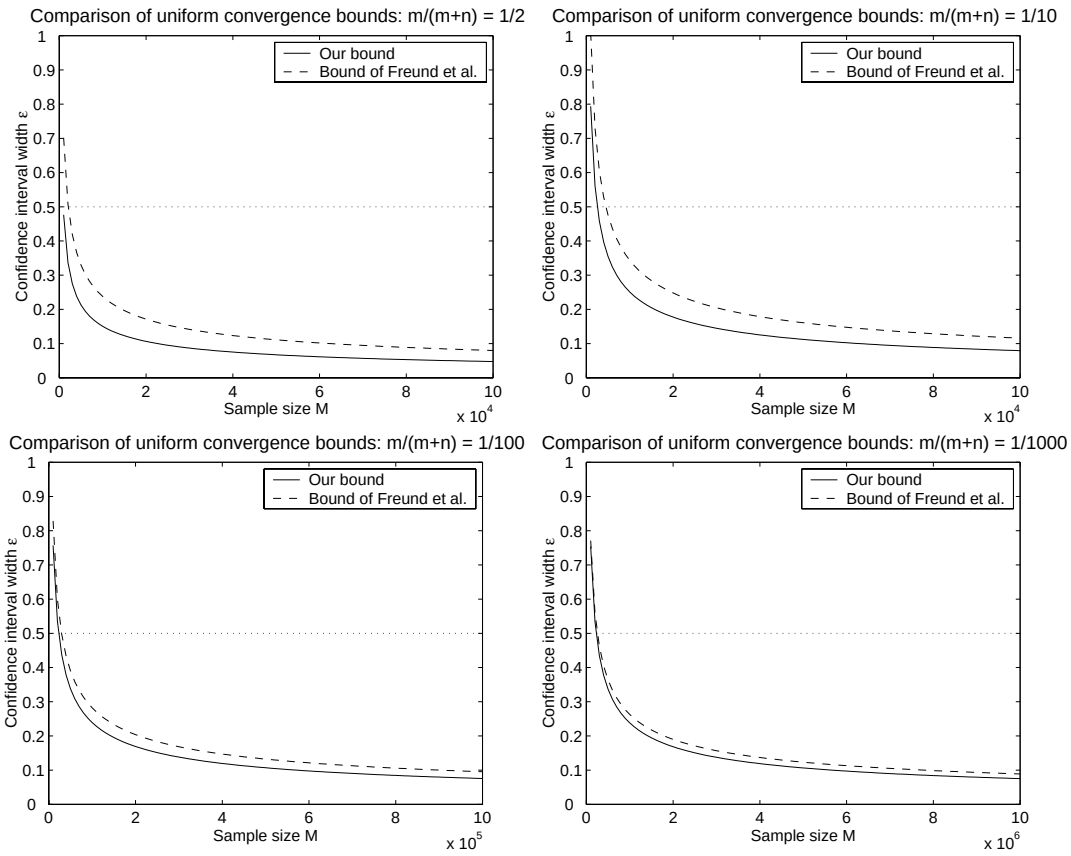

Figure 3.1: A comparison of our uniform convergence bound with that of Freund et al. [37] for the class of linear ranking functions on $\mathbb{R}$. The plots are for $\delta = 0.01$ and show how the confidence interval width ϵ given by the two bounds varies with the sample size $M = m + n$, for various values of $m/(m+n)$. In all cases where the bounds are meaningful ($\epsilon < 0.5$), our bound is tighter.

The above bounds are plotted in Figure 3.1 for $\delta = 0.01$ and various values of $m/(m+n)$. As can be seen, the bound provided by our result is considerably tighter.

3.6 Correctness of Functional Shape of Bound

Although our bound seems to be tighter than the previous bound of [37], it is still, in general, too loose to make quantitative predictions. Nevertheless, the bound can serve as a useful analysis tool if it displays a correct functional dependence on the training sample size parameters m and n. In this section we give an empirical assessment of the correctness of the functional shape of our bound.

We generated data points in $d = 16$ dimensions ($\mathcal{X} = \mathbb{R}^{16}$) as follows. We took $\mathcal{D}_+$ and $\mathcal{D}_-$

to be mixtures of two 16-dimensional Gaussians each, where each of the elements of both the means and the (diagonal) covariances of the Gaussians were chosen randomly from a uniform distribution on the interval $(0, 1)$. A test sample was generated by drawing 2500 points from $\mathcal{D}_+$ and 2500 points from $\mathcal{D}_-$.[4] Training samples of varying sizes were then generated by drawing m points from $\mathcal{D}_+$ and n points from $\mathcal{D}_-$ for various values of m and n. For each training sample, a linear ranking function in $\mathcal{F}_{\mathrm{lin}(16)}$ was learned using the RankBoost algorithm of Freund et al. [37] (the algorithm was run for $T = 20$ rounds). The empirical error of the learned ranking function on both training and test samples, as well as an upper bound on its expected error obtained from our uniform convergence result (using Corollary 3.1, at a confidence level $\delta = 0.01$) were then calculated. Since we do not have a means to characterize $r(\mathcal{F}_{\mathrm{lin}(16)}, m, n)$ exactly, we used the (loose) bound provided by Theorem 3.5 in calculating the upper bound on the expected error. The results, averaged over 10 trials (draws of the training sample) for each pair of values of m and n, are shown in Figure 3.2. As can be seen, the shape of the bound is in correspondence with that of the test error, suggesting that the bound does indeed display a correct functional dependence.

3.7 Conclusions and Open Questions

We have derived a distribution-free uniform convergence bound for the bipartite ranking error. Our bound is expressed in terms of a new set of combinatorial parameters that we have termed the bipartite rank-shatter coefficients. These coefficients define a new measure of complexity for real-valued function classes and play the same role in our result as do the standard VC-dimension related shatter coefficients in uniform convergence results for the classification error.

For the case of linear ranking functions on $\mathbb{R}$, for which we could compute the bipartite rank-shatter coefficients exactly, we have shown that our uniform convergence bound is considerably tighter than a recent uniform convergence bound derived by Freund et al. [37], which is expressed directly in terms of standard shatter coefficients from results for classification. This suggests that the bipartite rank-shatter coefficients we have introduced may be a more appropriate complexity measure for studying the bipartite ranking problem. However, in order to take advantage of our results, one needs to be able to characterize these coefficients for the class of ranking functions of interest. The biggest open question that arises from our study is, for what other function classes $\mathcal{F}$ can the bipartite rank-shatter coefficients $r(\mathcal{F}, m, n)$ be characterized? We have derived in Theorem 3.4 a general upper bound on the bipartite rank-shatter coefficients of a function class $\mathcal{F}$ in terms of the standard shatter coefficients of the function class $\tilde{\mathcal{F}}$

[4] To sample points from Gaussian mixtures we made use of the NETLAB toolbox written by Ian Nabney and Christopher Bishop, available from `http://www.ncrg.aston.ac.uk/netlab/`.

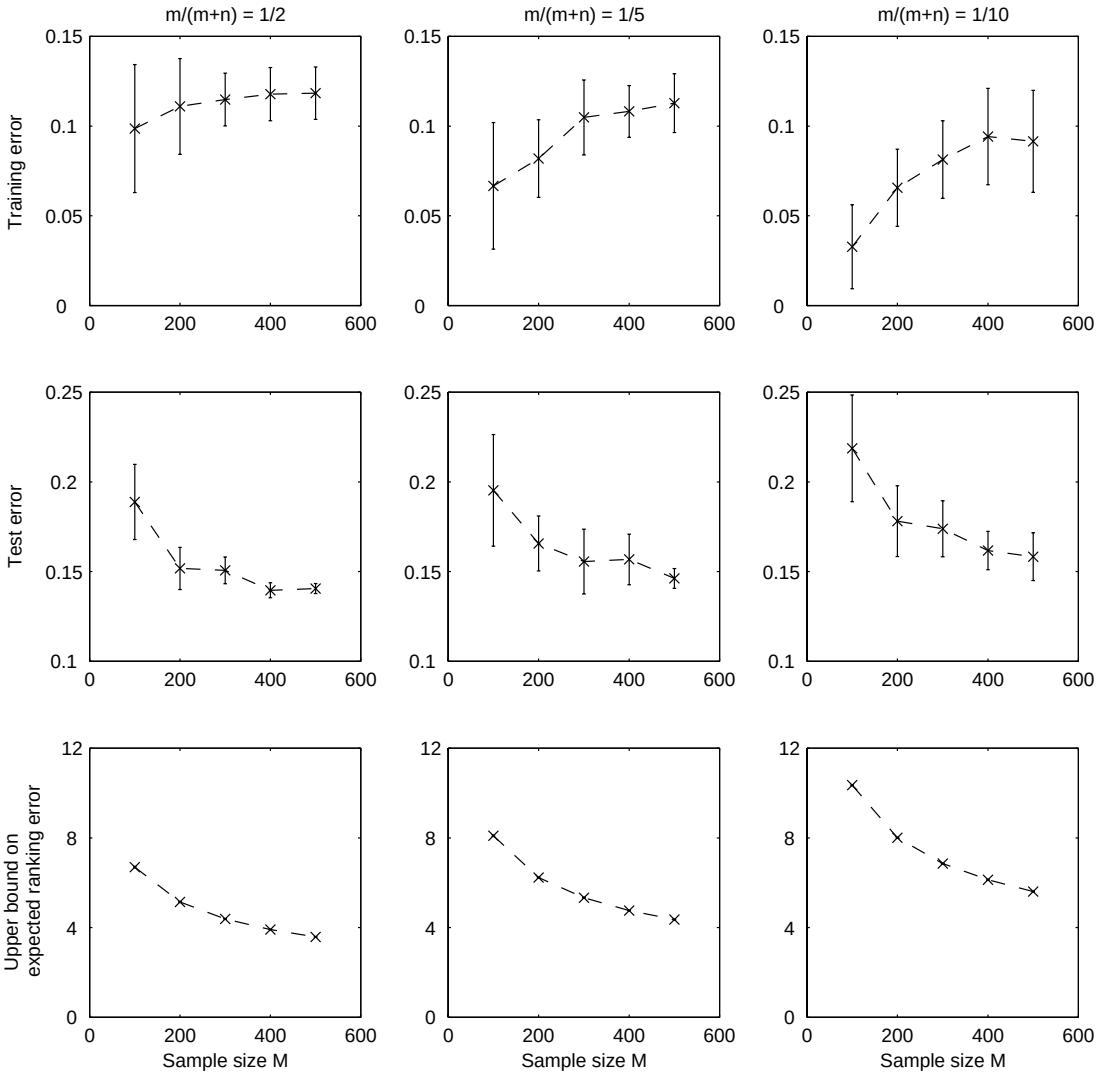

Figure 3.2: The training error (top row), test error (middle row), and upper bound on expected error (bottom row) of linear ranking functions learned from training samples of different sizes $M = m + n$ (see Section 3.6). The plots show mean values over 10 trials for each pair of values of m and n; the error bars show standard deviations (note that there are also error bars on the values of the upper bound; these have the same size as the error bars on the training error, but are invisible due to the difference in scale of the plots). Although the bound is quantitatively loose, its shape is in correspondence with that of the test error (and therefore correct).

(see Eq. (3.3)); this allows us to establish a polynomial upper bound on the bipartite rank-shatter coefficients for linear and higher-order polynomial ranking functions on $\mathbb{R}^d$ and other algebraically well-behaved function classes. However, this upper bound is inherently loose (see proof of Theorem 3.4). Is it possible to find tighter upper bounds on $r(\mathcal{F}, m, n)$ than that given by Theorem 3.4?

Our study also raises several other interesting questions. First, can we establish analogous complexity measures and generalization bounds for other forms of ranking problems (*i.e.*, other than bipartite)? Second, do there exist data-dependent bounds for ranking, analogous to existing margin bounds for classification? Finally, it also remains an open question whether tighter generalization bounds for the ranking error can be derived using different proof techniques.

Chapter 4

Learnability of Bipartite Ranking Functions

4.1 Introduction

One of the most important developments in machine learning has been the formulation of a rigorous theory of learnability, first proposed by Valiant [100] for binary classification functions defined on Boolean domains. Valiant's learning model (known now as the Probably Approximately Correct (PAC) learning model), and several variants and extensions thereof, have since been studied extensively, and have led to a rich set of theoretical results on classes of functions that can and cannot be learned, on algorithms that can be used to solve the learning problem, and on the computational complexity of learning various function classes. In particular, we now have a strong theoretical understanding of the learning problem for both classification and regression. In this chapter, we ask whether a similar theoretical understanding can be developed for ranking.

In the PAC model, a learning algorithm for a class $\mathcal{H}$ of binary ($\{-1,1\}$-valued) classification functions on $\mathcal{X}$ is a function $\mathcal{A} : \bigcup_{m=1}^{\infty}(\mathcal{X} \times \{-1,1\})^m \to \mathcal{H}$ with the following property: given any $\epsilon, \delta \in (0,1)$, there is an integer $m = m(\epsilon,\delta)$ such that for any distribution $\mathcal{D}$ on $\mathcal{X}$ and any target function $t \in \mathcal{H}$, given a random training sample $S = ((x_1, t(x_1)), \ldots, (x_m, t(x_m)))$ of size m in which the x_i are drawn i.i.d. according to $\mathcal{D}$, with probability at least $1-\delta$ the classification function $h = \mathcal{A}(S)$ output by $\mathcal{A}$ has prediction error $\mathbf{P}_{x\sim\mathcal{D}}\{h(x) \neq t(x)\} < \epsilon$. The smallest such integer $m(\epsilon,\delta)$ is called the sample complexity of $\mathcal{A}$. A class $\mathcal{H}$ is said to be learnable if there is a learning algorithm for $\mathcal{H}$, and efficiently learnable if there is a polynomial-time learning algorithm for $\mathcal{H}$.

In a classic paper, Blumer et al. [15] showed that, in the PAC model, the learnability of a class of binary classification functions $\mathcal{H}$ is characterized by a single combinatorial parameter of $\mathcal{H}$, namely its Vapnik-Chervonenkis (VC) dimension, in the sense that $\mathcal{H}$ is learnable if and

only if its VC dimension is finite. This characterization comprised two distinct results. The first made use of a uniform convergence result based on the work of Vapnik and Chervonenkis [102] to show the existence of a learning algorithm for $\mathcal{H}$ whose sample complexity could be upper bounded via the shatter coefficients (growth function) of $\mathcal{H}$, which in turn could be upper bounded in terms of the VC dimension of $\mathcal{H}$; this established that finiteness of the VC dimension is sufficient for learnability. The second result made use of the probabilistic method to show that the sample complexity of any learning algorithm for $\mathcal{H}$ is lower bounded by a linear function of the VC dimension of $\mathcal{H}$, so that an infinite VC dimension implies there is no learning algorithm with finite sample complexity; this established that finiteness of the VC dimension is also necessary for learnability.

The PAC model assumes the existence of an underlying 'target function'; this assumption was removed in a generalization of the PAC model studied in [15, 45, 57], often referred to as the 'agnostic' model. In this general model, examples are generated according to an arbitrary joint distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, 1\}$, and a learning algorithm is required to output with high probability a hypothesis $h \in \mathcal{H}$ with prediction error $\mathbf{P}_{(x,y)\sim\mathcal{D}}\{h(x) \neq y\}$ $(= L_{\mathcal{D}}(h))$ close to the best possible within the class $\mathcal{H}$. It has been shown that the VC dimension characterizes learnability also in this general model. Questions of the computational complexity of learning have been investigated for a large number of function classes in both models, leading to efficient algorithms in some cases and hardness results in others. For many common function classes, learning in the general model is hard, but polynomial-time algorithms exist for learning in the PAC model.

An analogous theory of learnability has been developed for the regression problem, starting with the work of Haussler [45] in which it was shown that finiteness of the pseudo-dimension of a class of (bounded) real-valued functions $\mathcal{F}$ is sufficient for learnability of $\mathcal{F}$ in the general learning model. As in the case of classification, this result made use of a uniform convergence result of [83] to show the existence of a learning algorithm for $\mathcal{F}$ whose sample complexity could be upper bounded via the covering numbers of $\mathcal{F}$, which in turn could be upper bounded in terms of the pseudo-dimension of $\mathcal{F}$. However, a lower bound on the sample complexity remained elusive. Later, Kearns and Schapire [56] showed that the sample complexity cannot be lower bounded in terms of the pseudo-dimension, and introduced a new measure of the richness of a real-valued function class known now as the fat-shattering dimension. It was then shown [56, 7, 13] that the sample complexity of any learning algorithm for a real-valued function class $\mathcal{F}$ is lower bounded by a linear function of the fat-shattering dimension of $\mathcal{F}$, and that the covering numbers of $\mathcal{F}$ can also be upper bounded in terms of this dimension, thus establishing a characterization of learnability for real-valued functions in terms of the fat-shattering dimension. Questions of the computational complexity of learning have also been

investigated for classes of real-valued functions, leading again to efficient algorithms in some cases and hardness results in others.

We define in this chapter a model of learnability for bipartite ranking functions (Section 4.2), and derive a number of results in this model. Our first main result (Section 4.3) provides a sufficient condition for the learnability of a class of ranking functions $\mathcal{F}$: we show that $\mathcal{F}$ is learnable if its bipartite rank-shatter coefficients (see Chapter 3) do not grow too quickly. As in the case of classification and regression, the proof of this result makes use of the uniform convergence result of Chapter 3 to show the existence of a learning algorithm for $\mathcal{F}$ whose sample complexity can be upper bounded via the bipartite rank-shatter coefficients of $\mathcal{F}$. Our second main result (Section 4.4) gives a necessary condition for learnability: we define a new combinatorial parameter for a class of ranking functions $\mathcal{F}$ that we term the rank dimension of $\mathcal{F}$, and show that $\mathcal{F}$ is learnable only if its rank dimension is finite. As in the case of classification, the proof of this result makes use of the probabilistic method to show that the sample complexity of any learning algorithm for $\mathcal{F}$ is lower bounded by a linear function of the rank dimension of $\mathcal{F}$. We use the above two results to give examples of both learnable and non-learnable classes of ranking functions. Finally, we investigate questions of the computational complexity of learning ranking functions (Section 4.5). As in classification, we find that for some common ranking function classes, learning in a general 'agnostic' model is hard, but efficient algorithms can be found for learning in a PAC-type model.

4.2 Learnability

Since the goal of learning is to find a ranking function that ranks accurately future instances, we would like a learning algorithm to find a ranking function with minimal expected ranking error. More specifically, if a learning algorithm selects a ranking function from a class of ranking functions $\mathcal{F}$, we would like it to output a ranking function $f \in \mathcal{F}$ with expected error $R_{\mathcal{D}_+,\mathcal{D}_-}(f)$ close to the best possible within the class $\mathcal{F}$, *i.e.*, close to

$$R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \;=\; \inf_{g\in\mathcal{F}} R_{\mathcal{D}_+,\mathcal{D}_-}(g)\,. \tag{4.1}$$

We formalize this idea below, following closely the notation and terminology used by Anthony and Bartlett [12] to describe learnability results for classification and regression.

Definition 4.1 (Learnability) *Let $\mathcal{F}$ be a class of real-valued ranking functions on $\mathcal{X}$. A* learning algorithm *$\mathcal{A}$ for $\mathcal{F}$ is a function $\mathcal{A} : (\bigcup_{m=1}^{\infty} \mathcal{X}^m) \times (\bigcup_{n=1}^{\infty} \mathcal{X}^n) \to \mathcal{F}$ with the following property: given any $\rho \in (0,1) \cap \mathbb{Q}$ and any $\epsilon, \delta \in (0,1)$, there is an integer $M = M(\epsilon, \delta, \rho)$ such*

that $m = \rho M \in \mathbb{N}$, $n = (1-\rho)M \in \mathbb{N}$, *and for any distributions* $\mathcal{D}_+, \mathcal{D}_-$ *on* $\mathcal{X}$,

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+, S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \ \geq \ \epsilon \right\} \ \leq \ \delta \,.$$

The smallest such integer $M(\epsilon, \delta, \rho)$ *is called the* sample complexity *of* $\mathcal{A}$, *denoted* $M_{\mathcal{A}}(\epsilon, \delta, \rho)$. *We say that* $\mathcal{F}$ *is* learnable *if there is a learning algorithm for* $\mathcal{F}$.

Notice the introduction of the additional parameter ρ in the above definition, which was not required in classification. This parameter represents the 'positive skew', *i.e.*, the proportion of positive examples. Its role will become clear in subsequent sections.

As in [12], our main model above corresponds to a general 'agnostic' model in which no assumption is made on the distributions $\mathcal{D}_+$ and $\mathcal{D}_-$; we refer to this as the *standard* model. We can also define a PAC-type model in which the distributions $\mathcal{D}_+$ and $\mathcal{D}_-$ are restricted to correspond to an underlying target function; following [12], we refer to this as the *restricted* model.

Definition 4.2 (Learnability in restricted model) *Let* $\mathcal{F}$ *be a class of real-valued ranking functions on* $\mathcal{X}$. *A* learning algorithm $\mathcal{A}$ *for* $\mathcal{F}$ *in the restricted model is a function* $\mathcal{A} : (\bigcup_{m=1}^{\infty} \mathcal{X}^m) \times (\bigcup_{n=1}^{\infty} \mathcal{X}^n) \to \mathcal{F}$ *with the following property: given any* $\rho \in (0,1) \cap \mathbb{Q}$ *and any* $\epsilon, \delta \in (0,1)$, *there is an integer* $M = M(\epsilon, \delta, \rho)$ *such that* $m = \rho M \in \mathbb{N}$, $n = (1-\rho)M \in \mathbb{N}$, *and for any distributions* $\mathcal{D}_+, \mathcal{D}_-$ *on* $\mathcal{X}$ *for which there is a target function* $t \in \mathcal{F}$ *such that* $R_{\mathcal{D}_+,\mathcal{D}_-}(t) = 0$,

$$\mathbf{P}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+, S_-)) \ \geq \ \epsilon \right\} \ \leq \ \delta \,.$$

The smallest such integer $M(\epsilon, \delta, \rho)$ *is called the* sample complexity *of* $\mathcal{A}$, *denoted* $M_{\mathcal{A}}(\epsilon, \delta, \rho)$. *We say that* $\mathcal{F}$ *is* learnable *in the restricted model if there is a learning algorithm for* $\mathcal{F}$ *in this model.*

Unless specified otherwise, we will use the term learnability to mean learnability in the standard model. Clearly, if a class of ranking functions $\mathcal{F}$ is learnable, then $\mathcal{F}$ is learnable in the restricted model.

4.3 Upper Bound on Sample Complexity

In this section we show that any algorithm that minimizes the empirical ranking error over a class of ranking functions $\mathcal{F}$ is a learning algorithm for $\mathcal{F}$ if the bipartite rank-shatter coefficients of $\mathcal{F}$ (see Definition 3.2) do not grow too quickly, and obtain an upper bound on the sample complexity of such an algorithm.

Definition 4.3 (Empirical error minimization (EEM) algorithm) *Let $\mathcal{F}$ be a class of ranking functions on $\mathcal{X}$. Define an* empirical error minimization (EEM) algorithm *for $\mathcal{F}$ to be any function $\mathcal{A} : (\bigcup_{m=1}^{\infty} \mathcal{X}^m) \times (\bigcup_{n=1}^{\infty} \mathcal{X}^n) \to \mathcal{F}$ with the property that for any $m, n \in \mathbb{N}$ and any $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$,*

$$\hat{R}_{S_+,S_-}(\mathcal{A}(S_+, S_-)) \quad = \quad \min_{g \in \mathcal{F}} \hat{R}_{S_+,S_-}(g)\,.$$

Theorem 4.1 *Let $\mathcal{F}$ be a class of ranking functions on $\mathcal{X}$, and let $\mathcal{A}$ be any EEM algorithm for $\mathcal{F}$. If there exist constants $c_1 > 0$, $c_2 \geq 0$ such that $r(\mathcal{F}, m, n) \leq c_1(mn)^{c_2}$ for all $m, n \in \mathbb{N}$, then $\mathcal{A}$ is a learning algorithm for $\mathcal{F}$ with sample complexity*

$$M_{\mathcal{A}}(\epsilon, \delta, \rho) \quad \leq \quad \left\lceil \frac{64}{\rho(1-\rho)\epsilon^2} \left(4c_2 \ln\left(\frac{16}{\epsilon}\right) + c_2 \ln\left(\frac{c_2^2}{e^2\rho(1-\rho)}\right) + \ln\left(\frac{4c_1}{\delta}\right)\right)\right\rceil_{\rho},$$

where $\lceil u \rceil_\rho$ denotes the smallest integer M greater than or equal to u for which $\rho M \in \mathbb{N}$.

Proof The proof of this result makes use of the uniform convergence result for the ranking error derived in Chapter 3. We first show that for any $m, n \in \mathbb{N}$, any $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ and any distributions $\mathcal{D}_+, \mathcal{D}_-$ on $\mathcal{X}$,

$$R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+, S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \quad \leq \quad 2 \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| .$$

Since $R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) = \inf_{g \in \mathcal{F}} R_{\mathcal{D}_+,\mathcal{D}_-}(g)$, for any $\alpha > 0$ there is an $f^* \in \mathcal{F}$ with

$$R_{\mathcal{D}_+,\mathcal{D}_-}(f^*) < R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) + \alpha\,.$$

Therefore, we have

$$\begin{aligned}
&R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+, S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \\
&\quad = \quad \left(R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+, S_-)) - \hat{R}_{S_+,S_-}(\mathcal{A}(S_+, S_-))\right) + \left(\hat{R}_{S_+,S_-}(\mathcal{A}(S_+, S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F})\right) \\
&\quad < \quad \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| + \left(\hat{R}_{S_+,S_-}(\mathcal{A}(S_+, S_-)) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*) + \alpha\right) \\
&\quad \leq \quad \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| + \left(\hat{R}_{S_+,S_-}(f^*) - R_{\mathcal{D}_+,\mathcal{D}_-}(f^*) + \alpha\right) \\
&\quad \leq \quad 2 \sup_{f \in \mathcal{F}} \left| \hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f) \right| + \alpha\,.
\end{aligned}$$

Since this is true for all $\alpha > 0$, we must have

$$R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+,S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \;\leq\; 2\sup_{f\in\mathcal{F}}\left|\hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f)\right| .$$

Now, let $\rho \in (0,1)\cup\mathbb{Q}$ and $\epsilon,\delta \in (0,1)$, and let $\mathcal{D}_+,\mathcal{D}_-$ be any distributions on $\mathcal{X}$. For any $M \in \mathbb{N}$ for which $m = \rho M \in \mathbb{N}$, $n = (1-\rho)M \in \mathbb{N}$, we then have

$$\begin{aligned}
&\mathbf{P}_{S_+\sim\mathcal{D}_+^m, S_-\sim\mathcal{D}_-^n}\left\{R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+,S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \;\geq\; \epsilon\right\} \qquad (4.2)\\
&\leq\; \mathbf{P}_{S_+\sim\mathcal{D}_+^m, S_-\sim\mathcal{D}_-^n}\left\{\sup_{f\in\mathcal{F}}\left|\hat{R}_{S_+,S_-}(f) - R_{\mathcal{D}_+,\mathcal{D}_-}(f)\right| \;\geq\; \epsilon/2\right\}\\
&\leq\; 4\cdot r(\mathcal{F},2m,2n)\cdot e^{-mn\epsilon^2/32(m+n)} \qquad \text{(by Theorem 3.2)}\\
&=\; 4\cdot r(\mathcal{F},2\rho M,2(1-\rho)M)\cdot e^{-\rho(1-\rho)M\epsilon^2/32} .
\end{aligned}$$

Therefore, to make the probability in Eq. (4.2) smaller than δ, it is sufficient if

$$M \;\geq\; \frac{32}{\rho(1-\rho)\epsilon^2}\left(\ln r\Big(\mathcal{F},2\rho M,2(1-\rho)M\Big) + \ln\left(\frac{4}{\delta}\right)\right).$$

Now, suppose there exist constants $c_1 > 0$, $c_2 \geq 0$ such that $r(\mathcal{F},m,n) \leq c_1(mn)^{c_2}$ for all $m,n \in \mathbb{N}$. Then it suffices to have

$$M \;\geq\; \frac{32}{\rho(1-\rho)\epsilon^2}\left(2c_2\ln M + c_2\ln(4\rho(1-\rho)) + \ln\left(\frac{4c_1}{\delta}\right)\right).$$

Since $\ln u \leq au - \ln a - 1$ for all $a,u > 0$, we have

$$\begin{aligned}
\frac{64c_2}{\rho(1-\rho)\epsilon^2}\ln M &\leq \frac{64c_2}{\rho(1-\rho)\epsilon^2}\left(\frac{\rho(1-\rho)\epsilon^2}{128c_2}M - \ln\left(\frac{\rho(1-\rho)\epsilon^2}{128c_2}\right) - 1\right)\\
&= \frac{M}{2} + \frac{64c_2}{\rho(1-\rho)\epsilon^2}\ln\left(\frac{128c_2}{e\rho(1-\rho)\epsilon^2}\right).
\end{aligned}$$

Therefore, it suffices to have

$$M \;\geq\; \frac{M}{2} + \frac{32}{\rho(1-\rho)\epsilon^2}\left(2c_2\ln\left(\frac{128c_2}{e\rho(1-\rho)\epsilon^2}\right) + c_2\ln(4\rho(1-\rho)) + \ln\left(\frac{4c_1}{\delta}\right)\right).$$

Simplifying terms, we have that

$$M \;\geq\; \frac{64}{\rho(1-\rho)\epsilon^2}\left(4c_2\ln\left(\frac{16}{\epsilon}\right) + c_2\ln\left(\frac{c_2^2}{e^2\rho(1-\rho)}\right) + \ln\left(\frac{4c_1}{\delta}\right)\right)$$

suffices to make the probability in Eq. (4.2) smaller than δ. The result then follows from the definition of sample complexity (Definition 4.1). □

Notice that unlike classification, the upper bound on the sample complexity in ranking for given (ϵ, δ) grows larger as the positive skew ρ departs from 1/2, *i.e.*, as the balance between positive and negative examples becomes more uneven. Theorem 4.1 can be used to show learnability of any class of ranking functions whose bipartite rank-shatter coefficients can be bounded appropriately; we give some examples below.

Example 4.1 (Finite function classes) *Let $\mathcal{F}$ be a finite class of ranking functions on some instance space $\mathcal{X}$. Then clearly, $r(\mathcal{F}, m, n) \leq |\mathcal{F}|$ for all $m, n \in \mathbb{N}$. Thus we have from Theorem 4.1 that $\mathcal{F}$ is learnable; in particular, taking $c_1 = |\mathcal{F}|$, $c_2 = 0$, we have that any EEM algorithm $\mathcal{A}$ for $\mathcal{F}$ is a learning algorithm for $\mathcal{F}$ with sample complexity*

$$M_{\mathcal{A}}(\epsilon, \delta, \rho) \leq \left\lceil \frac{64}{\rho(1-\rho)\epsilon^2} \ln\left(\frac{4|\mathcal{F}|}{\delta}\right) \right\rceil_{\rho}.$$

It is in fact possible to obtain a slightly tighter upper bound in this case; using the uniform convergence result of Section 2.5 for finite function classes, it can be shown that any EEM algorithm $\mathcal{A}$ for a finite class of ranking functions $\mathcal{F}$ is a learning algorithm for $\mathcal{F}$ with sample complexity

$$M_{\mathcal{A}}(\epsilon, \delta, \rho) \leq \left\lceil \frac{2}{\rho(1-\rho)\epsilon^2} \ln\left(\frac{2|\mathcal{F}|}{\delta}\right) \right\rceil_{\rho}.$$

Example 4.2 (Linear ranking functions) *Let $d \in \mathbb{N}$, and let $\mathcal{F}_{\text{lin}(d)}$ be the class of linear ranking functions on $\mathbb{R}^d$. Then from Theorem 3.5 we have $r(\mathcal{F}_{\text{lin}(d)}, m, n) \leq (2emn/d)^d$ for all $m, n \in \mathbb{N}$. Thus we have from Theorem 4.1 that $\mathcal{F}_{\text{lin}(d)}$ is learnable; in particular, taking $c_1 = (2e/d)^d$, $c_2 = d$, we have that any EEM algorithm $\mathcal{A}$ for $\mathcal{F}_{\text{lin}(d)}$ is a learning algorithm for $\mathcal{F}_{\text{lin}(d)}$ with sample complexity*

$$M_{\mathcal{A}}(\epsilon, \delta, \rho) \leq \left\lceil \frac{64}{\rho(1-\rho)\epsilon^2} \left(4d \ln\left(\frac{16}{\epsilon}\right) + d \ln\left(\frac{2d}{e\rho(1-\rho)}\right) + \ln\left(\frac{4}{\delta}\right) \right) \right\rceil_{\rho}.$$

Example 4.3 (Polynomial ranking functions) *Let $d, q \in \mathbb{N}$, and let $\mathcal{F}_{\text{poly}(d,q)}$ be the class of polynomial ranking functions on $\mathbb{R}^d$ with degree less than or equal to q. Then from Theorem 3.6 we have $r(\mathcal{F}_{\text{poly}(d,q)}, m, n) \leq (2emn/C(d,q))^{C(d,q)}$ for all $m, n \in \mathbb{N}$, where*

$$C(d, q) = \sum_{i=1}^{q} \left(\binom{d}{i} \sum_{j=1}^{q} \binom{j-1}{i-1} \right).$$

Thus we have from Theorem 4.1 that $\mathcal{F}_{\mathrm{poly}(d,q)}$ *is learnable; in particular, taking* $c_1 = (2e/C(d,q))^{C(d,q)}$, $c_2 = C(d,q)$, *we have that any EEM algorithm* $\mathcal{A}$ *for* $\mathcal{F}_{\mathrm{poly}(d,q)}$ *is a learning algorithm for* $\mathcal{F}_{\mathrm{poly}(d,q)}$ *with sample complexity*

$$M_{\mathcal{A}}(\epsilon,\delta,\rho) \leq \left\lceil \frac{64}{\rho(1-\rho)\epsilon^2}\left(4C(d,q)\ln\left(\frac{16}{\epsilon}\right) + C(d,q)\ln\left(\frac{2C(d,q)}{e\rho(1-\rho)}\right) + \ln\left(\frac{4}{\delta}\right)\right)\right\rceil_{\rho}.$$

4.4 Lower Bound on Sample Complexity

In this section we define a new combinatorial parameter for a class of ranking functions $\mathcal{F}$ that we term the rank dimension of $\mathcal{F}$, and show that the sample complexity of any learning algorithm for $\mathcal{F}$ is lower bounded by a linear function of its rank dimension.

Definition 4.4 (Rank-shattering) *Let* $\mathcal{F}$ *be a class of real-valued functions on* $\mathcal{X}$, *let* $r \in \mathbb{N}$, *and let* $S = \{(w_1', w_1''), \ldots, (w_r', w_r'')\}$ *be a set of* r *pairs of instances in* $\mathcal{X}$. *For each* $i \in \{1,\ldots,r\}$, $b \in \{0,1\}^r$, *define*

$$w_i^{b+} = \begin{cases} w_i' & \text{if } b_i = 1 \\ w_i'' & \text{if } b_i = 0 \end{cases}, \qquad w_i^{b-} = \begin{cases} w_i'' & \text{if } b_i = 1 \\ w_i' & \text{if } b_i = 0 \end{cases}.$$

We say that $\mathcal{F}$ rank-shatters S *if for each* $b \in \{0,1\}^r$, *there is a ranking function* $f_b \in \mathcal{F}$ *such that for all* $i,j \in \{1,\ldots,r\}$, $f_b(w_i^{b+}) > f_b(w_j^{b-})$.

Definition 4.5 (Rank dimension) *Let* $\mathcal{F}$ *be a class of real-valued functions on* $\mathcal{X}$. *Define the* rank dimension *of* $\mathcal{F}$, *denoted by* $\text{rank-dim}(\mathcal{F})$, *to be the largest positive integer* r *for which there exists a set of* r *pairs of instances in* $\mathcal{X}$ *that is rank-shattered by* $\mathcal{F}$.

Theorem 4.2 *Let* $\mathcal{F}$ *be a class of ranking functions on* $\mathcal{X}$ *with* $\text{rank-dim}(\mathcal{F}) = r \geq 2$. *Then for any function* $\mathcal{A} : (\bigcup_{m=1}^{\infty} \mathcal{X}^m) \times (\bigcup_{n=1}^{\infty} \mathcal{X}^n) \to \mathcal{F}$, *any* $m,n \in \mathbb{N}$ *such that* $m+n \geq 2r$, *and any* $\epsilon > 0$, *there exist distributions* $\mathcal{D}_+, \mathcal{D}_-$ *on* $\mathcal{X}$ *such that*

$$\begin{aligned} &\mathbf{E}_{S_+\sim\mathcal{D}_+^m, S_-\sim\mathcal{D}_-^n}\left\{R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+,S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F})\right\} \\ &\quad\geq \frac{1}{2^9}\sqrt{\frac{r}{m+n}}\left(1-\sqrt{1-e^{-(2m/(m+n)+1)}}\right)^2\left(1-\sqrt{1-e^{-(2n/(m+n)+1)}}\right)^2. \end{aligned}$$

We shall need the following result, the proof of which follows from Slud's inequality and standard tail bounds for the normal distribution (see [12, page 61]):

Lemma 4.1 *Let* $\mathrm{Bin}(m,p)$ *denote a Binomial random variable with parameters* $m \in \mathbb{N}$ *and* $p \in (0,1)$. *Then for any* $m \in \mathbb{N}$ *and* $\alpha \in (0,1)$,

$$\mathbf{P}\left\{\mathrm{Bin}\left(m, \frac{1-\alpha}{2}\right) \geq \frac{m}{2}\right\} \geq \frac{1}{4}\left(1-\sqrt{1-e^{-(m+1)\alpha^2/(1-\alpha^2)}}\right).$$

Proof (of Theorem 4.2)
The proof makes use of ideas similar to those used to prove lower bounds in the case of classification; specifically, a finite set of distributions is constructed, and it is shown, using the probabilistic method, that for any function $\mathcal{A}$ there exist distributions in this set for which the above lower bound holds.

Let $S = \{(w'_1, w''_1), \ldots, (w'_r, w''_r)\}$ be a set of r pairs of instances in $\mathcal{X}$ that is rank-shattered by $\mathcal{F}$. We construct a family of 2^r pairs of distributions $\{(\mathcal{D}_{b+}, \mathcal{D}_{b-}) : b \in \{0,1\}^r\}$ on $\mathcal{X}$ as follows. For each $b \in \{0,1\}^r$, define

$$\mathcal{D}_{b+}(w'_i) = \begin{cases} (1+\alpha)/2r & \text{if } b_i = 1 \\ (1-\alpha)/2r & \text{if } b_i = 0 \end{cases} \qquad \mathcal{D}_{b-}(w'_i) = \begin{cases} (1-\alpha)/2r & \text{if } b_i = 1 \\ (1+\alpha)/2r & \text{if } b_i = 0 \end{cases}$$

$$\mathcal{D}_{b+}(w''_i) = \begin{cases} (1-\alpha)/2r & \text{if } b_i = 1 \\ (1+\alpha)/2r & \text{if } b_i = 0 \end{cases} \qquad \mathcal{D}_{b-}(w''_i) = \begin{cases} (1+\alpha)/2r & \text{if } b_i = 1 \\ (1-\alpha)/2r & \text{if } b_i = 0 \end{cases}$$

$$\mathcal{D}_{b+}(x) = 0 \quad \text{for } x \neq w'_i, w''_i \qquad \mathcal{D}_{b-}(x) = 0 \quad \text{for } x \neq w'_i, w''_i$$

Here α is a constant in $(0,1)$ whose value will be determined later. Using the notation of Definition 4.4, it can be verified that for any $f : \mathcal{X} \to \mathbb{R}$,

$$R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f) = \left(\frac{1-\alpha}{2}\right) + \frac{\alpha}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r} \ell_{\text{rank}}\left(f, w_i^{b+}, w_j^{b-}\right).$$

Since S is rank-shattered by $\mathcal{F}$, for each $b \in \{0,1\}^r$ there is a function $f_b \in \mathcal{F}$ such that for all $i, j \in \{1, \ldots, r\}$, $f_b(w_i^{b+}) > f_b(w_j^{b-})$, and therefore $\ell_{\text{rank}}\left(f, w_i^{b+}, w_j^{b-}\right) = 0$. It follows from the above equation that

$$R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F}) = \left(\frac{1-\alpha}{2}\right).$$

Therefore, for any $f \in \mathcal{F}$, we have

$$R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F}) = \frac{\alpha}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r} \ell_{\text{rank}}\left(f, w_i^{b+}, w_j^{b-}\right). \tag{4.3}$$

Now, let $\mathcal{A} : (\bigcup_{m=1}^{\infty} \mathcal{X}^m) \times (\bigcup_{n=1}^{\infty} \mathcal{X}^n) \to \mathcal{F}$ be any function, and for any $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, denote by f_{S_+,S_-} the ranking function $\mathcal{A}(S_+, S_-) \in \mathcal{F}$ output by $\mathcal{A}$. Then from Eq. (4.3), we have for any $b \in \{0,1\}^r$,

$$\begin{aligned} &\mathbf{E}_{S_+ \sim \mathcal{D}_{b+}^m, S_- \sim \mathcal{D}_{b-}^n}\left\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{S_+,S_-}) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\right\} \\ &\quad = \frac{\alpha}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r} \mathbf{E}_{S_+ \sim \mathcal{D}_{b+}^m, S_- \sim \mathcal{D}_{b-}^n}\left\{\ell_{\text{rank}}\left(f_{S_+,S_-}, w_i^{b+}, w_j^{b-}\right)\right\}. \end{aligned} \tag{4.4}$$

We use the probabilistic method to show that the above quantity is greater than the stated lower bound for at least one pair of distributions $\mathcal{D}_{b+}, \mathcal{D}_{b-}$. In particular, we show that if $b \in \{0,1\}^r$ is chosen uniformly at random, then the expected value of the above quantity is greater than the stated lower bound; this implies that there is at least one $b \in \{0,1\}^r$ for which the bound holds. The techniques we use are similar to those used in the case of classification (see, for example, [12, Chapter 5]); the details are considerably more involved. Denoting the uniform distribution over $\{0,1\}^r$ by $\mathcal{U}$, we shall show that for any $\alpha \in (0,1)$,

$$\begin{aligned}
&\mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+\sim\mathcal{D}_{b+}^m, S_-\sim\mathcal{D}_{b-}^n}\Big\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{S_+,S_-}) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\Big\}\Big\} \\
&\quad\geq \frac{\alpha}{2^9}\Big(1-\sqrt{1-e^{-(2m/r+1)\alpha^2/(1-\alpha^2)}}\Big)^2\Big(1-\sqrt{1-e^{-(2n/r+1)\alpha^2/(1-\alpha^2)}}\Big)^2.
\end{aligned}$$

For $S_+ \in \mathcal{X}^m$ and $i \in \{1,\ldots,r\}$, let $m_i(S_+)$ denote the total number of occurrences of either w_i' or w_i'' in S_+, and let $\underline{m}(S_+) = (m_1(S_+),\ldots,m_r(S_+))$. Similarly, for $S_- \in \mathcal{X}^n$ and $j \in \{1,\ldots,r\}$, let $n_j(S_-)$ denote the total number of occurrences of either w_j' or w_j'' in S_-, and let $\underline{n}(S_-) = (n_1(S_-),\ldots,n_r(S_-))$. Let $\Pi_r(m)$ denote the set of all r-tuples $\underline{m} = (m_1,\ldots,m_r) \in \mathbb{N}^r$ such that $\sum_{i=1}^r m_i = m$, and Let $\Pi_r(n)$ denote the set of all r-tuples $\underline{n} = (n_1,\ldots,n_r) \in \mathbb{N}^r$ such that $\sum_{j=1}^r n_j = n$. Then, from Eq. (4.4), we have

$$\begin{aligned}
&\mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+\sim\mathcal{D}_{b+}^m, S_-\sim\mathcal{D}_{b-}^n}\Big\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{S_+,S_-}) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\Big\}\Big\} \\
&= \frac{\alpha}{r^2}\sum_{i=1}^r\sum_{j=1}^r \mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+\sim\mathcal{D}_{b+}^m, S_-\sim\mathcal{D}_{b-}^n}\Big\{\ell_{\text{rank}}\Big(f_{S_+,S_-}, w_i^{b+}, w_j^{b-}\Big)\Big\}\Big\} \\
&= \frac{\alpha}{r^2}\sum_{i=1}^r\sum_{j=1}^r \mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{\underline{m}(S_+),\underline{n}(S_-)|b}\Big\{\mathbf{E}_{S_+,S_-|b,\underline{m}(S_+),\underline{n}(S_-)}\Big\{\ell_{\text{rank}}\Big(f_{S_+,S_-}, w_i^{b+}, w_j^{b-}\Big)\Big\}\Big\}\Big\} \\
&= \frac{\alpha}{r^2}\sum_{i=1}^r\sum_{j=1}^r \mathbf{E}_{b\sim\mathcal{U}}\Bigg\{\sum_{\{\underline{m}\in\Pi_r(m)\}}\sum_{\{\underline{n}\in\Pi_r(n)\}} \mathbf{P}^b\Big\{\underline{m}(S_+)=\underline{m}, \underline{n}(S_-)=\underline{n}\Big\}\times \\
&\qquad\qquad \mathbf{E}_{S_+,S_-|b,\underline{m}(S_+)=\underline{m},\underline{n}(S_-)=\underline{n}}\Big\{\ell_{\text{rank}}\Big(f_{S_+,S_-}, w_i^{b+}, w_j^{b-}\Big)\Big\}\Bigg\}, \quad (4.5)
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{P}^b\Big\{\underline{m}(S_+)=\underline{m}, \underline{n}(S_-)=\underline{n}\Big\} &\equiv \mathbf{P}_{S_+\sim\mathcal{D}_{b+}^m, S_-\sim\mathcal{D}_{b-}^n}\Big\{\underline{m}(S_+)=\underline{m}, \underline{n}(S_-)=\underline{n}\Big\} \\
&= \mathbf{P}_{S_+\sim\mathcal{D}_{b+}^m}\Big\{\underline{m}(S_+)=\underline{m}\Big\}\,\mathbf{P}_{S_-\sim\mathcal{D}_{b-}^n}\Big\{\underline{n}(S_-)=\underline{n}\Big\}.
\end{aligned}$$

Now, for each $b \in \{0,1\}^r$, $\underline{m}(S_+)$ and $\underline{n}(S_-)$ follow multinomial distributions, each with parameters $(1/r,\ldots,1/r)$. Thus $\mathbf{P}^b\Big\{\underline{m}(S_+)=\underline{m}, \underline{n}(S_-)=\underline{n}\Big\}$ is the same for all b, and we can

write Eq. (4.5) as

$$\begin{aligned}
&\mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+\sim\mathcal{D}_{b+}^m,S_-\sim\mathcal{D}_{b-}^n}\Big\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{S_+,S_-})-R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\Big\}\Big\}\\
&\quad=\ \frac{\alpha}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{\{\underline{m}\in\Pi_r(m)\}}\sum_{\{\underline{n}\in\Pi_r(n)\}}\Bigg[\mathbf{P}\Big\{\underline{m}(S_+)=\underline{m}\Big\}\ \mathbf{P}\Big\{\underline{n}(S_-)=\underline{n}\Big\}\times\\
&\qquad\qquad\mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+,S_-|b,\underline{m}(S_+)=\underline{m},\underline{n}(S_-)=\underline{n}}\Big\{\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_i^{b+},w_j^{b-}\Big)\Big\}\Big\}\Bigg]\\
&\quad=\ \frac{1}{2}\frac{\alpha}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r}\sum_{\{\underline{m}\in\Pi_r(m)\}}\sum_{\{\underline{n}\in\Pi_r(n)\}}\mathbf{P}\Big\{\underline{m}(S_+)=\underline{m}\Big\}\ \mathbf{P}\Big\{\underline{n}(S_-)=\underline{n}\Big\}\ \lambda(i,j,\underline{m},\underline{n})\,, \qquad (4.6)
\end{aligned}$$

where

$$\lambda(i,j,\underline{m},\underline{n})\ =\ \mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+,S_-|b,\underline{m},\underline{n}}\Big\{\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_i^{b+},w_j^{b-}\Big)+\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_j^{b+},w_i^{b-}\Big)\Big\}\Big\}. \qquad (4.7)$$

Now, for each $i,j\in\{1,\ldots,r\}$, $i\neq j$, let $S_+^{(ij)}$ denote the subsequence of S_+ that contains only w_i',w_i'' and w_j',w_j'', and let $S_+^{(-ij)}$ denote the subsequence of S_+ that does *not* contain w_i',w_i'' and w_j',w_j''. Similarly, let $S_-^{(ij)}$ denote the subsequence of S_- that contains only w_i',w_i'' and w_j',w_j'', and let $S_-^{(-ij)}$ denote the subsequence of S_- that does *not* contain w_i',w_i'' and w_j',w_j''. Let $b^(-ij)\in\{0,1\}^{r-2}$ denote the bit vector formed from $b\in\{0,1\}^r$ by removing the ith and jth components b_i and b_j from b. Then for $i\neq j$, we have,

$$\begin{aligned}
\lambda(i,j,\underline{m},\underline{n})\ &=\ \mathbf{E}_{b^{(-ij)}}\Big\{\mathbf{E}_{b_i,b_j}\Big\{\mathbf{E}_{S_+^{(-ij)},S_-^{(-ij)}|b,\underline{m},\underline{n}}\Big\{\mathbf{E}_{S_+^{(ij)},S_-^{(ij)}|b,\underline{m},\underline{n},S_+^{(-ij)},S_-^{(-ij)}}\Big\{\\
&\qquad\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_i^{b+},w_j^{b-}\Big)+\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_j^{b+},w_i^{b-}\Big)\Big\}\Big\}\Big\}\Big\}. \qquad (4.8)
\end{aligned}$$

Now, given $\underline{m}(S_+)=\underline{m}$ and $\underline{n}(S_-)=\underline{n}$, the probability of $S_+^{(-ij)}$ and $S_-^{(-ij)}$ depends only on $b^{(-ij)}$, and not on $b^{(ij)}$. Thus, we get

$$\begin{aligned}
\lambda(i,j,\underline{m},\underline{n})\ &=\ \mathbf{E}_{b^{(-ij)}}\Big\{\mathbf{E}_{b_i,b_j}\Big\{\mathbf{E}_{S_+^{(-ij)},S_-^{(-ij)}|b^{(-ij)},\underline{m},\underline{n}}\Big\{\mathbf{E}_{S_+^{(ij)},S_-^{(ij)}|b,\underline{m},\underline{n},S_+^{(-ij)},S_-^{(-ij)}}\Big\{\\
&\qquad\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_i^{b+},w_j^{b-}\Big)+\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_j^{b+},w_i^{b-}\Big)\Big\}\Big\}\Big\}\Big\}\\
&=\ \mathbf{E}_{b^{(-ij)}}\Big\{\mathbf{E}_{S_+^{(-ij)},S_-^{(-ij)}|b^{(-ij)},\underline{m},\underline{n}}\Big\{\mathbf{E}_{b_i,b_j}\Big\{\mathbf{E}_{S_+^{(ij)},S_-^{(ij)}|b,\underline{m},\underline{n},S_+^{(-ij)},S_-^{(-ij)}}\Big\{\\
&\qquad\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_i^{b+},w_j^{b-}\Big)+\ell_{\mathrm{rank}}\Big(f_{S_+,S_-},w_j^{b+},w_i^{b-}\Big)\Big\}\Big\}\Big\}\Big\}\\
&=\ \mathbf{E}_{b^{(-ij)}}\Big\{\mathbf{E}_{S_+^{(-ij)},S_-^{(-ij)}|b^{(-ij)},\underline{m},\underline{n}}\Big\{\nu\Big(i,j,\underline{m},\underline{n},b^{(-ij)},S_+^{(-ij)},S_-^{(-ij)}\Big)\Big\}\Big\}, \qquad (4.9)
\end{aligned}$$

where

$$\nu\Big(i,j,\underline{m},\underline{n},b^{(-ij)},S_+^{(-ij)},S_-^{(-ij)}\Big) \;=\; \mathbf{E}_{b_i,b_j}\Big\{\mathbf{E}_{S_+^{(ij)},S_-^{(ij)}|b,\underline{m},\underline{n},S_+^{(-ij)},S_-^{(-ij)}}\Big\{\ell_{\rm rank}\Big(f_{S_+,S_-},w_i^{b+},w_j^{b-}\Big)+\ell_{\rm rank}\Big(f_{S_+,S_-},w_j^{b+},w_i^{b-}\Big)\Big\}\Big\}. \tag{4.10}$$

Now, fix i, j, $\underline{m}$, $\underline{n}$, $b^{(-ij)}$, $S_+^{(-ij)}$, $S_-^{(-ij)}$. Also, to keep notation concise, let $f_S \equiv f_{S_+,S_-}$. Then we have,

$$\begin{aligned}
&\mathbf{E}_{b_i,b_j}\Big\{\mathbf{E}_{S_+^{(ij)},S_-^{(ij)}|b_i,b_j}\Big\{\ell_{\rm rank}(f_S,w_i^{b+},w_j^{b-})+\ell_{\rm rank}(f_S,w_j^{b+},w_i^{b-})\Big\}\Big\}\\
&= \mathbf{E}_{b_i,b_j}\Big\{\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i,b_j}\Big\{f_S(w_i^{b+})<f_S(w_j^{b-})\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i,b_j}\Big\{f_S(w_i^{b+})=f_S(w_j^{b-})\Big\}\\
&\qquad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i,b_j}\Big\{f_S(w_j^{b+})<f_S(w_i^{b-})\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i,b_j}\Big\{f_S(w_j^{b+})=f_S(w_i^{b-})\Big\}\Big\}\\
&= \frac{1}{4}\Big[\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=1}\Big\{f_S(w_i')<f_S(w_j'')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=1}\Big\{f_S(w_i')=f_S(w_j'')\Big\}\\
&\quad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=1}\Big\{f_S(w_j')<f_S(w_i'')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=1}\Big\{f_S(w_j')=f_S(w_i'')\Big\}\\
&\quad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\Big\{f_S(w_i'')<f_S(w_j')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\Big\{f_S(w_i'')=f_S(w_j')\Big\}\\
&\quad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\Big\{f_S(w_j'')<f_S(w_i')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\Big\{f_S(w_j'')=f_S(w_i')\Big\}\\
&\quad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=0}\Big\{f_S(w_i')<f_S(w_j')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=0}\Big\{f_S(w_i')=f_S(w_j')\Big\}\\
&\quad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=0}\Big\{f_S(w_j'')<f_S(w_i'')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=0}\Big\{f_S(w_j'')=f_S(w_i'')\Big\}\\
&\quad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\Big\{f_S(w_i'')<f_S(w_j'')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\Big\{f_S(w_i'')=f_S(w_j'')\Big\}\\
&\quad +\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\Big\{f_S(w_j')<f_S(w_i')\Big\}+\frac{1}{2}\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\Big\{f_S(w_j')=f_S(w_i')\Big\}\Big]
\end{aligned} \tag{4.11}$$

Now, let $m_i'(S_+)$ denote the number of occurrences of w_i' in S_+, and $m_i''(S_+)$ the number of occurrences of w_i'' in S_+. Similarly, let $n_i'(S_-)$ denote the number of occurrences of w_i' in S_-, and $n_i''(S_-)$ the number of occurrences of w_i'' in S_-. Define $m_j'(S_+)$, $m_j''(S_+)$ and $n_j'(S_-)$, $n_j''(S_-)$ similarly. Let Φ_1 and Φ_2 be the events defined by

$$\begin{aligned}
\Phi_1 &\equiv \Big\{m_i'(S_+)\ge\frac{m_i}{2},m_j'(S_+)\ge\frac{m_j}{2},n_i''(S_-)\ge\frac{n_i}{2},n_j''(S_-)\ge\frac{n_j}{2}\Big\}\\
\Phi_2 &\equiv \Big\{m_i'(S_+)\ge\frac{m_i}{2},m_j''(S_+)\ge\frac{m_j}{2},n_i''(S_-)\ge\frac{n_i}{2},n_j'(S_-)\ge\frac{n_j}{2}\Big\}.
\end{aligned}$$

Then it can be verified that

$$\begin{aligned}
\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=1}\{\Phi_1\} &\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\{\Phi_1\} && (4.12)\\
\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=0}\{\Phi_2\} &\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\{\Phi_2\}\ . && (4.13)
\end{aligned}$$

Thus we have that for any event E,

$$\begin{aligned}
\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=1}\{E\} &\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=1}\{E\cap\Phi_1\}\\
&\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\{E\cap\Phi_1\} \quad \text{(from Eq. (4.12))} && (4.14)\\
\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\{E\} &\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\{E\cap\Phi_1\} && (4.15)\\
\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=0}\{E\} &\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=1,b_j=0}\{E\cap\Phi_2\}\\
&\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\{E\cap\Phi_2\} \quad \text{(from Eq. (4.13))} && (4.16)\\
\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\{E\} &\geq \mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\{E\cap\Phi_2\}\ . && (4.17)
\end{aligned}$$

Applying the facts in Eqs. (4.14–4.17) to Eq. (4.11), we get

$$\begin{aligned}
&\mathbf{E}_{b_i,b_j}\left\{\mathbf{E}_{S_+^{(ij)},S_-^{(ij)}|b_i,b_j}\left\{\ell_{\rm rank}(f_S,w_i^{b+},w_j^{b-})+\ell_{\rm rank}(f_S,w_j^{b+},w_i^{b-})\right\}\right\}\\
&\geq \frac{1}{4}\left[2\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\left\{\Phi_1\right\}+2\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\left\{\Phi_2\right\}\right]. \qquad (4.18)
\end{aligned}$$

Now, given $\underline{m}(S_+)=\underline{m},\underline{n}(S_-)=\underline{n},b_i=0,b_j=0$, we have that $m_i'(S_+)$ is a $\mathrm{Bin}(m_i,(1-\alpha)/2)$ random variable, $m_j'(S_+)$ a $\mathrm{Bin}(m_j,(1-\alpha)/2)$ random variable, $n_i''(S_+)$ a $\mathrm{Bin}(n_i,(1-\alpha)/2)$ random variable, and $n_j''(S_+)$ a $\mathrm{Bin}(n_j,(1-\alpha)/2)$ random variable. Therefore,

$$\begin{aligned}
\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=0}\left\{\Phi_1\right\} &= \mathbf{P}\left\{\mathrm{Bin}\left(m_i,\frac{1-\alpha}{2}\right)\geq\frac{m_i}{2}\right\}\mathbf{P}\left\{\mathrm{Bin}\left(m_j,\frac{1-\alpha}{2}\right)\geq\frac{m_j}{2}\right\}\times\\
&\qquad \mathbf{P}\left\{\mathrm{Bin}\left(n_i,\frac{1-\alpha}{2}\right)\geq\frac{n_i}{2}\right\}\mathbf{P}\left\{\mathrm{Bin}\left(n_j,\frac{1-\alpha}{2}\right)\geq\frac{n_j}{2}\right\}\\
&\geq \frac{1}{4^4}\phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\,, \qquad (4.19)
\end{aligned}$$

where

$$\phi(u,\alpha) \equiv \left(1-\sqrt{1-e^{-(u+1)\alpha^2/(1-\alpha^2)}}\right). \qquad (4.20)$$

(This follows from Lemma 4.1.) A similar analysis gives

$$\mathbf{P}_{S_+^{(ij)},S_-^{(ij)}|b_i=0,b_j=1}\left\{\Phi_2\right\} \geq \frac{1}{4^4}\phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\,. \qquad (4.21)$$

Thus, using Eqs. (4.19) and (4.21) in Eq. (4.18), we get that

$$\begin{aligned}\mathbf{E}_{b_i,b_j}\Big\{\mathbf{E}_{S_+^{(ij)},S_-^{(ij)}|b_i,b_j}\Big\{\ell_{\rm rank}(f_S,w_i^{b+},w_j^{b-})+\ell_{\rm rank}(f_S,w_j^{b+},w_i^{b-})\Big\}\Big\}\\ \geq\ \frac{1}{4^4}\phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\,.\end{aligned}\tag{4.22}$$

From Eq. (4.10), this gives for all i, j, $\underline{m}$, $\underline{n}$, $b^{(-ij)}$, $S_+^{(-ij)}$, $S_-^{(-ij)}$,

$$\nu\Big(i,j,\underline{m},\underline{n},b^{(-ij)},S_+^{(-ij)},S_-^{(-ij)}\Big)\ \geq\ \frac{1}{4^4}\phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\,.\tag{4.23}$$

Combining this with Eq. (4.9), we have that for all $i\neq j$, $\underline{m},\underline{n}$,

$$\begin{aligned}\lambda(i,j,\underline{m},\underline{n})\ &\geq\ \mathbf{E}_{b^{(-ij)}}\Big\{\mathbf{E}_{S_+^{(-ij)},S_-^{(-ij)}|b^{(-ij)},\underline{m},\underline{n}}\Big\{\frac{1}{4^4}\phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\Big\}\Big\}\\ &=\ \frac{1}{4^4}\phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\,.\end{aligned}\tag{4.24}$$

Also, it is clear that for all i, $\underline{m},\underline{n}$,

$$\lambda(i,i,\underline{m},\underline{n})\ =\ 1\ \geq\ \frac{1}{4^4}(\phi(m_i,\alpha))^2(\phi(n_i,\alpha))^2\,.\tag{4.25}$$

Thus, using Eqs. (4.24) and (4.25) in Eq. (4.6), we have

$$\begin{aligned}&\mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+\sim\mathcal{D}_{b+}^m,S_-\sim\mathcal{D}_{b-}^n}\Big\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{S_+,S_-})-R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\Big\}\Big\}\\ &\geq\ \frac{1}{2^9}\frac{\alpha}{r^2}\sum_{i=1}^r\sum_{j=1}^r\sum_{\{\underline{m}\in\Pi_r(m)\}}\sum_{\{\underline{n}\in\Pi_r(n)\}}\Big[\mathbf{P}\Big\{\underline{m}(S_+)=\underline{m}\Big\}\,\mathbf{P}\Big\{\underline{n}(S_-)=\underline{n}\Big\}\ \times\\ &\qquad\qquad\phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\Big]\,.\end{aligned}\tag{4.26}$$

Now, for each $i,j\in\{1,\ldots,r\}$, we have

$$\begin{aligned}&\sum_{\{\underline{m}\in\Pi_r(m)\}}\sum_{\{\underline{n}\in\Pi_r(n)\}}\Big[\mathbf{P}\Big\{\underline{m}(S_+)=\underline{m}\Big\}\,\mathbf{P}\Big\{\underline{n}(S_-)=\underline{n}\Big\}\ \phi(m_i,\alpha)\ \phi(m_j,\alpha)\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\Big]\\ &=\ \Big[\sum_{m_i=0}^m\sum_{m_j=0}^{m-m_i}\mathbf{P}\Big\{m_i(S_+)=m_i\Big\}\,\mathbf{P}\Big\{m_j(S_+)=m_j\Big|m_i(S_+)=m_i\Big\}\ \phi(m_i,\alpha)\ \phi(m_j,\alpha)\Big]\ \times\\ &\quad\Big[\sum_{n_i=0}^n\sum_{n_j=0}^{n-n_i}\mathbf{P}\Big\{n_i(S_-)=n_i\Big\}\,\mathbf{P}\Big\{n_j(S_-)=n_j\Big|n_i(S_-)=n_i\Big\}\ \phi(n_i,\alpha)\ \phi(n_j,\alpha)\Big]\,.\end{aligned}\tag{4.27}$$

$\phi(u,\alpha)$ is a convex function of u. Also, given $m_i(S_+)=m_i$, $m_j(S_+)$ is a $\mathrm{Bin}(m-m_i,1/(r-1))$ random variable, so that $\mathbf{E}\{m_j(S_+)|m_i(S_+)=m_i\}=(m-m_i)/(r-1)$. Therefore, by Jensen's

inequality,

$$\sum_{m_j=0}^{m-m_i} \mathbf{P}\Big\{m_j(S_+) = m_j \Big| m_i(S_+) = m_i\Big\}\ \phi(m_j,\alpha) \quad \geq \quad \phi\bigg(\frac{m-m_i}{r-1},\alpha\bigg) \quad \geq \quad \phi\bigg(\frac{2m}{r},\alpha\bigg)\,,$$

since $\phi(u,\alpha)$ is also a decreasing function of u. Similarly, $m_i(S_+)$ is a $\mathrm{Bin}(m,1/r)$ random variable, so that $\mathbf{E}\{m_i(S_+)\} = m/r$, and so again by Jensen's inequality,

$$\sum_{m_i=0}^{m} \mathbf{P}\Big\{m_i(S_+) = m_i\Big\}\ \phi(m_i,\alpha) \quad \geq \quad \phi\bigg(\frac{m}{r},\alpha\bigg) \quad \geq \quad \phi\bigg(\frac{2m}{r},\alpha\bigg)\,.$$

The sums over n_i and n_j in Eq. (4.27) can be bounded below in a similar way. Combining with Eq. (4.26), this gives for all $\alpha \in (0,1)$,

$$\begin{aligned}
&\mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+\sim\mathcal{D}_{b+}^m, S_-\sim\mathcal{D}_{b-}^n}\Big\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{S_+,S_-}) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\Big\}\Big\} \\
&\quad\geq\quad \frac{1}{2^9}\frac{\alpha}{r^2}\sum_{i=1}^{r}\sum_{j=1}^{r}\bigg(\phi\bigg(\frac{2m}{r},\alpha\bigg)\bigg)^2\bigg(\phi\bigg(\frac{2n}{r},\alpha\bigg)\bigg)^2 \\
&\quad=\quad \frac{\alpha}{2^9}\Big(1-\sqrt{1-e^{-(2m/r+1)\alpha^2/(1-\alpha^2)}}\Big)^2\Big(1-\sqrt{1-e^{-(2n/r+1)\alpha^2/(1-\alpha^2)}}\Big)^2.
\end{aligned}$$

Assuming $m+n \geq 2r$ and setting $\alpha^2 = r/(m+n)$ then gives

$$\begin{aligned}
&\mathbf{E}_{b\sim\mathcal{U}}\Big\{\mathbf{E}_{S_+\sim\mathcal{D}_{b+}^m, S_-\sim\mathcal{D}_{b-}^n}\Big\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{S_+,S_-}) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\Big\}\Big\} \\
&\quad\geq\quad \frac{1}{2^9}\sqrt{\frac{r}{m+n}}\Big(1-\sqrt{1-e^{-(2m/(m+n)+1)}}\Big)^2\Big(1-\sqrt{1-e^{-(2n/(m+n)+1)}}\Big)^2.
\end{aligned}$$

This proves the theorem. □

Corollary 4.1 *Let $\mathcal{F}$ be a class of ranking functions on $\mathcal{X}$ with* rank-dim$(\mathcal{F}) = r$*, and let $\mathcal{A}$ be any learning algorithm for $\mathcal{F}$. Then $\mathcal{A}$ has sample complexity*

$$M_{\mathcal{A}}(\epsilon,\delta,\rho) \quad \geq \quad \frac{r}{2^{18}(\epsilon+\delta)^2}\left(1-\sqrt{1-e^{-(2\rho+1)}}\right)^4\left(1-\sqrt{1-e^{-(2(1-\rho)+1)}}\right)^4.$$

Proof Let $\rho \in (0,1)\cup\mathbb{Q}$ and $\epsilon,\delta \in (0,1)$. Let $M = M_{\mathcal{A}}(\epsilon,\delta,\rho)$, and let $m = \rho M$, $n = (1-\rho)M$. Then for all distributions $\mathcal{D}_+, \mathcal{D}_-$ on $\mathcal{X}$,

$$\mathbf{P}_{S_+\sim\mathcal{D}_+^m, S_-\sim\mathcal{D}_-^n}\Big\{R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+,S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \ \geq\ \epsilon\Big\} \quad \leq \quad \delta\,.$$

Using the fact that any $[0,1]$-valued random variable Z satisfies $\mathbf{E}\{Z\} \leq \mathbf{P}\{Z \geq \epsilon\} + \epsilon$ for all $\epsilon \in (0,1)$, we thus get that for all distributions $\mathcal{D}_+, \mathcal{D}_-$ on $\mathcal{X}$,

$$\mathbf{E}_{S_+ \sim \mathcal{D}_+^m, S_- \sim \mathcal{D}_-^n} \left\{ R_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{A}(S_+,S_-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \right\} \quad \leq \quad \epsilon + \delta \,.$$

Theorem 4.2 then implies that

$$\epsilon + \delta \quad \geq \quad \frac{1}{2^9} \sqrt{\frac{r}{M}} \left(1 - \sqrt{1 - e^{-(2\rho+1)}}\right)^2 \left(1 - \sqrt{1 - e^{-(2(1-\rho)+1)}}\right)^2 .$$

Solving for M gives the desired result. □

As in the case of the upper bound, the lower bound on sample complexity grows larger as the proportion of positive examples ρ departs from $1/2$.

Corollary 4.2 *Let $\mathcal{F}$ be a class of ranking functions on $\mathcal{X}$. If $\mathcal{F}$ is learnable, then* rank-dim$(\mathcal{F})$ *is finite.*

Proof This follows directly from Corollary 4.1. □

Example 4.4 *Let $\mathcal{F}$ be the class of all ranking functions $f : \mathbb{R} \to \mathbb{R}$ on $\mathbb{R}$. Then clearly, $\mathcal{F}$ rank-shatters arbitrarily large sets of pairs of instances in $\mathbb{R}$. The rank dimension of $\mathcal{F}$ is therefore infinite, and hence by Corollary 4.2, $\mathcal{F}$ is not learnable.*

Remark 4.1 *We note that since the distributions constructed in the proof of Theorem 4.2 do not correspond to a target function, the lower bound on sample complexity and the necessary condition for learnability derived above do not apply to learning in the restricted model of Definition 4.2.*

4.5 Computational Complexity

So far, we have viewed a learning algorithm as simply a function that maps training samples to ranking functions, and have focused only on the sample complexity of this function. However, in order to be of practical use, this function must also be *computable*, *i.e.*, the learning algorithm must truly be an *algorithm* that takes as input a training sample and returns as output a ranking function. Moreover, the learning algorithm must be computationally efficient.

In order to study the computational complexity of learning algorithms for ranking, we need to consider learning at a somewhat broader level than we have done above. In particular, a learning algorithm is usually defined for sets of ranking functions over domains of arbitrary

dimension (*e.g.*, a learning algorithm for the class of linear ranking functions over $\mathbb{R}^d$ for any d), and it is then of interest to study how the computational complexity of the algorithm grows with the dimension. As in [12, 15], we formalize this by defining learning algorithms for *graded* function classes. For each $d \in \mathbb{N}$, let $\mathcal{X}_d$ be a subset of $\mathbb{R}^d$, and let $\mathcal{F}_d$ be a set of ranking functions on $\mathcal{X}_d$. We refer to the union $\mathcal{F} = \bigcup \mathcal{F}_d$ as a *graded* class of ranking functions. A learning algorithm for $\mathcal{F}$ is then a function $\mathcal{A} : \bigcup_{d=1}^{\infty} ((\bigcup_{m=1}^{\infty} \mathcal{X}_d^m) \times (\bigcup_{n=1}^{\infty} \mathcal{X}_d^n)) \to \mathcal{F}$ such that if $(S_+, S_-) \in \mathcal{X}_d^m \times \mathcal{X}_d^n$, then $\mathcal{A}(S_+, S_-) \in \mathcal{F}_d$, and for each d, $\mathcal{A}$ is a learning algorithm for $\mathcal{F}_d$ (in the sense of Definition 4.1). Assuming that learning algorithms are computable functions, we can now ask how the computational complexity of a learning algorithm $\mathcal{A}$ for a graded class of ranking functions $\mathcal{F} = \bigcup \mathcal{F}_d$ grows with d.

Definition 4.6 (Efficient learnability) *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions and let $\mathcal{A}$ be a learning algorithm for $\mathcal{F}$. We say that $\mathcal{A}$ is* efficient *if*

(i) the worst-case time complexity $T_{\mathcal{A}}(m, n, d)$ of $\mathcal{A}$ on samples $(S_+, S_-) \in \mathcal{X}_d^m \times \mathcal{X}_d^n$ is polynomial[1] in m, n and d, and

(ii) the sample complexity $M_{\mathcal{A}}(\epsilon, \delta, \rho, d)$ of $\mathcal{A}$ on $\mathcal{F}_d$ is polynomial in $1/\epsilon$, $1/\delta$, $1/\rho(1-\rho)$ and d (up to an $\lceil \cdot \rceil_\rho$ operation).

We say $\mathcal{F}$ is efficiently learnable *if there is an efficient learning algorithm for $\mathcal{F}$.*

Efficient learnability in the restricted model can be defined in a similar manner. The sufficient and necessary conditions for learnability established in Sections 4.3 and 4.4 can be extended to efficient learnability as follows.

Definition 4.7 (Efficient EEM algorithm) *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions. An* efficient EEM algorithm *for $\mathcal{F}$ is an algorithm that takes as input a sample $(S_+, S_-) \in \mathcal{X}_d^m \times \mathcal{X}_d^n$, and in time polynomial in m, n and d, returns a ranking function $f \in \mathcal{F}_d$ such that $\hat{R}_{S_+,S_-}(f) = \min_{g \in \mathcal{F}_d} \hat{R}_{S_+,S_-}(g)$.*

Theorem 4.3 *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions, and suppose that there exist functions $c_1 : \mathbb{N} \to \mathbb{R}^+$, $c_2 : \mathbb{N} \to \mathbb{R}^+ \cup \{0\}$ such that $r(\mathcal{F}_d, m, n) \leq c_1(d)(mn)^{c_2(d)}$ for all $d, m, n \in \mathbb{N}$, and such that $c_2(d)$ is polynomial in d. Then any efficient EEM algorithm for $\mathcal{F}$ is an efficient learning algorithm for $\mathcal{F}$.*

Proof Suppose that $\mathcal{A}$ is an efficient EEM algorithm for $\mathcal{F}$. Then

[1] In the logarithmic cost model of computation [6], the time complexity is also allowed to depend polynomially on the number of bits required to represent the input.

(i) by Theorem 4.1, $\mathcal{A}$ is a learning algorithm for $\mathcal{F}_d$ for each d and therefore a learning algorithm for $\mathcal{F}$,

(ii) by Definition 4.7, the time complexity $T_{\mathcal{A}}(m, n, d)$ of $\mathcal{A}$ on $\mathcal{F}_d$ is polynomial in m, n and d, and

(iii) by Theorem 4.1, the sample complexity $M_{\mathcal{A}}(\epsilon, \delta, \rho, d)$ of $\mathcal{A}$ on $\mathcal{F}_d$ is polynomial in $1/\epsilon$, $1/\delta$, $1/\rho(1-\rho)$ and d (up to an $\lceil\cdot\rceil_\rho$ operation).

Thus, $\mathcal{A}$ is an efficient learning algorithm for $\mathcal{F}$. □

Theorem 4.4 *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions. If there is an efficient learning algorithm for $\mathcal{F}$, then* rank-dim$(\mathcal{F}_d)$ *is polynomial in d.*

Proof This follows directly from Definition 4.6 and Corollary 4.1. □

Theorem 4.3 shows that, under appropriate conditions on the bipartite rank-shatter coefficients, the existence of an efficient EEM algorithm for a graded class of ranking functions $\mathcal{F} = \bigcup \mathcal{F}_d$ is sufficient for efficient learnability of $\mathcal{F}$. Conversely, it can be shown that if $\mathcal{F}$ is efficiently learnable, then there is an efficient randomized EEM algorithm for $\mathcal{F}$. The proof is similar to that for classification [12]; we omit the details.

Definition 4.8 (Efficient randomized EEM algorithm) *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions. An* efficient randomized EEM algorithm *for $\mathcal{F}$ is a randomized algorithm $\mathcal{A}$ such that, given any sample $(S_+, S_-) \in \mathcal{X}_d^m \times \mathcal{X}_d^n$, $\mathcal{A}$ halts in time polynomial in m, n and d and outputs a ranking function $f \in \mathcal{F}_d$ which, with probability at least $1/2$, satisfies $\hat{R}_{S_+,S_-}(f) = \min_{g \in \mathcal{F}_d} \hat{R}_{S_+,S_-}(g)$.*

Theorem 4.5 *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions. If there is an efficient learning algorithm for $\mathcal{F}$, then there is an efficient randomized EEM algorithm for $\mathcal{F}$.*

Next we define the following decision problem associated with a graded ranking function class $\mathcal{F} = \bigcup \mathcal{F}_d$. As in the case of classification [12], it can be shown that if this problem is NP-hard, then, assuming RP $\neq$ NP, $\mathcal{F}$ is not efficiently learnable. The proof is based on the result of Theorem 4.5 and is similar to that for classification; again, we omit the details.

$\mathcal{F}$-FIT
Instance: $(S_+, S_-) \in \mathcal{X}_d^m \times \mathcal{X}_d^n$ and an integer $k \in \{1, \ldots, mn\}$.
Question: Is there $f \in \mathcal{F}_d$ such that $\hat{R}_{S_+,S_-}(f) \leq k/mn$?

Theorem 4.6 *Let $\mathcal{F}$ be a graded class of ranking functions. If there is an efficient learning algorithm for $\mathcal{F}$, then there is a polynomial-time randomized algorithm*[2] *for $\mathcal{F}$-*FIT*, i.e., $\mathcal{F}$-*FIT *is in* RP.

Corollary 4.3 *Suppose* RP $\neq$ NP*, and let $\mathcal{F}$ be a graded class of ranking functions. If $\mathcal{F}$-*FIT *is* NP*-hard, then $\mathcal{F}$ is not efficiently learnable.*

We now have the formal tools necessary to study the computational complexity of learning ranking functions. Below we use these tools to investigate the computational complexity of learning for the commonly used classes of linear and polynomial ranking functions. Our first result is a hardness result for linear ranking functions.

Theorem 4.7 *Let $\mathcal{F}_{\text{lin}} = \bigcup \mathcal{F}_{\text{lin}(d)}$, where $\mathcal{F}_{\text{lin}(d)}$ is the class of linear ranking functions on $\mathbb{R}^d$. If* RP $\neq$ NP*, then $\mathcal{F}_{\text{lin}}$ is not efficiently learnable.*

Proof We show that $\mathcal{F}_{\text{lin}}$-FIT is NP-hard; the result then follows by Corollary 4.3. To show that $\mathcal{F}_{\text{lin}}$-FIT is NP-hard, we give a reduction from an NP-hard classification problem to $\mathcal{F}_{\text{lin}}$-FIT. For each $d \in \mathbb{N}$, let

$$\mathcal{H}_{\text{lin}(d)} = \{h : \mathbb{R}^d \to \{-1, 0, 1\} \mid h(\mathbf{x}) = \text{sign}(\textstyle\sum_{l=1}^{d} w_l x_l + \theta) \text{ for some } \mathbf{w} \in \mathbb{R}^d, \theta \in \mathbb{R}\}\,.$$

Let $\mathcal{H}_{\text{lin}} = \bigcup \mathcal{H}_{\text{lin}(d)}$, and define the following decision problem associated with $\mathcal{H}_{\text{lin}}$ (where we use the extended definition of empirical classification error given by Eqs. (3.8) and (3.6)):

$\mathcal{H}_{\text{lin}}$-FIT
Instance: $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)) \in (\mathbb{R}^d \times \{-1, 1\})^m$ and an integer $k' \in \{1, \ldots, m\}$.
Question: Is there $h \in \mathcal{H}_{\text{lin}(d)}$ such that $\hat{L}_S(h) \leq k'/m$?

Using exactly the same construction as that used to show the NP-hardness of a similar decision problem relating to linear threshold functions for binary classification [12], it can be shown that the problem $\mathcal{H}_{\text{lin}}$-FIT defined above is NP-hard. We give now a reduction from $\mathcal{H}_{\text{lin}}$-FIT to $\mathcal{F}_{\text{lin}}$-FIT.

Let $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)) \in (\mathbb{R}^d \times \{-1, 1\})^m$, $k' \in \{1, \ldots, m\}$ be an instance of $\mathcal{H}_{\text{lin}}$-FIT. We construct from S, k' an instance $(S_+, S_-) \in (\mathbb{R}^{d+1})^m \times (\mathbb{R}^{d+1}), k \in \{1, \ldots, m\}$ of $\mathcal{F}_{\text{lin}}$-FIT as follows. For each $i \in \{1, \ldots, m\}$, define $x_i^+ = (\mathbf{x}_i, 1) \in \mathbb{R}^{d+1}$ if $y_i = 1$, and $x_i^+ = (-\mathbf{x}_i, -1) \in \mathbb{R}^{d+1}$ if $y_i = -1$. Define $x_1^- = \mathbf{0} \in \mathbb{R}^{d+1}$. Let $S_+ = (x_1^+, \ldots, x_m^+)$, $S_- = (x_1^-)$, and $k = k'$. We claim that the answer to $\mathcal{F}_{\text{lin}}$-FIT on the instance $(S_+, S_-), k$ thus constructed is

[2]Recall that a randomized algorithm $\mathcal{A}$ solves a decision problem Π if $\mathcal{A}$ always halts and produces an output – either 'yes' or 'no' – such that if the answer to Π on the given instance is 'no', the output of $\mathcal{A}$ is 'no', and if the answer to Π on the given instance is 'yes', then with probability at least 1/2, the output of $\mathcal{A}$ is 'yes'.

the same as the answer to $\mathcal{H}_{\text{lin}}$-FIT on the given instance S, k', *i.e.*, that there exists $h \in \mathcal{H}_{\text{lin}(d)}$ with $\hat{L}_S(h) \leq k'/m$ if and only if there exists $f \in \mathcal{F}_{\text{lin}(d+1)}$ with $\hat{R}_{S_+,S_-}(f) \leq k/m$.

First, suppose there exists $h \in \mathcal{H}_{\text{lin}(d)}$ with $\hat{L}_S(h) \leq k'/m$, given by $h(\mathbf{x}) = \text{sign}(\sum_{l=1}^{d} w_l x_l + \theta)$ for some $\mathbf{w} \in \mathbb{R}^d, \theta \in \mathbb{R}$. Define $f : \mathbb{R}^{d+1} \to \mathbb{R}$ as $f(\mathbf{x}) = \sum_{l=1}^{d} w_l x_l + \theta x_{d+1}$ for all $\mathbf{x} \in \mathbb{R}^{d+1}$. Then clearly, $f \in \mathcal{F}_{\text{lin}(d+1)}$. Furthermore, it can be verified that for each $i \in \{1, \ldots, m\}$, $\ell_{\text{rank}}(f, x_i^+, x_1^-) = \ell_{\text{class}}(h, \mathbf{x}_i, y_i)$, and therefore we get $\hat{R}_{S_+,S_-}(f) = \hat{L}_S(h) \leq k'/m = k/m$.

Conversely, suppose there exists $f \in \mathcal{F}_{\text{lin}(d+1)}$ with $\hat{R}_{S_+,S_-}(f) \leq k/m$, given by $f(\mathbf{x}) = \sum_{l=1}^{d+1} w_l x_l + \theta$ for some $\mathbf{w} \in \mathbb{R}^{d+1}, \theta \in \mathbb{R}$. Define $h : \mathbb{R}^d \to \{-1, 0, 1\}$ as $h(\mathbf{x}) = \text{sign}(\sum_{l=1}^{d} w_l x_l + w_{d+1})$ for all $\mathbf{x} \in \mathbb{R}^d$. Then clearly, $h \in \mathcal{H}_{\text{lin}(d)}$. Furthermore, it can be verified that for each $i \in \{1, \ldots, m\}$, $\ell_{\text{class}}(h, \mathbf{x}_i, y_i) = \ell_{\text{rank}}(f, x_i^+, x_1^-)$, and therefore we get $\hat{L}_S(h) = \hat{R}_{S_+,S_-}(f) \leq k/m = k'/m$.

Since the time required to construct the instance $(S_+, S_-), k$ from S, k' is polynomial in the size of S, k', we conclude that $\mathcal{F}_{\text{lin}}$-FIT is NP-hard. □

Our next result shows that $\mathcal{F}_{\text{lin}}$ is efficiently learnable in the restricted learning model. We first specialize Definition 4.7 and Theorem 4.3 to the restricted model case.

Definition 4.9 (Efficient consistent-hypothesis-finder) *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions. An* efficient consistent-hypothesis-finder *for $\mathcal{F}$ is an algorithm $\mathcal{A}$ such that, given any sample $(S_+, S_-) \in \mathcal{X}_d^m \times \mathcal{X}_d^n$ for which there exists a target function $t \in \mathcal{F}_d$ satisfying $\hat{R}_{S_+,S_-}(t) = 0$, $\mathcal{A}$ halts in time polynomial in m, n and d and returns a ranking function $f \in \mathcal{F}_d$ such that $\hat{R}_{S_+,S_-}(f) = 0$.*

Theorem 4.8 *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions, and suppose that there exist functions $c_1 : \mathbb{N} \to \mathbb{R}^+$, $c_2 : \mathbb{N} \to \mathbb{R}^+ \cup \{0\}$ such that $r(\mathcal{F}_d, m, n) \leq c_1(d)(mn)^{c_2(d)}$ for all $d, m, n \in \mathbb{N}$, and such that $c_2(d)$ is polynomial in d. Then any efficient consistent-hypothesis-finder for $\mathcal{F}$ is an efficient learning algorithm for $\mathcal{F}$ in the restricted model.*

Theorem 4.9 *The class of linear ranking functions $\mathcal{F}_{\text{lin}} = \bigcup \mathcal{F}_{\text{lin}(d)}$ is efficiently learnable in the restricted model.*

Proof As discussed in Example 2 (Section 4.3), $r(\mathcal{F}_{\text{lin}(d)}, m, n) \leq (2emn/d)^d$ for all $d, m, n \in \mathbb{N}$. Therefore, by Theorem 4.8, it suffices to show the existence of an efficient consistent-hypothesis-finder for $\mathcal{F}_{\text{lin}}$.

Let $(S_+, S_-) \in (\mathbb{R}^d)^m \times (\mathbb{R}^d)^n$ be a training sample for which there exists a target function $t \in \mathcal{F}_{\text{lin}(d)}$ satisfying $\hat{R}_{S_+,S_-}(t) = 0$. We formulate a linear program whose solution gives a ranking function $f \in \mathcal{F}_{\text{lin}(d)}$ such that $\hat{R}_{S_+,S_-}(f) = 0$. In particular, consider the following linear program in $2d + 1$ variables $w_1, \ldots, w_d, \xi_1, \ldots, \xi_d, \gamma$, with $mn + 2d + 1$ linear constraints:

Maximize γ subject to

$$\begin{aligned}
\textstyle\sum_{l=1}^{d} w_l(x_{il}^+ - x_{jl}^-) &\geq \gamma && \text{for } i \in \{1,\ldots,m\},\ j \in \{1,\ldots,n\} \\
w_l &\leq \xi_l && \text{for } l \in \{1,\ldots,d\} \\
-w_l &\leq \xi_l && \text{for } l \in \{1,\ldots,d\} \\
\xi_1 + \ldots + \xi_d &\leq 1\,. &&
\end{aligned}$$

The feasible region for this linear program is non-empty, since $w_1 = \ldots = w_d = \xi_1 = \ldots = \xi_d = \gamma = 0$ satisfies the constraints. When (S_+, S_-) corresponds to a target function in $\mathcal{F}_{\text{lin}(d)}$, the solution to this linear program has $\gamma > 0$, and in this case, the ranking function $f \in \mathcal{F}_{\text{lin}(d)}$ defined by $f(\mathbf{x}) = \sum_{l=1}^{d} w_l x_l$ clearly satisfies $\hat{R}_{S_+,S_-}(f) = 0$. (The variables $\xi_1, \ldots, \xi_d$ are auxiliary variables; the constraints involving these variables simply enforce the condition $|w_1| + \ldots + |w_d| \leq 1$, thus ensuring that the feasible region is bounded.) Solving the above linear program using a polynomial-time linear programming algorithm such as Karmarkar's [54] therefore constitutes an efficient consistent-hypothesis-finder for $\mathcal{F}_{\text{lin}}$. □

Remark 4.2 *We note that since the polynomial time bound for linear programming algorithms such as Karmarkar's holds only in the logarithmic cost model of computation, the above proof establishes efficient learnability of $\mathcal{F}_{\text{lin}}$ in the restricted learning model only under this model of computation.*

Remark 4.3 *In the above proof, we could also have used a linear program that finds a classification function $h \in \mathcal{H}_{\text{lin}(d)}$ of the form $h(\mathbf{x}) = \text{sign}(\sum_{l=1}^{d} w_l x_l + \theta)$ such that $\hat{L}_S(h) = 0$, where $S = ((x_1^+, 1), \ldots, (x_m^+, 1), (x_1^-, -1), \ldots, (x_n^-, -1))$, and then taken f to be the linear function $f(\mathbf{x}) = \sum_{l=1}^{d} w_l x_l$.*

Finally, we show that learning linear ranking functions over Boolean domains is hard even in the restricted model.

Theorem 4.10 *Let $\mathcal{F}_{\text{lin}}^b = \bigcup \mathcal{F}_{\text{lin}(d)}^b$, where $\mathcal{F}_{\text{lin}(d)}^b$ is the class of linear ranking functions on $\{0,1\}^d$. If RP $\neq$ NP, then $\mathcal{F}_{\text{lin}}^b$ is not efficiently learnable in the restricted model.*

Proof Let, if possible, $\mathcal{F}_{\text{lin}}^b$ be efficiently learnable in the restricted model. Then by Theorem 4.5, there is an efficient randomized consistent-hypothesis-finder $\mathcal{A}$ for $\mathcal{F}_{\text{lin}}^b$. Clearly, $\mathcal{A}$ can be used to construct an efficient randomized consistent-hypothesis-finder for $\mathcal{H}_{\text{lin}}^b = \bigcup \mathcal{H}_{\text{lin}(d)}^b$, where $\mathcal{H}_{\text{lin}(d)}^b$ is the class of Boolean threshold functions on $\{0,1\}^d$. This, in turn, implies the existence of an efficient learning algorithm for $\mathcal{H}_{\text{lin}}^b$ in the restricted (PAC) model (see [12]).

Since the problem of learning Boolean threshold functions in the PAC model is known to be NP-hard [81], this implies RP = NP. Thus, if RP $\neq$ NP, then $\mathcal{F}^b_{\text{lin}}$ is not efficiently learnable in the restricted model. □

The techniques used above can be used also to establish that for any $q \in \mathbb{N}$, the class $\mathcal{F}_{\text{poly}(q)} = \bigcup \mathcal{F}_{\text{poly}(d,q)}$, where $\mathcal{F}_{\text{poly}(d,q)}$ is the class of polynomial ranking functions on $\mathbb{R}^d$ with degree at most q, is not efficiently learnable in the standard model, but is efficiently learnable in the restricted model, and that the class $\mathcal{F}^b_{\text{poly}(q)} = \bigcup \mathcal{F}^b_{\text{poly}(d,q)}$, where $\mathcal{F}^b_{\text{poly}(d,q)}$ is the class of polynomial ranking functions on $\{0,1\}^d$ with degree at most q, is not efficiently learnable even in the restricted model.

4.6 Conclusions and Open Questions

Our goal in this chapter has been to initiate a formal study of learnability for ranking functions. We have defined a model of learnability for bipartite ranking functions, and have derived a number of results in this model. In particular, we have established both a sufficient condition for learnability of a class of ranking functions, expressed in terms of the bipartite rank-shatter coefficients defined in Chapter 3, and a necessary condition for learnability, expressed in terms of a new combinatorial parameter that we have termed the rank dimension. We have also initiated a study of the computational complexity of learning ranking functions.

There are several questions to be answered. First, is there a single quantity that characterizes learnability of a class of ranking functions, analogous to the VC dimension for classification and the fat-shattering dimension for regression? For example, based on our results, an upper bound of the form $r(\mathcal{F}, m, n) = O((mn)^{\text{rank-dim}(\mathcal{F})})$ on the bipartite rank-shatter coefficients would establish the rank dimension as such a quantity. Second, for what other classes of ranking functions can efficient learning algorithms or hardness results be shown? Finally, for what other settings of the ranking problem can learnability be studied?

Chapter 5

Stability and Generalization of Bipartite Ranking Algorithms

5.1 Introduction

The study of generalization properties of learning algorithms, *i.e.*, the extent to which their performance on training data is indicative of their expected performance on future data, has been a central focus in learning theory research. The results of Vapnik and Chervonenkis [102], in which generalization bounds for classification algorithms were derived based on uniform convergence, were perhaps the first in this direction. Since then, a large number of different tools have been developed for studying generalization, and have been applied successfully to analyze algorithms for both classification and regression. It is natural to ask whether these tools can be adapted to study generalization properties of ranking algorithms. We derived in Chapter 3 a generalization bound for bipartite ranking algorithms based on uniform convergence. In this chapter, we ask whether such a result can be obtained using the notion of algorithmic stability, which has been used recently to derive generalization bounds for classification and regression algorithms, and which offers a different viewpoint than uniform convergence [17, 63].

The question of the generalization behaviour of ranking algorithms has only recently begun to be addressed. Generalization properties of algorithms for a distinct but closely related problem were considered in [48]. Freund et al. [37] were the first to derive generalization bounds for ranking; as discussed in Chapter 3, they derived a generalization bound for the bipartite RankBoost algorithm based on uniform convergence. Their bound was derived from a uniform convergence result for the classification error, and was expressed in terms of the VC-dimension of a class of binary classification functions derived from the class of ranking functions searched by RankBoost. The bound we derived in Chapter 3 is also based on uniform convergence and is expressed in terms of a new set of combinatorial parameters that measure directly the complexity of the class of ranking functions searched by an algorithm.

Uniform convergence requires the empirical errors of all functions in the searched class to converge to their expected errors. Generalization bounds based on uniform convergence are therefore necessarily loose, as they therefore depend only on properties of the function class being searched, and do not take into account the manner in which the function class is actually searched by the algorithm. In addition, these bounds can be applied only to algorithms that search function classes of bounded complexity.

The notion of algorithmic stability, first studied for learning algorithms by Devroye and Wagner [32], has been used recently to directly obtain generalization bounds, without needing to show uniform convergence, for classification and regression algorithms that satisfy certain stability conditions [17, 63]. In particular, a stable learning algorithm is one whose output does not change much with small changes in the training sample; the above studies have shown that classification and regression algorithms that satisfy this condition have good generalization properties. The resulting stability-based bounds depend on properties of the algorithm rather than the function class that is searched, and can be applied also to algorithms that search function classes of unbounded complexity. Algorithms that have been shown to be stable include, for example, kernel-based classification and regression algorithms such as support vector machines (SVMs), which often cannot be analyzed using uniform convergence tools. We show in this chapter that the notion of algorithmic stability can be used also to analyze the generalization behaviour of (bipartite) ranking algorithms.

We define notions of stability for bipartite ranking algorithms (Section 5.2), and use these notions to analyze the generalization behaviour of such algorithms. In particular, we derive generalization bounds for bipartite ranking algorithms that exhibit good stability properties (Section 5.3). We show that kernel-based ranking algorithms that perform regularization in a reproducing kernel Hilbert space (RKHS) have such stability properties, and therefore our bounds can be applied to these algorithms (Section 5.4); this is in contrast with bounds based on uniform convergence, which in many cases cannot be applied to these algorithms. A comparison of the bounds we obtain with corresponding bounds for classification algorithms yields some interesting insights into the difference in generalization behaviour between ranking and classification. In particular, we find that for a training sample of M elements containing m positive and $n = M - m$ negative instances, the sample size M in the classification bounds is replaced with the quantity $mn/(m+n)$ in the ranking bounds. If we define the 'positive skew' of the sample as the proportion of positive examples $\rho = m/(m+n)$, then this means that the 'effective' sample size in ranking is reduced from M to $\rho(1-\rho)M$, with the reduction being more drastic as ρ departs from $1/2$, *i.e.*, as the balance between positive and negative examples becomes more uneven.

5.2 Stability of Bipartite Ranking Algorithms

A bipartite ranking algorithm takes as input a training sample $(S_+, S_-) \in (\bigcup_{m=1}^{\infty} \mathcal{X}^m) \times (\bigcup_{n=1}^{\infty} \mathcal{X}^n)$ and returns as output a ranking function $f_{S_+,S_-} : \mathcal{X} \to \mathbb{R}$. For simplicity, we consider only deterministic algorithms. We are concerned in this chapter with generalization properties of such algorithms; in particular, we are interested in bounding the expected error of a learned ranking function in terms of an empirically observable quantity such as its empirical error on the training sample from which it is learned. We use a slightly different notation in this chapter for the expected and empirical errors of a ranking function, leaving the distributions in the expected error implicit:

$$\begin{aligned} R(f) &\equiv R_{\mathcal{D}_+,\mathcal{D}_-}(f) \\ \hat{R}(f; S_+, S_-) &\equiv \hat{R}_{S_+,S_-}(f)\,. \end{aligned}$$

The following definitions will be useful in our study.

Definition 5.1 (Ranking loss function) *A* ranking loss function *is a function $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ that assigns, for each $f : \mathcal{X} \to \mathbb{R}$ and $x, x' \in \mathcal{X}$, a non-negative real number $\ell(f, x, x')$ that is interpreted as the loss suffered by f in its relative ranking of x and x'.*

Definition 5.2 (Expected ℓ-error) *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$. Let $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ be a ranking loss function. Define the* expected ℓ-error *of f, denoted by $R_\ell(f)$, as*

$$R_\ell(f) = \mathbf{E}_{x^+ \sim \mathcal{D}_+, x^- \sim \mathcal{D}_-} \left\{ \ell(f, x^+, x^-) \right\}.$$

Definition 5.3 (Empirical ℓ-error) *Let $f : \mathcal{X} \to \mathbb{R}$ be a ranking function on $\mathcal{X}$, and let $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ be a finite sample. Let $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ be a ranking loss function. Define the* empirical ℓ-error *of f with respect to S_+ and S_-, denoted by $\hat{R}_\ell(f; S_+, S_-)$, as*

$$\hat{R}_\ell(f; S_+, S_-) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell(f, x_i^+, x_j^-)\,.$$

Comparing with Definitions 1.9 and 1.10, we see that the bipartite ranking error can be expressed as the ℓ_{rank}-error, *i.e.*, $R \equiv R_{\ell_{\text{rank}}}$ and $\hat{R} \equiv \hat{R}_{\ell_{\text{rank}}}$ (where ℓ_{rank} is the bipartite ranking loss given by Definition 1.8).

A stable algorithm is one whose output does not change significantly with small changes in the input. The input to a bipartite ranking algorithm is a training sample of the form

$(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ for some $m, n \in \mathbb{N}$; we consider changes to such a sample that consist of replacing a single element of the sample with a new instance. As in Chapter 2, for any $i \in \{1, \ldots, m\}$ and $z \in \mathcal{X}$, we use $S_+^{i,z}$ to denote the sequence obtained from S_+ by replacing x_i^+ with z, and for any $j \in \{1, \ldots, n\}$ and $z \in \mathcal{X}$, we use $S_-^{j,z}$ to denote the sequence obtained from S_- by replacing x_j^- with z.

Several different notions of stability have been used in the study of classification and regression algorithms [32, 55, 17, 63, 82]. The notions of stability that we define for (bipartite) ranking algorithms below are most closely related to those used by Bousquet and Elisseeff [17].

Definition 5.4 (Uniform loss stability) *Let $\mathcal{A}$ be a bipartite ranking algorithm whose output on a training sample (S_+, S_-) we denote by f_{S_+,S_-}, and let $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ be a ranking loss function. Let $\alpha : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$, $\beta : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$. We say that $\mathcal{A}$ has* uniform loss stability (α, β) *with respect to ℓ if for all $m, n \in \mathbb{N}$, $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, $z \in \mathcal{X}$, $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$, we have for all $x^+, x^- \in \mathcal{X}$,*

$$\begin{aligned} |\ell(f_{S_+,S_-}, x^+, x^-) - \ell(f_{S_+^{i,z},S_-}, x^+, x^-)| &\leq \alpha(m,n)\,, \\ |\ell(f_{S_+,S_-}, x^+, x^-) - \ell(f_{S_+,S_-^{j,z}}, x^+, x^-)| &\leq \beta(m,n)\,. \end{aligned}$$

Definition 5.5 (Uniform score stability) *Let $\mathcal{A}$ be a bipartite ranking algorithm whose output on a training sample (S_+, S_-) we denote by f_{S_+,S_-}. Let $\mu : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$, $\nu : \mathbb{N} \times \mathbb{N} \to \mathbb{R}$. We say that $\mathcal{A}$ has* uniform score stability (μ, ν) *if for all $m, n \in \mathbb{N}$, $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, $z \in \mathcal{X}$, $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$, we have for all $x \in \mathcal{X}$,*

$$\begin{aligned} |f_{S_+,S_-}(x) - f_{S_+^{i,z},S_-}(x)| &\leq \mu(m,n)\,, \\ |f_{S_+,S_-}(x) - f_{S_+,S_-^{j,z}}(x)| &\leq \nu(m,n)\,. \end{aligned}$$

5.3 Generalization Bounds for Stable Ranking Algorithms

In this section we derive generalization bounds for ranking algorithms that exhibit good stability properties. Our methods are based on those of Bousquet and Elisseeff [17], who derived such bounds for classification and regression algorithms. We note that our results are all distribution-free, in the sense that they hold for all distributions $\mathcal{D}_+$ and $\mathcal{D}_-$ on $\mathcal{X}$. We start with the following technical lemma.

Lemma 5.1 *Let $\mathcal{A}$ be a symmetric bipartite ranking algorithm*[1] *whose output on a training sample $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ we denote by f_{S_+,S_-}, and let $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ be a*

[1] A symmetric bipartite ranking algorithm is one whose output is independent of the order of elements in the training sequences S_+ and S_-.

ranking loss function. Then for all $i \in \{1,\ldots,m\}$, $j \in \{1,\ldots,n\}$, we have

$$\begin{aligned}
&\mathbf{E}_{S_+,S_-}\left\{R_\ell(f_{S_+,S_-}) - \hat{R}_\ell(f_{S_+,S_-};S_+,S_-)\right\} \\
&\quad = \quad \mathbf{E}_{S_+,S_-,x^+,x^-}\left\{\ell(f_{S_+,S_-},x^+,x^-) - \ell(f_{S_+^{i,x^+},S_-^{j,x^-}},x^+,x^-)\right\}.
\end{aligned}$$

Proof We have,

$$\mathbf{E}_{S_+,S_-}\left\{\hat{R}_\ell(f_{S_+,S_-};S_+,S_-)\right\} \quad = \quad \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{E}_{S_+,S_-}\left\{\ell(f_{S_+,S_-},x_i^+,x_j^-)\right\}.$$

By symmetry, the term in the summation is the same for all i,j. Therefore, for each $i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots,n\}$, we get

$$\begin{aligned}
\mathbf{E}_{S_+,S_-}\left\{\hat{R}_\ell(f_{S_+,S_-};S_+,S_-)\right\} &= \mathbf{E}_{S_+,S_-}\left\{\ell(f_{S_+,S_-},x_i^+,x_j^-)\right\} \\
&= \mathbf{E}_{S_+,S_-,x^+,x^-}\left\{\ell(f_{S_+,S_-},x_i^+,x_j^-)\right\}.
\end{aligned}$$

Interchanging the roles of x_i^+ with x^+ and x_j^- with x^-, we get

$$\mathbf{E}_{S_+,S_-}\left\{\hat{R}_\ell(f_{S_+,S_-};S_+,S_-)\right\} \quad = \quad \mathbf{E}_{S_+,S_-,x^+,x^-}\left\{\ell(f_{S_+^{i,x^+},S_-^{j,x^-}},x^+,x^-)\right\}.$$

Since by definition

$$\mathbf{E}_{S_+,S_-}\left\{R_\ell(f_{S_+,S_-})\right\} \quad = \quad \mathbf{E}_{S_+,S_-,x^+,x^-}\left\{\ell(f_{S_+,S_-},x^+,x^-)\right\},$$

the result follows. □

We are now ready to give our main result, which bounds the expected ℓ-error of a ranking function learned by an algorithm with good uniform loss stability in terms of its empirical ℓ-error on the training sample. Our main tool will again be McDiarmid's inequality [73] which was used in Chapter 2 (Theorem 2.1).

Theorem 5.1 *Let $\mathcal{A}$ be a symmetric bipartite ranking algorithm whose output on a training sample $(S_+,S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ we denote by f_{S_+,S_-}, and let $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ be a ranking loss function such that $0 \le \ell(f,x,x') \le B$ for all $f : \mathcal{X}\to\mathbb{R}$ and $x,x' \in \mathcal{X}$. Let $\alpha : \mathbb{N}\times\mathbb{N}\to\mathbb{R}$, $\beta : \mathbb{N}\times\mathbb{N}\to\mathbb{R}$ be such that $\mathcal{A}$ has uniform loss stability (α,β) with respect to ℓ. Then for any $0<\delta<1$, with probability at least $1-\delta$ over the draw of (S_+,S_-),*

$$\begin{aligned}
R_\ell(f_{S_+,S_-}) \quad < \quad & \hat{R}_\ell(f_{S_+,S_-};S_+,S_-) + \alpha(m,n) + \beta(m,n) \\
& + \sqrt{\frac{\{n(2m\,\alpha(m,n)+B)^2 + m(2n\,\beta(m,n)+B)^2\}\ln(1/\delta)}{2mn}}.
\end{aligned}$$

Proof Let $\phi : \mathcal{X}^m \times \mathcal{X}^n \to \mathbb{R}$ be defined as

$$\phi(S_+, S_-) \;=\; R_\ell(f_{S_+,S_-}) - \hat{R}_\ell(f_{S_+,S_-}; S_+, S_-)\,.$$

We shall show that ϕ satisfies the conditions of McDiarmid's inequality (Theorem 2.1). Let $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, and let $z \in \mathcal{X}$. For each $i \in \{1, \ldots, m\}$, we have

$$\begin{aligned}
\left|R_\ell(f_{S_+,S_-}) - R_\ell(f_{S_+^{i,z},S_-})\right| &= \left|\mathbf{E}_{x^+,x^-}\left\{\ell(f_{S_+,S_-}, x^+, x^-) - \ell(f_{S_+^{i,z},S_-}, x^+, x^-)\right\}\right| \\
&\le \mathbf{E}_{x^+,x^-}\left\{\left|\ell(f_{S_+,S_-}, x^+, x^-) - \ell(f_{S_+^{i,z},S_-}, x^+, x^-)\right|\right\} \\
&\le \alpha(m,n)\,,
\end{aligned}$$

and

$$\begin{aligned}
&\left|\hat{R}_\ell(f_{S_+,S_-}; S_+, S_-) - \hat{R}_\ell(f_{S_+^{i,z},S_-}; S_+^{i,z}, S_-)\right| \\
&\le \frac{1}{mn}\sum_{i'=1,i'\neq i}^{m}\sum_{j=1}^{n}\left|\ell(f_{S_+,S_-}, x_{i'}^+, x_j^-) - \ell(f_{S_+^{i,z},S_-}, x_{i'}^+, x_j^-)\right| \\
&\quad + \frac{1}{mn}\sum_{j=1}^{n}\left|\ell(f_{S_+,S_-}, x_i^+, x_j^-) - \ell(f_{S_+^{i,z},S_-}, z, x_j^-)\right| \\
&\le \alpha(m,n) + \frac{B}{m}\,.
\end{aligned}$$

This gives

$$\left|\phi(S_+, S_-) - \phi(S_+^{i,z}, S_-)\right| \;\le\; 2\alpha(m,n) + \frac{B}{m}\,.$$

Similarly, it can be shown that for each $j \in \{1, \ldots, n\}$,

$$\left|\phi(S_+, S_-) - \phi(S_+, S_-^{j,z})\right| \;\le\; 2\beta(m,n) + \frac{B}{n}\,.$$

Thus, applying McDiarmid's inequality to ϕ, we get for any $\epsilon > 0$,

$$\begin{aligned}
&\mathbf{P}_{S_+,S_-}\left\{\left\{R_\ell(f_{S_+,S_-}) - \hat{R}_\ell(f_{S_+,S_-}; S_+, S_-)\right\} - \mathbf{E}_{S_+,S_-}\left\{R_\ell(f_{S_+,S_-}) - \hat{R}_\ell(f_{S_+,S_-}; S_+, S_-)\right\} \ge \epsilon\right\} \\
&\qquad \le e^{-2\epsilon^2/(m(2\alpha(m,n)+B/m)^2 + n(2\beta(m,n)+B/n)^2)} \\
&\qquad = e^{-2mn\epsilon^2/(n(2m\alpha(m,n)+B)^2 + m(2n\beta(m,n)+B)^2)}\,.
\end{aligned}$$

Now, by Lemma 5.1, we have

$$\begin{aligned}
&\mathbf{E}_{S_+,S_-}\left\{R_\ell(f_{S_+,S_-}) - \hat{R}_\ell(f_{S_+,S_-};S_+,S_-)\right\} \\
&\quad= \mathbf{E}_{S_+,S_-,x^+,x^-}\left\{\ell(f_{S_+,S_-},x^+,x^-) - \ell(f_{S_+^{i,x^+},S_-^{j,x^-}},x^+,x^-)\right\} \\
&\quad= \mathbf{E}_{S_+,S_-,x^+,x^-}\Big\{\ell(f_{S_+,S_-},x^+,x^-) - \ell(f_{S_+^{i,x^+},S_-},x^+,x^-) \\
&\qquad\qquad + \ell(f_{S_+^{i,x^+},S_-},x^+,x^-) - \ell(f_{S_+^{i,x^+},S_-^{j,x^-}},x^+,x^-)\Big\} \\
&\quad\leq \mathbf{E}_{S_+,S_-,x^+,x^-}\left\{\left|\ell(f_{S_+,S_-},x^+,x^-) - \ell(f_{S_+^{i,x^+},S_-},x^+,x^-)\right|\right\} \\
&\qquad + \mathbf{E}_{S_+,S_-,x^+,x^-}\left\{\left|\ell(f_{S_+^{i,x^+},S_-},x^+,x^-) - \ell(f_{S_+^{i,x^+},S_-^{j,x^-}},x^+,x^-)\right|\right\} \\
&\quad\leq \alpha(m,n) + \beta(m,n)\,.
\end{aligned}$$

Thus we get for any $\epsilon > 0$,

$$\begin{aligned}
&\mathbf{P}_{S_+,S_-}\left\{R_\ell(f_{S_+,S_-}) - \hat{R}_\ell(f_{S_+,S_-};S_+,S_-) - (\alpha(m,n)+\beta(m,n)) \geq \epsilon\right\} \\
&\qquad\leq e^{-2mn\epsilon^2/(n(2m\alpha(m,n)+B)^2+m(2n\beta(m,n)+B)^2)}\,.
\end{aligned}$$

The result follows by setting the right hand side equal to δ and solving for ϵ. □

Theorem 5.1 gives meaningful bounds when $\alpha(m,n) = o(1/\sqrt{m})$ and $\beta(m,n) = o(1/\sqrt{n})$. This means the theorem cannot be applied directly to obtain bounds on the expected ranking error, since it is not possible to have non-trivial uniform loss stability with respect to the bipartite ranking loss ℓ_{rank} (except by an algorithm that picks the same ranking function for all training samples of a given size m,n). However, for any ranking loss ℓ that satisfies $\ell_{\text{rank}} \leq \ell$, Theorem 5.1 can be applied to ranking algorithms that have good uniform loss stability with respect to ℓ to obtain bounds on the expected ℓ-error; since in this case $R \leq R_\ell$, these bounds apply also to the expected ranking error. We consider below a specific ranking loss that satisfies this condition.

For $\gamma > 0$, let the γ *ranking loss*, denoted by ℓ_γ, be defined as follows:

$$\ell_\gamma(f,x,x') = \begin{cases} 1 & \text{if } (f(x) - f(x')) \leq 0 \\ 1 - \frac{(f(x)-f(x'))}{\gamma} & \text{if } 0 < (f(x) - f(x')) < \gamma \\ 0 & \text{if } (f(x) - f(x')) \geq \gamma \end{cases}. \tag{5.1}$$

Clearly, for all $\gamma > 0$, we have $\ell_{\text{rank}} \leq \ell_\gamma$. Therefore, for any ranking algorithm that has good uniform loss stability with respect to ℓ_γ for some $\gamma > 0$, Theorem 5.1 can be applied to bound the expected ranking error of a learned ranking function in terms of its empirical ℓ_γ-error on

the training sample. The following lemma shows that, for every $\gamma > 0$, a ranking algorithm that has good uniform score stability also has good uniform loss stability with respect to ℓ_γ.

Lemma 5.2 *Let $\mathcal{A}$ be a bipartite ranking algorithm whose output on a training sample (S_+, S_-) we denote by f_{S_+,S_-}. Let $\mu : \mathbb{N}\times\mathbb{N}\to\mathbb{R}$, $\nu : \mathbb{N}\times\mathbb{N}\to\mathbb{R}$ be such that $\mathcal{A}$ has uniform score stability (μ, ν). Then for every $\gamma > 0$, $\mathcal{A}$ has uniform loss stability $(\alpha_\gamma, \beta_\gamma)$ with respect to the γ ranking loss ℓ_γ, where for all $m, n \in \mathbb{N}$,*

$$\alpha_\gamma(m,n) = \frac{2\mu(m,n)}{\gamma}\,, \qquad \beta_\gamma(m,n) = \frac{2\nu(m,n)}{\gamma}\,.$$

Proof By the definition of ℓ_γ in Eq. (5.1), we have that

$$\begin{aligned} \ell_\gamma(f,x,x') &\leq 1 - \tfrac{(f(x)-f(x')}{\gamma} \quad \text{if } (f(x)-f(x')) \leq 0\,, &(5.2)\\ \ell_\gamma(f,x,x') &\geq 1 - \tfrac{(f(x)-f(x')}{\gamma} \quad \text{if } (f(x)-f(x')) \geq \gamma\,. &(5.3)\end{aligned}$$

Now, let $m, n \in \mathbb{N}$, $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, $z \in \mathcal{X}$, $i \in \{1,\ldots,m\}$ and $j \in \{1,\ldots,n\}$, and let $x^+, x^- \in \mathcal{X}$. The case $\ell_\gamma(f_{S_+,S_-}, x^+, x^-) = \ell_\gamma(f_{S_+^{i,z},S_-}, x^+, x^-)$ is trivial. Assume $\ell_\gamma(f_{S_+,S_-}, x^+, x^-) \neq \ell_\gamma(f_{S_+^{i,z},S_-}, x^+, x^-)$. Then, using the observations in Eqs. (5.2–5.3), it can be verified that

$$\begin{aligned} &\left|\ell_\gamma(f_{S_+,S_-}, x^+, x^-) - \ell_\gamma(f_{S_+^{i,z},S_-}, x^+, x^-)\right| \\ &\leq \left|\left(1 - \frac{(f_{S_+,S_-}(x^+) - f_{S_+,S_-}(x^-))}{\gamma}\right) - \left(1 - \frac{(f_{S_+^{i,z},S_-}(x^+) - f_{S_+^{i,z},S_-}(x^-))}{\gamma}\right)\right| \\ &\leq \frac{1}{\gamma}\Big(\left|f_{S_+,S_-}(x^+) - f_{S_+^{i,z},S_-}(x^+)\right| + \left|f_{S_+,S_-}(x^-) - f_{S_+^{i,z},S_-}(x^-)\right|\Big) \\ &\leq \frac{2\mu(m,n)}{\gamma}\,. \end{aligned}$$

Similarly, it can be shown that

$$\left|\ell_\gamma(f_{S_+,S_-}, x^+, x^-) - \ell_\gamma(f_{S_+,S_-^{j,z}}, x^+, x^-)\right| \leq \frac{2\nu(m,n)}{\gamma}\,.$$

The result follows. □

Putting everything together, we thus get the following result which bounds the expected ranking error of a learned ranking function in terms of its empirical ℓ_γ-error for any ranking algorithm that has good uniform score stability.

Theorem 5.2 *Let $\mathcal{A}$ be a symmetric bipartite ranking algorithm whose output on a training sample $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$ we denote by f_{S_+,S_-}. Let $\mu : \mathbb{N}\times\mathbb{N}\to\mathbb{R}$, $\nu : \mathbb{N}\times\mathbb{N}\to\mathbb{R}$ be such that $\mathcal{A}$ has uniform score stability (μ, ν), and let $\gamma > 0$. Then for any $0 < \delta < 1$, with probability at least $1 - \delta$ over the draw of (S_+, S_-),*

$$\begin{aligned} R(f_{S_+,S_-}) \;<\; & \hat{R}_{\ell_\gamma}(f_{S_+,S_-}; S_+, S_-) + \frac{2\mu(m,n)}{\gamma} + \frac{2\nu(m,n)}{\gamma} \\ & + \sqrt{\frac{\left\{n\left(\frac{4m\,\mu(m,n)}{\gamma}+1\right)^2 + m\left(\frac{4n\,\nu(m,n)}{\gamma}+1\right)^2\right\}\ln(1/\delta)}{2mn}}\,. \end{aligned}$$

Proof The result follows by applying Theorem 5.1 to $\mathcal{A}$ with the ranking loss ℓ_γ (using Lemma 5.2), which satisfies $0 \le \ell_\gamma \le 1$, and from the fact that $R \le R_{\ell_\gamma}$. □

5.4 Stable Ranking Algorithms

In this section we show stability of certain ranking algorithms that select a ranking function by minimizing a regularized objective function. We start by deriving a general result for regularization-based ranking algorithms in Section 5.4.1. In Section 5.4.2 we use this result to show stability of kernel-based ranking algorithms that perform regularization in a reproducing kernel Hilbert space. We show, in particular, stability of an SVM-like ranking algorithm, and apply the results of Section 5.3 to obtain a generalization bound for this algorithm. Again, our methods are based on those of Bousquet and Elisseeff [17], who showed similar results for classification and regression algorithms.

5.4.1 General Regularizers

Given a ranking loss function $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$, a class $\mathcal{F}$ of real-valued functions on $\mathcal{X}$, and a regularization functional $N : \mathcal{F}\to\mathbb{R}^+ \cup \{0\}$, consider the following regularized empirical ℓ-error of a ranking function $f \in \mathcal{F}$ (with respect to a sample $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$), with regularization parameter $\lambda > 0$:

$$\hat{R}^\lambda_\ell(f; S_+, S_-) \;=\; \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\ell(f, x_i^+, x_j^-) + \lambda N(f)\,. \tag{5.4}$$

We consider bipartite ranking algorithms that minimize such a regularized objective function, *i.e.*, ranking algorithms that, given a training sample (S_+, S_-), output a ranking function

$f_{S_+,S_-} \in \mathcal{F}$ that satisfies

$$\begin{aligned} f_{S_+,S_-} &= \arg\min_{f\in\mathcal{F}} \hat{R}^{\lambda}_{\ell}(f; S_+, S_-) \\ &= \arg\min_{f\in\mathcal{F}} \left\{ \hat{R}_{\ell}(f; S_+, S_-) + \lambda N(f) \right\}, \end{aligned} \tag{5.5}$$

for some fixed choice of ranking loss ℓ, function class $\mathcal{F}$, regularizer N, and regularization parameter λ. We derive below a general result that will be useful for showing stability of such regularization-based algorithms.

Definition 5.6 (σ-admissibility) *Let $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ be a ranking loss and $\mathcal{F}$ a class of real-valued functions on $\mathcal{X}$. Let $\sigma > 0$. We say that ℓ is* σ-admissible *with respect to $\mathcal{F}$ if for all $f_1, f_2 \in \mathcal{F}$ and all $x, x' \in \mathcal{X}$, we have*

$$|\ell(f_1, x, x') - \ell(f_2, x, x')| \;\leq\; \sigma\Big(|f_1(x) - f_2(x)| + |f_1(x') - f_2(x')|\Big).$$

Lemma 5.3 *Let $\ell : \mathbb{R}^{\mathcal{X}} \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+ \cup \{0\}$ be a ranking loss such that $\ell(f, x, x')$ is convex in f. Let $\mathcal{F}$ be a convex class of real-valued functions on $\mathcal{X}$, and let $\sigma > 0$ be such that ℓ is σ-admissible with respect to $\mathcal{F}$. Let $\lambda > 0$, and let $N : \mathcal{F} \to \mathbb{R}^+ \cup \{0\}$ be a functional defined on $\mathcal{F}$ such that for all samples $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, the regularized empirical ℓ-error $\hat{R}^{\lambda}_{\ell}(f; S_+, S_-)$ has a minimum (not necessarily unique) in $\mathcal{F}$. Let $\mathcal{A}$ be a ranking algorithm defined by Eq. (5.5), and let $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, $z \in \mathcal{X}$, $i \in \{1, \ldots, m\}$, and $j \in \{1, \ldots, n\}$. For brevity, denote*

$$f \;\equiv\; f_{S_+,S_-}, \qquad f_+^{i,z} \;\equiv\; f_{S_+^{i,z},S_-}, \qquad f_-^{j,z} \;\equiv\; f_{S_+,S_-^{j,z}},$$

and let

$$\Delta f_+ \;=\; (f_+^{i,z} - f), \qquad \Delta f_- \;=\; (f_-^{j,z} - f).$$

Then we have that for any $t \in [0, 1]$,

$$\begin{aligned} &N(f) - N(f + t\Delta f_+) + N(f_+^{i,z}) - N(f_+^{i,z} - t\Delta f_+) \\ &\qquad\leq\; \frac{t\sigma}{\lambda mn} \sum_{j=1}^{n} \Big(|\Delta f_+(x_i^+)| + 2|\Delta f_+(x_j^-)| + |\Delta f_+(z)| \Big), \end{aligned}$$

and

$$\begin{aligned} &N(f) - N(f + t\Delta f_-) + N(f_-^{j,z}) - N(f_-^{j,z} - t\Delta f_-) \\ &\qquad\leq\; \frac{t\sigma}{\lambda mn} \sum_{i=1}^{m} \Big(|\Delta f_-(x_j^-)| + 2|\Delta f_-(x_i^+)| + |\Delta f_-(z)| \Big). \end{aligned}$$

Proof Recall that a convex function $\phi : \mathcal{U} \to \mathbb{R}$ satisfies for all $u, v \in \mathcal{U}$ and for all $t \in [0,1]$,

$$\phi(u + t(v-u)) - \phi(u) \;\leq\; t(\phi(v) - \phi(u))\,.$$

Since $\ell(f, x, x')$ is convex in f, we have that $\hat{R}_\ell(f; S_+, S_-)$ is convex in f. Therefore for any $t \in [0,1]$, we have

$$\hat{R}_\ell(f + t\Delta f_+; S_+, S_-) - \hat{R}_\ell(f; S_+, S_-) \;\leq\; t\Big(\hat{R}_\ell(f_+^{i,z}; S_+, S_-) - \hat{R}_\ell(f; S_+, S_-)\Big)\,, \tag{5.6}$$

and also (interchanging the roles of f and $f_+^{i,z}$),

$$\hat{R}_\ell(f_+^{i,z} - t\Delta f_+; S_+, S_-) - \hat{R}_\ell(f_+^{i,z}; S_+, S_-) \;\leq\; t\Big(\hat{R}_\ell(f; S_+, S_-) - \hat{R}_\ell(f_+^{i,z}; S_+, S_-)\Big)\,, \tag{5.7}$$

Adding Eqs. (5.6) and (5.7), we get

$$\hat{R}_\ell(f + t\Delta f_+; S_+, S_-) - \hat{R}_\ell(f; S_+, S_-) + \hat{R}_\ell(f_+^{i,z} - t\Delta f_+; S_+, S_-) - \hat{R}_\ell(f_+^{i,z}; S_+, S_-) \;\leq\; 0\,. \tag{5.8}$$

Now, since $\mathcal{F}$ is convex, we have that $(f + t\Delta f_+) \in \mathcal{F}$ and $(f_+^{i,z} - t\Delta f_+) \in \mathcal{F}$. Since f minimizes $\hat{R}_\ell^\lambda(f; S_+, S_-)$ in $\mathcal{F}$ and $f_+^{i,z}$ minimizes $\hat{R}_\ell^\lambda(f; S_+^{i,z}, S_-)$ in $\mathcal{F}$, we thus have

$$\hat{R}_\ell^\lambda(f; S_+, S_-) - \hat{R}_\ell^\lambda(f + t\Delta f_+; S_+, S_-) \;\leq\; 0\,, \tag{5.9}$$

$$\hat{R}_\ell^\lambda(f_+^{i,z}; S_+^{i,z}, S_-) - \hat{R}_\ell^\lambda(f_+^{i,z} - t\Delta f_+; S_+^{i,z}, S_-) \;\leq\; 0\,. \tag{5.10}$$

Adding Eqs. (5.8), (5.9) and (5.10), we get

$$\begin{aligned}
&\lambda\Big(N(f) - N(f + t\Delta f_+) + N(f_+^{i,z}) - N(f_+^{i,z} - t\Delta f_+)\Big) \\
&\quad\leq\; \hat{R}_\ell(f_+^{i,z}; S_+^{i,z}, S_-) - \hat{R}_\ell(f_+^{i,z}; S_+ S_-) + \hat{R}_\ell(f_+^{i,z} - t\Delta f_+; S_+, S_-) - \hat{R}_\ell(f_+^{i,z} - t\Delta f_+; S_+^{i,z}, S_-) \\
&\quad=\; \frac{1}{mn}\sum_{j=1}^{n}\Big(\ell(f_+^{i,z}, z, x_j^-) - \ell(f_+^{i,z}, x_i^+, x_j^-) + \ell(f_+^{i,z} - t\Delta f_+, x_i^+, x_j^-) - \ell(f_+^{i,z} - t\Delta f_+, z, x_j^-)\Big) \\
&\quad\leq\; \frac{t\sigma}{mn}\sum_{j=1}^{n}\Big(|\Delta f_+(x_i^+)| + 2|\Delta f_+(x_j^-)| + |\Delta f_+(z)|\Big)\,,
\end{aligned}$$

by σ-admissibility. This proves the first inequality. The second inequality can be proved similarly. □

As we show below, the above result can be used to establish stability of certain regularization-based ranking algorithms.

5.4.2 Regularization in Hilbert Spaces

Let $\mathcal{F}$ be a reproducing kernel Hilbert space (RKHS) with kernel K. That is, $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric, positive definite function, and $\mathcal{F}$ is the completion of the linear space $\mathcal{F}_0$ of real-valued functions on $\mathcal{X}$ generated by the functions $\{K_x \mid x \in \mathcal{X}\}$, where

$$K_x(x') \;=\; K(x, x')\,.$$

In other words, $\mathcal{F}$ is the completion of

$$\mathcal{F}_0 \;=\; \left\{ f : \mathcal{X} \to \mathbb{R} \;\middle|\; f = \sum_{i=1}^{m} c_i K_{x_i} \;\text{ for some }\; m \in \mathbb{N}, c_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}.$$

An inner product $\langle \cdot, \cdot \rangle_K$ is defined on $\mathcal{F}_0$ as follows: if $f = \sum_{i=1}^{m} c_i K_{x_i}$ and $g = \sum_{j=1}^{n} d_j K_{x'_j}$, then

$$\langle f, g \rangle_K \;=\; \sum_{j=1}^{n} d_j f(x'_j) \;=\; \sum_{i=1}^{m} \sum_{j=1}^{n} c_i d_j K(x_i, x'_j) \;=\; \sum_{i=1}^{m} c_i g(x_i)\,.$$

It is easily verified that this inner product satisfies the following *reproducing* property: for all $f \in \mathcal{F}_0$ and all $x \in \mathcal{X}$,

$$\langle f, K_x \rangle_K \;=\; f(x)\,. \tag{5.11}$$

This gives for all $x, x' \in \mathcal{X}$,

$$\langle K_x, K_{x'} \rangle_K \;=\; K(x, x')\,. \tag{5.12}$$

The norm of a function $f \in \mathcal{F}_0$ is defined as

$$\|f\|_K \;=\; \sqrt{\langle f, f \rangle_K}\,. \tag{5.13}$$

It can be verified that if $\mathcal{F}_0$ is completed to form $\mathcal{F}$ (*i.e.*, the limits of all Cauchy sequences in $\mathcal{F}_0$ are added to the space) and its inner product extended to $\mathcal{F}$, then Eqs. (5.11), (5.12) and (5.13) hold for all $f \in \mathcal{F}$ and $x, x' \in \mathcal{X}$. Applying the Cauchy-Schwartz inequality to Eq. (5.11) then gives for all $f \in \mathcal{F}$ and all $x \in \mathcal{X}$,

$$\begin{aligned} |f(x)| \;&=\; |\langle f, K_x \rangle_K| \\ &\leq\; \|f\|_K \, \|K_x\|_K \\ &=\; \|f\|_K \sqrt{K(x, x)}\,. \end{aligned} \tag{5.14}$$

Further details about RKHSs can be found, for example, in [46, 35]. We shall consider ranking algorithms that perform regularization in the RKHS $\mathcal{F}$ using the squared norm in $\mathcal{F}$ as a regularizer. Specifically, let $N : \mathcal{F} \to \mathbb{R}^+ \cup \{0\}$ be the regularizer defined by

$$N(f) \quad = \quad \|f\|_K^2 \,. \tag{5.15}$$

We show below that, if the kernel K is such that $K(x,x)$ is bounded for all $x \in \mathcal{X}$, then a ranking algorithm that minimizes an appropriate regularized error over $\mathcal{F}$, with regularizer N as defined above, has good uniform score stability.

Theorem 5.3 *Let $\mathcal{F}$ be an RKHS with kernel K such that for all $x \in \mathcal{X}$, $K(x,x) \leq \kappa^2 < \infty$. Let ℓ be a ranking loss such that $\ell(f, x, x')$ is convex in f and ℓ is σ-admissible with respect to $\mathcal{F}$. Let $\lambda > 0$, and let N be given by Eq. (5.15). Let $\mathcal{A}$ be a ranking algorithm defined by Eq. (5.5). Then $\mathcal{A}$ has uniform score stability (μ, ν), where for all $m, n \in \mathbb{N}$,*

$$\mu(m,n) = \frac{4\sigma\kappa^2}{\lambda m}\,, \qquad \nu(m,n) = \frac{4\sigma\kappa^2}{\lambda n}\,.$$

Proof Let $m, n \in \mathbb{N}$, $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, $z \in \mathcal{X}$, and $i \in \{1, \ldots, m\}$. Applying Lemma 5.3 with $t = 1/2$, we get (using the notation of Lemma 5.3) that

$$\begin{aligned} \|f\|_K^2 - \|f + \tfrac{1}{2}\Delta f_+\|_K^2 + \|f_+^{i,z}\|_K^2 - \|f_+^{i,z} - \tfrac{1}{2}\Delta f_+\|_K^2 & \\ \leq \quad \frac{\sigma}{2\lambda m n} \sum_{j=1}^{n} \Big(|\Delta f_+(x_i^+)| + 2|\Delta f_+(x_j^-)| + |\Delta f_+(z)| \Big) \,. & \end{aligned} \tag{5.16}$$

Note that since $\mathcal{F}$ is a vector space, $\Delta f_+ \in \mathcal{F}$, $(f + \frac{1}{2}\Delta f_+) \in \mathcal{F}$, and $(f_+^{i,z} - \frac{1}{2}\Delta f_+) \in \mathcal{F}$, so that $\|f + \frac{1}{2}\Delta f_+\|_K$ and $\|f_+^{i,z} - \frac{1}{2}\Delta f_+\|_K$ are well-defined. Now, we have

$$\begin{aligned} \|f\|_K^2 - \|f + \tfrac{1}{2}\Delta f_+\|_K^2 + \|f_+^{i,z}\|_K^2 - \|f_+^{i,z} - \tfrac{1}{2}\Delta f_+\|_K^2 & \\ = \quad \|f\|_K^2 + \|f_+^{i,z}\|_K^2 - \frac{1}{2}\|f + f_+^{i,z}\|_K^2 & \\ = \quad \frac{1}{2}\|f\|_K^2 + \frac{1}{2}\|f_+^{i,z}\|_K^2 - \langle f, f_+^{i,z}\rangle_K & \\ = \quad \frac{1}{2}\|\Delta f_+\|_K^2 \,. & \end{aligned}$$

Combined with Eq. (5.16), this gives

$$\frac{1}{2}\|\Delta f_+\|_K^2 \quad \leq \quad \frac{\sigma}{2\lambda m n} \sum_{j=1}^{n} \Big(|\Delta f_+(x_i^+)| + 2|\Delta f_+(x_j^-)| + |\Delta f_+(z)| \Big) \,.$$

Since (as noted above) $\Delta f_+ \in \mathcal{F}$, by Eq. (5.14), we thus get that

$$\begin{aligned}\frac{1}{2}\|\Delta f_+\|_K^2 &\leq \frac{\sigma}{2\lambda mn}\|\Delta f_+\|_K \sum_{j=1}^{n}\Big(\sqrt{K(x_i^+, x_i^+)} + 2\sqrt{K(x_j^-, x_j^-)} + \sqrt{K(z,z)}\Big) \\ &\leq \frac{4\sigma\kappa}{2\lambda m}\|\Delta f_+\|_K\,,\end{aligned}$$

which gives

$$\|\Delta f_+\|_K \leq \frac{4\sigma\kappa}{\lambda m}\,. \tag{5.17}$$

Thus, by Eqs. (5.14) and (5.17), we have for all $x \in \mathcal{X}$,

$$|f_{S_+,S_-}(x) - f_{S_+^{i,z},S_-}(x)| = |\Delta f_+(x)| \leq \frac{4\sigma\kappa^2}{\lambda m}\,.$$

Similarly, for each $j \in \{1,\ldots,n\}$, we can show that

$$|f_{S_+,S_-}(x) - f_{S_+,S_-^{j,z}}(x)| \leq \frac{4\sigma\kappa^2}{\lambda n}\,.$$

The result follows. □

Corollary 5.1 *Under the conditions of Theorem 5.3, we have that for any $0 < \delta < 1$, with probability at least $1-\delta$ over the draw of $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, the expected ranking error of the ranking function f_{S_+,S_-} learned by $\mathcal{A}$ is bounded by*

$$R(f_{S_+,S_-}) < \hat{R}_{\ell_1}(f_{S_+,S_-}; S_+, S_-) + \frac{8\sigma\kappa^2}{\lambda}\Big(\frac{m+n}{mn}\Big) + \Big(1 + \frac{16\sigma\kappa^2}{\lambda}\Big)\sqrt{\frac{(m+n)\ln(1/\delta)}{2mn}}\,.$$

Proof The result follows from Theorem 5.3, by applying Theorem 5.2 with $\gamma = 1$. □

Consider now the following ranking loss function, which we refer to as the *hinge ranking loss* due to its similarity to the hinge loss in classification:

$$\ell_h(f, x, x') = \begin{cases} 1 - (f(x) - f(x')) & \text{if } (f(x) - f(x')) < 1 \\ 0 & \text{if } (f(x) - f(x')) \geq 1 \end{cases}\,. \tag{5.18}$$

We consider a ranking algorithm $\mathcal{A}$ that minimizes the regularized ℓ_h-error in an RKHS $\mathcal{F}$. Specifically, let $\mathcal{A}$ be a ranking algorithm which, given a training sample (S_+, S_-), outputs a

ranking function $f_{S_+,S_-} \in \mathcal{F}$ that satisfies (for some fixed $\lambda > 0$)

$$f_{S_+,S_-} \quad = \quad \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_{\ell_h}(f; S_+, S_-) + \lambda \|f\|_K^2 \right\}. \tag{5.19}$$

We note that this algorithm has an equivalent quadratic programming formulation similar to SVMs in the case of classification. In particular, the problem of minimizing $\hat{R}^{\lambda}_{\ell_h}(f; S_+, S_-)$ is equivalent to that of minimizing

$$\frac{1}{2}\|f\|_K^2 + C \sum_{i=1}^{m} \sum_{j=1}^{n} \xi_{ij}$$

subject to

$$\begin{aligned} f(x_i^+) - f(x_j^-) &\geq 1 - \xi_{ij} \qquad \text{for } i \in \{1, \ldots, m\},\ j \in \{1, \ldots, n\} \\ \xi_{ij} &\geq 0 \qquad \text{for } i \in \{1, \ldots, m\},\ j \in \{1, \ldots, n\}, \end{aligned}$$

where $C = 1/(2\lambda mn)$. The dual formulation of this problem obtained by introducing Lagrange multipliers leads to a quadratic program similar to that obtained for SVMs, and has been studied, for example, in [86] (for earlier work on SVMs for ranking problems, see [47, 48]).

It can be verified that $\ell_h(f, x, x')$ is convex in f, and that ℓ_h is 1-admissible with respect to $\mathcal{F}$. Thus, if $K(x, x) \leq \kappa^2$ for all $x \in \mathcal{X}$, then from Corollary 5.1 we get that for any $0 < \delta < 1$, with probability at least $1 - \delta$ over the draw of $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, the expected ranking error of the ranking function f_{S_+,S_-} learned by the above algorithm $\mathcal{A}$ is bounded by

$$R(f_{S_+,S_-}) \quad < \quad \hat{R}_{\ell_1}(f_{S_+,S_-}; S_+, S_-) + \frac{8\kappa^2}{\lambda}\Big(\frac{m+n}{mn}\Big) + \Big(1 + \frac{16\kappa^2}{\lambda}\Big)\sqrt{\frac{(m+n)\ln(1/\delta)}{2mn}}. \tag{5.20}$$

In particular, for the RKHS corresponding to the linear kernel defined on the unit ball in $\mathbb{R}^d$, *i.e.*, for the RKHS corresponding to $\mathcal{X} = \left\{\mathbf{x} \in \mathbb{R}^d \;\middle|\; \|\mathbf{x}\|^2 \leq 1\right\}$ and $K(\mathbf{x}, \mathbf{x}') = \mathbf{x} \cdot \mathbf{x}'$, so that $K(\mathbf{x}, \mathbf{x}) \leq 1$ for all $\mathbf{x} \in \mathcal{X}$, we have that with probability at least $1 - \delta$ over the draw of $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, the ranking function f_{S_+,S_-} learned by the above algorithm (defined by Eq. (5.19)) satisfies

$$R(f_{S_+,S_-}) \quad < \quad \hat{R}_{\ell_1}(f_{S_+,S_-}; S_+, S_-) + \frac{8}{\lambda}\Big(\frac{m+n}{mn}\Big) + \Big(1 + \frac{16}{\lambda}\Big)\sqrt{\frac{(m+n)\ln(1/\delta)}{2mn}}.$$

On the other hand, the confidence interval obtained for this algorithm using the uniform convergence bound of Chapter 3 gives that, with probability at least $1 - \delta$ over the draw of

$(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$,

$$R(f_{S_+,S_-}) \quad < \quad \hat{R}(f_{S_+,S_-}; S_+, S_-) + \sqrt{\frac{8(m+n)(d(\ln(8mn/d)+1)+\ln(4/\delta))}{mn}}\,.$$

The above bounds are plotted in Figure 5.1 for $\delta = 0.01$, $\lambda = 1$, and various values of d and $m/(m+n)$. As can be seen, directly analyzing stability properties of the algorithm gives considerable benefit over the general uniform convergence based analysis. In particular, since the uniform convergence bound depends on the complexity of the function class that is searched, the bound quickly becomes uninformative in high dimensions; on the other hand, the stability bound is independent of the dimensionality of the space. In the case of kernel spaces whose complexity cannot be bounded, *e.g.*, the RKHS corresponding to the Gaussian kernel $K(\mathbf{x}, \mathbf{x}') = e^{-\|\mathbf{x}-\mathbf{x}'\|^2/2}$, the uniform convergence bound cannot be applied at all, while the stability analysis continues to hold.

Comparing the bound derived in Eq. (5.20) to the corresponding bound for classification derived by Bousquet and Elisseeff [17], we find that if the total number of training examples is denoted by $M = m + n$, then the sample size M in their bound is replaced by the quantity $mn/(m+n)$ in our bound.[2] If we define the 'positive skew' of the sample as the proportion of positive examples $\rho = m/(m+n)$, then this is equivalent to replacing M in the classification bound with $\rho(1-\rho)M$ in our bound. The 'effective' sample size in ranking is thus reduced from M to $\rho(1-\rho)M$, the reduction being more drastic as the skew ρ departs from $1/2$, *i.e.*, as the balance between positive and negative examples becomes more uneven. Interestingly, a similar observation holds for the large deviation and uniform convergence bounds for the ranking error derived in Chapters 2 and 3 when compared to corresponding bounds for the classification error.

As in the case of classification [17], the above results show that a larger regularization parameter λ leads to better stability and, therefore, a tighter confidence interval in the resulting generalization bound. In particular, one must have $\lambda \gg \sqrt{(m+n)/mn}$ in order for the above bound to be meaningful.

5.5 Conclusions and Open Questions

We have derived generalization bounds for bipartite ranking algorithms using notions of algorithmic stability. Bounds based on algorithmic stability offer a different viewpoint than bounds

[2] The difference in constants in the two bounds is due in part to the difference in loss functions in ranking and classification, and in part to a slight difference in definitions of stability; in particular, our definitions are in terms of changes to a training sample that consist of replacing one element in the sample with a new one, while the definitions of Bousquet and Elisseeff are in terms of changes that consist of removing one element from the sample.

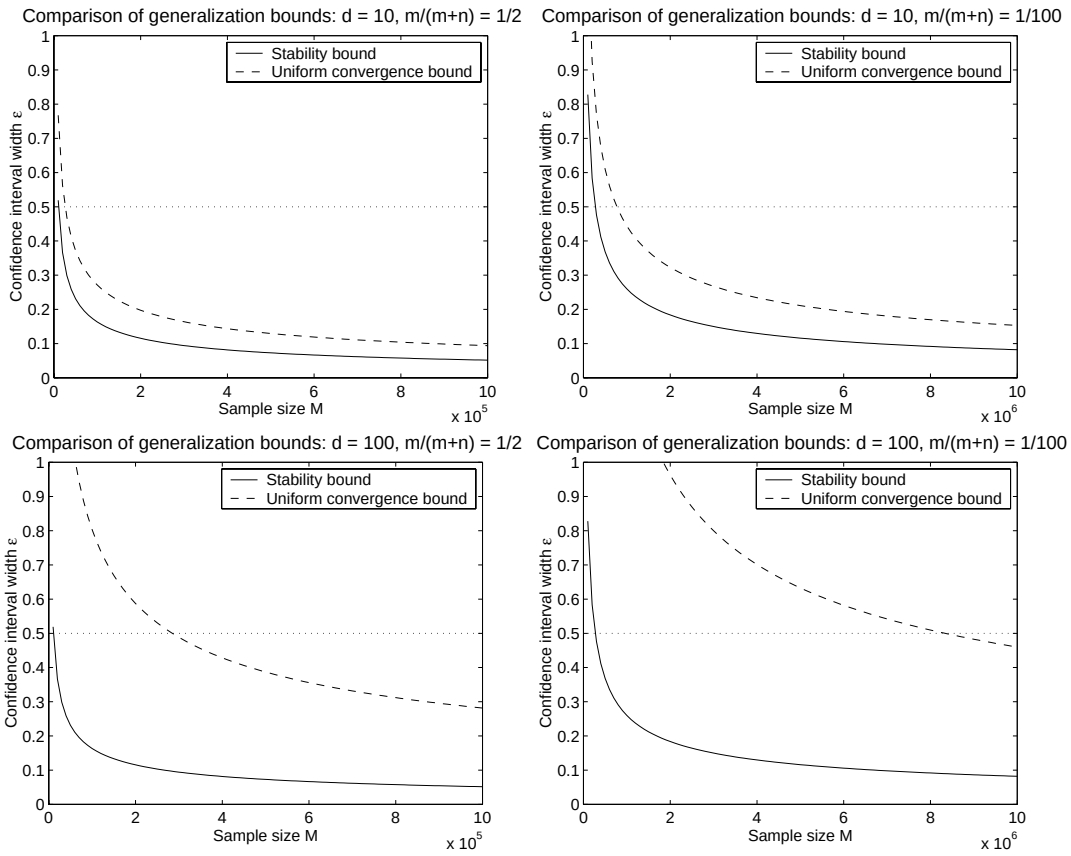

Figure 5.1: A comparison of our stability bound with the uniform convergence bound of Chapter 3 for the kernel-based algorithm described in Section 5.4.2, with a linear kernel over the unit ball in $\mathbb{R}^d$. The plots are for $\delta = 0.01$, $\lambda = 1$, and show how the confidence interval size ϵ given by the two bounds varies with the sample size $M = m + n$, for various values of d and $m/(m+n)$.

based on uniform convergence, and can often be applied where uniform convergence bounds cannot; in particular, we have applied our results to obtain generalization bounds for kernel-based ranking algorithms, to which bounds based on uniform convergence are in many cases inapplicable.

The main difference in the mathematical formulation of the (bipartite) ranking problem as compared to the classification problem is that the loss function in ranking is 'pair-wise' rather than 'point-wise'. The general analysis of ranking is otherwise similar to that for classification, and indeed, ranking algorithms often resemble 'classification on pairs'. However, generalization bounds from classification *cannot* be applied directly to ranking, due to dependences among the instance pairs. Indeed, the bounds we have obtained for ranking suggest that the effective sample size in ranking is not only smaller than the number of positive-negative pairs mn, but

is even smaller than the number of instances $M = m + n$; the dependences reduce the effective sample size to $\rho(1-\rho)M$, where $\rho = m/(m+n)$ is the 'positive skew' of the sample.

The notions of uniform stability studied in this chapter correspond most closely to those studied by Bousquet and Elisseeff [17]. These notions are strict in that they require changes in a sample to have bounded effect uniformly over all samples and replacements. Kutin and Niyogi [63] have derived generalization bounds (for classification and regression algorithms) using a less strict notion of stability termed 'almost-everywhere' stability; this requires changes in a sample to have bounded effect only with high probability (over the draw of the sample and the replacement element). The notion of almost-everywhere stability leads to a distribution-dependent treatment as opposed to the distribution-free treatment obtained with uniform stability, and it would be particularly interesting to see if making distributional assumptions in ranking can mitigate the reduced sample size effect discussed above.

An open question concerns the analysis of other ranking algorithms using the algorithmic stability framework. It has been shown [62] that the AdaBoost algorithm for classification [38] is stability-preserving, in the sense that stability of base classifiers implies stability of the final learned classifier. It would be interesting if a similar result could be shown for the bipartite RankBoost algorithm [37], which is based on the same principles of boosting as AdaBoost.

Finally, it is also an open question to analyze generalization properties of ranking algorithms in other settings of the ranking problem (*i.e.*, other than bipartite).

Chapter 6

Bipartite Ranking in Action: Identifying Genes Related to Cancer

6.1 Introduction

One of the greatest challenges in post-genome medical research is to identify genes that are involved in a particular disease. Identification of such genes would not only provide a better understanding of how a disease works, but would also lead to novel treatments for the disease.

With the widespread adoption of DNA microarray technology, which allows expression levels of thousands of genes to be measured simultaneously, vast amounts of gene expression data are being generated, and it is hoped that computational analysis of this data can help in identification of important genes. In particular, from a biologist's point of view, there is an immediate, unmet need for a computer program that can generate a *ranking* of genes such that genes relevant to the disease under study are likely to appear at the top of the ranking. The proteins corresponding to the top few genes can then be subjected to biological tests to elucidate their structural and functional properties, with a good chance that many of those tested will emerge as targets for the development of new drugs or find use as disease markers.

Although several methods have been proposed to obtain a ranking of genes from expression data, none of the current methods addresses directly the problem of finding genes that are relevant to a given disease. In this chapter, we show that the gene ranking problem can be formulated naturally as a bipartite ranking problem; this allows us to address directly the problem of finding a ranking that places relevant genes at the top and irrelevant ones at the bottom (Section 6.2). We describe experiments using this approach on microarray data sets for two forms of cancer, namely leukemia and colon cancer. An extensive validation with the biomedical literature indicates very promising results on both sets of data, including the identification of some exciting candidate genes, ranked in the top few by our method, as potential targets for drug development (Section 6.3).

6.2 Methods

In this section we describe the formulation of the gene ranking problem as a bipartite ranking problem (Section 6.2.1), the RankBoost algorithm of Freund et al. [37] that we use in our experiments (Section 6.2.2), the data sets we use (Section 6.2.3), and the methodology we use for selecting training data (Section 6.2.4) and for validating our results (Section 6.2.5). Our results are given in Section 6.3.

6.2.1 Formulation as a Bipartite Ranking Problem

The problem of ranking genes based on expression data pertaining to a given disease can be described formally as follows. Let N denote the number of genes whose expression levels are measured, and d the number of biological samples (these could represent, for example, tissue samples from different patients, or different experimental conditions). Let $\mathbf{X} = [x_{ik}] \in \mathbb{R}^{N \times d}$ denote the given expression matrix, where x_{ik} denotes the expression level of gene i in sample k. We shall use $\mathbf{x}_i \in \mathbb{R}^d$ to denote the expression vector of gene i across the d samples, and $\tilde{\mathbf{x}}_k \in \mathbb{R}^N$ to denote the expression vector of sample k across the N genes. The problem of finding a ranking of genes can then be viewed as a problem of finding a ranking function $f : \mathbb{R}^d \to \mathbb{R}$ that assigns a real-valued score $f(\mathbf{x}_i)$ to gene i; gene i is ranked higher than gene j if $f(\mathbf{x}_i) > f(\mathbf{x}_j)$.

A number of methods have been proposed for obtaining such a ranking function f. Most such methods require that the biological samples come from two distinct classes, *i.e.*, that with each sample k there be associated a class label $y_k \in \{-1, 1\}$ denoting its class membership (*e.g.*, disease or normal, or one of two different forms of a disease). The function f is then chosen to rank genes based on their ability to distinguish between the two classes. For example, in [40, 97], the score $f(\mathbf{x}_i)$ assigned to gene i is taken to be some measure of correlation between the gene's expression vector $\mathbf{x}_i$ and the class vector $\mathbf{y}$. In [42, 59], a learning algorithm is used to find a linear classification function $h : \mathbb{R}^N \to \{-1, 1\}$ of the form $h(\tilde{\mathbf{x}}) = \text{sign}(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}} + \theta)$ for some $\tilde{\mathbf{w}} \in \mathbb{R}^N, \theta \in \mathbb{R}$, based on some training examples $S = ((\tilde{\mathbf{x}}_{k_1}, y_{k_1}), \ldots, (\tilde{\mathbf{x}}_{k_m}, y_{k_m})) \in (\mathbb{R}^N \times \{-1, 1\})^m$; the goal in learning the classification function is actually to classify future samples into one of the two classes, but an implicit ranking over the genes is obtained from the weights in the classification function by taking $f(\mathbf{x}_i) = |\tilde{w}_i|$ for each $i \in \{1, \ldots, N\}$.

Although classification of biological samples is an important problem, for the purpose of identifying genes relevant to a given disease, the above methods have several drawbacks. First, in studying a given disease, one may want to include samples representing various types of experiments that may be related to the disease. In this case, the samples cannot be divided into two classes, and the above methods cannot be applied. Second, even when the samples fall into two natural classes, a ranking of genes based on their ability to distinguish between the two

classes may not be the best ranking in terms of relevance to the disease under study. Indeed, it is unlikely that a simple correlation measure with the class vector can accurately determine the relevance of a gene to a disease. In addition, when ranking genes based on a classification function chosen for accurate classification of future samples, important genes can easily be missed; if two or more genes have similar expression patterns, the classification function would typically assign a high weight to only one of them.

It turns out that the problem of ranking genes by their relevance to a given disease can be formulated naturally as a bipartite ranking problem. In this formulation, the biologist provides a few examples $S_+ = (\mathbf{x}_1^+, \ldots, \mathbf{x}_m^+) \in (\mathbb{R}^d)^m$ of (expression vectors corresponding to) genes that are known to be relevant to the disease, and a few examples $S_- = (\mathbf{x}_1^-, \ldots, \mathbf{x}_n^-) \in (\mathbb{R}^d)^n$ of (expression vectors corresponding to) genes known to be irrelevant. Any learning algorithm for the bipartite ranking problem can then be used to automatically learn from these examples a ranking over the remaining genes that tends to place relevant genes at the top and irrelevant ones at the bottom. This formulation of the gene ranking problem exploits existing biological knowledge, in the form of training examples, to address directly the problem of identifying relevant genes. In addition, it does not make any assumptions about the nature of the biological samples; in particular, unlike previous approaches, we do not need to assume that the samples come from two distinct classes.

The learning algorithm we use in our experiments is the bipartite RankBoost algorithm of Freund et al. [37]; we describe this algorithm next.

6.2.2 The RankBoost Algorithm

RankBoost [37] is an algorithm for the ranking problem based on the principles of boosting [91]. The general RankBoost algorithm applies to a more general setting of the ranking problem; we describe here the bipartite RankBoost algorithm that applies to the bipartite setting.

An outline of the bipartite RankBoost algorithm is given in Figure 6.1. The algorithm takes as input a training sample of the form $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$, where $\mathcal{X}$ is the instance space and $S_+ = (x_1^+, \ldots, x_m^+)$, $S_- = (x_1^-, \ldots, x_n^-)$, and produces as output a ranking function $f : \mathcal{X} \to \mathbb{R}$ that is a linear combination of some 'weak' ranking functions chosen from some base class $\mathcal{F}_{\text{base}}$. The algorithm works in rounds and maintains a distribution D_t over the set of positive-negative pairs $\left\{(x_i^+, x_j^-) : i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}\right\}$. On each round t, it chooses a weak ranking $f_t \in \mathcal{F}_{\text{base}}$ and a real number $\alpha_t \in \mathbb{R}$, and updates the distribution D_t such that instance pairs (x_i^+, x_j^-) that are mis-ranked by f_t are weighted more heavily by D_{t+1}, forcing the weak ranking chosen in the next round to focus on these mis-ranked pairs. The extent of the update is determined by α_t (typically, $\alpha_t > 0$). The final ranking is given by a weighted combination of the weak rankings chosen in different rounds.

Algorithm **RankBoost (Bipartite)**

Input: $(S_+, S_-) \in \mathcal{X}^m \times \mathcal{X}^n$.

Initialize: $D_1(x_i^+, x_j^-) = 1/mn$ for all $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$.

For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t; get weak ranking $f_t : \mathcal{X} \to \mathbb{R}$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update: $D_{t+1}(x_i^+, x_j^-) = \dfrac{D_t(x_i^+, x_j^-) \exp\Big(-\alpha(f_t(x_i^+) - f_t(x_j^-))\Big)}{Z_t}$,

  where $Z_t = \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} D_t(x_i^+, x_j^-) \exp\Big(-\alpha(f_t(x_i^+) - f_t(x_j^-))\Big)$.

Output the final ranking: $f(x) = \displaystyle\sum_{t=1}^{T} \alpha_t f_t(x)$.

Figure 6.1: The bipartite RankBoost algorithm [37].

In our gene ranking problem, the instances to be ranked are genes, each represented by a d-dimensional expression vector (where d is the number of biological samples; see Section 6.2.1). The instance space $\mathcal{X}$ in our problem is thus simply $\mathbb{R}^d$. The base function class $\mathcal{F}_{\text{base}}$ we use contains the d coordinate projection functions $f^{(k)} : \mathbb{R}^d \to \mathbb{R}$, given by $f^{(k)}(\mathbf{x}) = x_k$ for each $k \in \{1, \ldots, d\}$. Thus on each round t, our weak learner chooses $f_t : \mathbb{R}^d \to \mathbb{R}$ to be $f_t(\mathbf{x}) = x_{k_t}$ for some $k_t \in \{1, \ldots, d\}$. In accordance with the theory behind RankBoost [37], k_t is chosen as

$$k_t = \operatorname*{arg\,min}_{k \in \{1,\ldots,d\}} \left\{ \min_{\alpha \in \mathbb{R}} \sum_{i=1}^{m}\sum_{j=1}^{n} D_t(\mathbf{x}_i^+, \mathbf{x}_j^-) \exp\Big(-\alpha(x_{ik}^+ - x_{jk}^-)\Big) \right\},$$

and α_t is then chosen as

$$\alpha_t = \operatorname*{arg\,min}_{\alpha \in \mathbb{R}} \sum_{i=1}^{m}\sum_{j=1}^{n} D_t(\mathbf{x}_i^+, \mathbf{x}_j^-) \exp\Big(-\alpha(f_t(\mathbf{x}_i^+) - f_t(\mathbf{x}_j^-))\Big) .$$

As in the case of the AdaBoost algorithm for classification [38], with the above choice of k_t and α_t, the bipartite RankBoost algorithm can be viewed as performing coordinate descent on an objective function that is a convex upper bound on the empirical ranking error (with respect to the training sample). Our final ranking function $f : \mathbb{R}^d \to \mathbb{R}$ is a linear function given by $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$, where $w_k = \sum_{\{t : k_t = k\}} \alpha_t$ for each $k \in \{1, \ldots, d\}$.

6.2.3 Data Sets

We conducted experiments on two publicly available microarray data sets. The first of these is a leukemia gene expression data set that was first used in [40] and was subsequently made available by the authors of that study.[1] The data set contains expression levels of 7129 genes across 72 samples. The samples in this data set correspond to tissue samples obtained from different leukemia patients. Of the 72 samples, 25 are from acute myeloid leukemia (AML) and 47 from acute lymphoblastic leukemia (ALL); in most studies involving this data set, the goal has been to classify samples as belonging to AML or ALL.

The second data set is a colon cancer gene expression data set that was first used in [8] and was subsequently made available.[2] The data set contains expression levels of 2000 genes across 62 samples. The samples in this data set correspond to tissue samples obtained from patients with and without colon cancer. Of the 62 samples, 40 are from tumor tissue and 22 from normal tissue; again, in most studies involving this data set, the goal has been to classify samples as tumor or normal.

We note that although the samples in both data sets come from two classes, our ranking method does not take this distinction into account.

6.2.4 Selection of Training Genes

Since the above data sets have not been used before in a way similar to our study, we did not have access to a pre-defined set of genes that could be used as training examples. We selected training genes based on existing biological knowledge.[3]

Of the 7129 genes in the leukemia data set, we selected 10 genes as positive training examples S_+ and 157 genes as negative training examples S_-. Each of the 10 genes selected as a positive example is a known classical marker[4] for either AML or ALL. Of the 157 genes selected as negative examples, 59 are internal controls available on the Affymetrix chip (indicated in the data set), and the other 98 are genes that are involved in a variety of physiological cellular functions (and therefore unrelated to cancer).

Of the 2000 genes in the colon cancer data set, we selected 10 genes as positive training examples S_+ and 56 genes as negative training examples S_-. Each of the 10 genes selected as a positive example is a known marker for colon cancer. The 56 genes selected as negative examples are again genes that are involved in a variety of physiological cellular functions.

[1] This data set is available from `http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi` .

[2] This data set is available from `http://microarray.princeton.edu/oncology/affydata/index.html` .

[3] The biological aspects of our study, including selection of training genes and validation of our results, were conducted in collaboration with Shiladitya Sengupta, an expert in cancer biology.

[4] A marker for a disease is a gene whose expression levels are distinctly altered in the disease. Markers are useful in diagnosis of diseases.

Table 6.1: Positive training genes for leukemia.

Markers for AML	Myeloperoxidase
	CD13
	CD33
	HOXA9 Homeo box A9
	V-myb avian myeloblastosis viral oncogene homolog-like
Markers for ALL	CD19
	CD10 (CALLA)
	TCL1 (T cell leukemia)
	C-myb
	Deoxyhypusine synthase

Table 6.2: Positive training genes for colon cancer.

Markers for colon cancer	Phospholipase A2
	Keratin 6 isoform
	Protein-tyrosine phosphatase PTP-H1
	Transcription factor IIIA
	Viral (v-raf) oncogene homolog 1
	Dual specificity mitogen-activated protein kinase kinase 1
	Transmembrane carcinoembryonic antigen
	Oncoprotein 18
	Phosphoenolpyruvate carboxykinase
	Extracellular signal-regulated kinase 1

The genes selected as positive training examples for leukemia are shown in Table 6.1; those selected as positive training examples for colon cancer are shown in Table 6.2. Examples of genes used as negative training examples are shown in Table 6.3.

6.2.5 Validation

In order to assess the quality of the ranking produced by our method, we performed an extensive validation with the biomedical literature to determine the biological relevance of the 25 top-ranked genes for each data set. Our main resource for this literature search was PubMed[5], an online indexing service for life sciences literature provided by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM). We also performed

[5]PubMed website: `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed`.

Table 6.3: Examples of negative training genes. These genes are involved in various physiological cellular functions and are unrelated to cancer.

House-keeping genes	GAPDH, Keratin, Paxillin, Beta actin, Adult heart mRNA for neutral calponin, Tropomyosin (Cytoskeletal), etc.
Ion channels	Inward rectifier K^+ channel protein (hirk1), Cholinergic receptor (nicotinic alpha), Potassium channel (Voltage-gated Kcnc1), CAB3b (calcium channel beta3 subunit), Delayed rectifier potassium channel, etc.
Essential enzymes	Mitochondrial cytochromes, ACAT, Alcohol dehydrogenase, Pyruvate dehydrogenase, H,K-ATPase, Cytochrome P450 reductase, Mitochondrial serine hydroxymethyltransferase, Glutathione peroxidase 1, Aldehyde reductase 1, Lysozyme, Biliverdin-IXalpha reductase, Type 3 iodothyronine deiodinase, Nitric oxide synthase 3 (endothelial cell), etc.
Cellular transport	Kidney water channel (hKID), Excitatory amino acid transporter 4, H,K-ATPase beta subunit, Gastric H,K-ATPase catalytic subunit, ATPases (Ca^{++} transporting in cardiac muscle / plasma membrane), Synaptotagmin i, ATP5B ATP synthase (H^+ transporting-mitochondrial F1 complex), GB DEF (Na^+ / myo-inositol cotransporter), Na^+ / glucose cotransporter 1, Renal Na^+-dependent phosphate cotransporter, Anion Exchanger 3 (Cardiac Isoform), etc.
Hormone-associated	Oxytocin, Thyroxine-binding globulin precursor, etc.
Receptors	Opioid receptor, Delta 1, Neurotrophic tyrosine kinase receptor type 3, CHRNA7 Cholinergic receptor nicotinic, high affinity immunoglobulin gamma fc receptor i 'alpha form' precursor, etc.
Focal cell-specific functions	High density lipoprotein binding protein (HBP), Gastric inhibitory polypeptide, MAG Myelin-associated glycoprotein, Low density lipoprotein receptor precursor, Fetus brain mRNA for vacuolar ATPase, Platelet-activating factor acetylhydrolase 2, ZAKI-4 mRNA in human skin fibroblast, Retina-specific amine oxidase, Adrenomedullin, Immunoglobulin Heavy Chain, Enhancer Element, Integrin Beta 1, Folate receptor beta precursor, Guanylyl cyclase (RetGC-2) mRNA, Dopamine D1A receptor gene, etc.

database searches using Entrez Gene[6], an online tool that allows one to search specifically for information related to a given gene, and BLAST[7] [9, 10], a sequence alignment tool that allows one to search for DNA sequences similar to a given sequence (and to thus find genes that are homologous to a given gene).

6.3 Results

Using the training genes described in Section 6.2.4, a ranking over the remaining genes in each data set was learned using the bipartite RankBoost algorithm (in each case, the algorithm was run for $T = 20$ rounds). Below we describe our results. Section 6.3.1 describes our results on the leukemia data set; Section 6.3.2 describes our results on the colon cancer data set.

6.3.1 Results on Leukemia Data

Below we give a list of the top 25 genes in the ranking learned for leukemia, together with discussions of their biological relevance to leukemia as could be determined from validation with the biomedical literature. A summary is provided in Table 6.4. Altogether, 21 of the 25 top-ranked genes have either a known relation to leukemia or, based on our validation, a potential relation to leukemia. Of these, 9 are known markers, 1 is a known therapeutic target, 2 are potential markers, and 9 are potential therapeutic targets. For comparison, Table 6.4 also shows the ranks of these genes in a ranking based on (absolute values of) the t-statistic computed from the AML/ALL distinction [97].[8]

1. *KIAA0220*
   KIAA0220 codes for a PI3-kinase related kinase SMG-1 like protein. Although the molecular function of the encoded protein is not yet known, its homology to PI3-kinase makes it an exciting pharmacological target. The dysregulation of the PI3-kinase signaling pathway has been implicated in multiple cancer types [68], and pharmacological agents targeting this pathway are currently in early clinical trials. Our ranking suggests that the protein encoded by KIAA0220 could possibly evolve as a similar target for the therapeutic management of leukemia.

2. *G-gamma globin*
   Higher levels of G-gamma globin have been reported in ALL [103]. Translation of both gamma and delta globin mRNAs is blocked by AZT, an anti-HIV drug which also inhibits the proliferation of leukemic cells [105].

[6]Entrez Gene website: `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=gene` .
[7]BLAST website: `http://www.ncbi.nlm.nih.gov/BLAST/` .
[8]For a fair comparison, the genes we used for training were removed before calculating the t-statistic ranks.

Table 6.4: Leukemia results: 25 top-ranked genes. ♦ Known marker; ◊ Potential marker; ■ Known therapeutic target; □ Potential therapeutic target; × No link found. (See text for detailed validation.) Ranks of the genes in a ranking based on (absolute values of) the t-statistic computed from the AML/ALL distinction [97] are shown for comparison.

	Gene	**Validation Summary**	**t-Statistic Rank**
1.	KIAA0220	□	6790
2.	G-gamma globin	♦	3681
3.	Delta-globin	♦	3768
4.	Brain-expressed HHCPA78 homolog (human, HL-60 acute promyelocytic leukemia cells)	□	6898
5.	Myeloperoxidase	♦	153
6.	Probable protein disulfide isomerase ER-60 precursor	□	6812
7.	NPM1 Nucleophosmin (nucleolar phosphoprotein B23, numatrin)	♦	430
8.	CD34	♦	6896
9.	Elongation factor-1-beta	×	4591
10.	CD24	♦	91
11.	60S ribosomal protein L23	□	2015
12.	5-aminolevulinic acid synthase	□	4890
13.	HLA class II histocompatibility antigen, DR alpha chain precursor	♦	5260
14.	Epstein-Barr virus small RNA-associated protein	□	6549
15.	HNRPA1 Heterogeneous nuclear ribonucleoprotein A1	□	4315
16.	Azurocidin	♦	176
17.	Red cell anion exchanger (EPB3, AE1, Band 3)	×	3968
18.	Topoisomerase (DNA) II beta (180kD)	■	21
19.	HLA class I histocompatibility antigen, F alpha chain precursor	×	284
20.	Probable G protein-coupled receptor LCR1 homolog	□	37
21.	HLA-SB alpha gene (class II antigen)	×	6535
22.	Int-6	◊	3994
23.	Alpha-tubulin	□	5654
24.	Terminal transferase	♦	7
25.	Glycophorin B precursor	◊	3128

3. *Delta-globin*
   See 2 above.

4. *Brain-expressed HHCPA78 homolog (human, HL-60 acute promyelocytic leukemia cells)*
   Although the brain-expressed HHCPA78 homolog has not been implicated in AML/ALL, it has been identified in leukemia cells. It is expected to be a homolog of thioredoxin interacting protein (Entrez Gene), which could possibly be involved in the conversion of post-mitotic cells to differentiating ones.

5. *Myeloperoxidase*
   Myeloperoxidase is an established marker for AML.[9]

6. *Probable protein disulfide isomerase ER-60 precursor*
   A set of chaperone proteins that included protein disulfide isomerase were identified as interesting targets in a global profiling of the cell surface proteome of cancer cells [94]. Furthermore, in a clinical study of leukemia patients, levels of protein disulfide isomerase were shown to be distinctly altered, and correlated with resistance to chemotherapy [104].

7. *NPM1 Nucleophosmin (nucleolar phosphoprotein B23, numatrin)*
   Nucleophosmin (NPM), a nucleocytoplasmic shuttling protein with prominent nucleolar localization, regulates the ARF-p53 tumor-suppressor pathway. NPM is a characteristic feature of a large subgroup of patients with AML [36].

8. *CD34*
   CD34 is over-expressed in AML and may be valuable in detecting minimal residual disease [92]. In a meta-analysis of 2483 patients, it was shown to be associated with a poor remission rate [53]. It has also been used to target drugs to leukemic cells [19].

9. *Elongation factor-1-beta*
   No link found.

10. *CD24*
    CD24 is expressed on a majority of B-lineage ALL [64], and on CD31+/CD33+ myeloid cells in the bone marrow of children with AML[10]. Furthermore, among ALL patients, a low CD24/CD45 antigen density ratio has been associated with a good prognosis [65].

[9] Myeloperoxidase was actually included in our set of positive training genes. Its appearance in the ranked list is an artefact of the data set; in some instances, the data set has multiple occurrences of the same gene. Although we removed the specific occurrences of genes that we used in training and output a ranking only over the remaining genes, training genes with multiple occurrences can appear in the ranked list due to this artefact.

[10] Source: `http://www.cancer.gov/cancerinfo/pdq/treatment/childAML/healthprofessional` .

11. *60S ribosomal protein L23*

    Ribosomal proteins have been implicated by multiple studies to be linked with different types of cancer [11]. Although the direct relationship between 60S ribosomal protein L23 and AML/ALL has not yet been established, it was reported that pokeweed antiviral protein (PAP) and ricin A chains, which inactivate 60S subunits, could prevent the growth of leukemia cells in mice [87], indicating that this ribosomal protein could emerge as a key therapeutic target in the management of leukemia.

12. *5-aminolevulinic acid synthase*

    5-Aminolevulinic acid synthase (ALAS) is the first enzyme of the heme biosynthesis pathway [96]. Although ALAS has not been directly implicated in AML/ALL, heme enhances globin gene transcription and is essential for globin translation (see 2 and 3 above). Furthermore, heme also seems to play a role in regulating either synthesis or stability of hemoproteins, many of which have been implicated in tumorigenesis [84].

13. *HLA class II histocompatibility antigen, DR alpha chain precursor*

    HLA-DR is a positive immunophenotyping marker in most AML cells. In a study carried out to investigate the clinical significance of surface antigens in AML, the expression of HLA-DR was reported to be associated with a lower remission rate [41].

14. *Epstein-Barr virus small RNA-associated protein*

    This gene encodes a cytoplasmic ribosomal protein that is a component of the 60S subunit (see 11 above), and belongs to the L22E family of ribosomal proteins. Although this ribosomal protein has not been implicated in the context of ALL/AML, one of the pseudogenes of this gene is fused to the acute myeloid leukemia 1 (AML1) gene (Entrez Gene).

15. *HNRPA1 Heterogeneous nuclear ribonucleoprotein A1*

    The expression of heterogeneous nuclear ribonucleoparticule A1 and A2 proteins is elevated in a variety of human cancers, and is lower or absent in normal tissues. Interestingly, the knock-down of the ribonucleoprotein with RNA-interference was shown to induce apoptosis (cell death) in a variety of tumors, suggesting that these could be developed as interesting therapeutic targets [79].

16. *Azurocidin*

    Azurocidin is known to be a marker for AML [25, Chapter 15].

17. *Red cell anion exchanger (EPB3, AE1, Band 3)*

    No link found.

18. *Topoisomerase (DNA) II beta (180kD)*
    Topoisomerase inhibitors, including teniposide and etoposide, are currently being used to treat certain forms of leukemia [44].

19. *HLA class I histocompatibility antigen, F alpha chain precursor*
    No link found.

20. *Probable G protein-coupled receptor LCR1 homolog*
    This G protein-coupled receptor is related to LCR1/ chemokine receptor-4 (CXCR-4), which is over-expressed in bone-marrow derived blasts, and is implicated in leukemic marrow infiltration [76]. Although this homolog has not been studied in leukemia, the culture of AML cells with the CXCR-4 ligand, SDF-1, promoted their survival, whereas addition of neutralizing CXCR4 antibodies or SDF-1 antibodies significantly decreased it [99], suggesting that the probable G protein-coupled receptor LCR1 could be an interesting target for therapeutics.

21. *HLA-SB alpha gene (class II antigen)*
    No link found.

22. *Int-6*
    Int-6/eIF3-p48 has been identified as a human protein that binds to the human T-cell leukemia virus type I Tax oncoprotein. Although the role of Int-6/eIF3-p48 in human carcinogenesis is unknown at the present time, its expression is down-regulated in two of the most common forms of cancer in humans, namely breast and lung tumors [70].

23. *Alpha-tubulin*
    Tubulin, the protein component of microtubules (cytoskeletal elements that are important for mitotic spindle assembly and cell division), is a key molecular target for cancer therapy. Interestingly, alpha tubulin is phosphorylated in leukemic cells, in contrast to normal lymphocytes where it exists in a non-phosphorylated state, suggesting that it might play a role in progression of leukemia [71].

24. *Terminal transferase*
    Terminal transferase (TdT) is an established marker for ALL and is expressed in over 95% of ALL cases [25, Chapter 15].

25. *Glycophorin B precursor*
    The role of glycophorin in tumor malignancies is not yet well-understood. However, the over-expression of the proto-oncogene c-myc, which is implicated in leukemia and other cancers, has been shown to repress the expression of glycophorin [95].

6.3.2 Results on Colon Cancer Data

Below we give a list of the top 25 genes in the ranking learned for colon cancer, together with discussions of their biological relevance to colon cancer as could be determined from validation with the biomedical literature. A summary is provided in Table 6.5. Altogether, 19 of the 25 top-ranked genes have either a known relation to colon cancer or, based on our validation, a potential relation to colon cancer. Of these, 3 are known markers, 6 are potential markers, and 11 are potential therapeutic targets (one is both a potential marker and a potential therapeutic target). For comparison, Table 6.5 also shows the ranks of these genes in a ranking based on (absolute values of) the t-statistic computed from the tumor/normal distinction [97].

1. *26-kDa cell surface protein TAPA-1*
   TAPA-1/DC1/CD81 has been implicated in the migration of endothelial cells, a key step in angiogenesis (growth of new blood vessels; increases tumor growth) and carcinoma (metastasis) [16].

2. *Id1*
   Id helix-loop-helix (HLH) proteins function as regulators of cell growth and differentiation and, when over-expressed, can induce malignant transformation from normal to cancer cells. The expression of Id proteins in adenocarcinoma has been shown to be at least in part a consequence of loss of p53 function (p53 is a tumor-suppressing gene), and contributes to the uncontrolled proliferation of tumor cells [106]. Id1 has also been implicated in tumor angiogenesis [69]. Interestingly, Id genes are normally expressed at very low levels in adults, making them attractive new targets for anti-cancer drug design.

3. *Cleavage and polyadenylation specificity factor*
   No link found.

4. *Interferon-inducible protein 9-27*
   This 17-KDa membrane protein plays a key role in mediating the anti-proliferative effects of interferons, which have proven clinically effective as anti-tumor agents in a subset of cancer types [29]. Furthermore, the silencing of this protein is also implicated in immortalisation [60], a key step towards tumorigenesis. Also see 18 below.

5. *Nonspecific crossreacting antigen*
   Nonspecific crossreacting antigen (NCA) is a major component of carcinoembryonic antigen (CEA), which is an important tumor marker. It has been shown using northern blot hybridization that NCA is expressed predominantly in cancerous tissues, making it a useful marker for colon cancer [90].

Table 6.5: Colon cancer results: 25 top-ranked genes. ♦ Known marker; ◊ Potential marker; ■ Known therapeutic target; □ Potential therapeutic target; × No link found. (See text for detailed validation.) Ranks of the genes in a ranking based on (absolute values of) the t-statistic computed from the tumor/normal distinction [97] are shown for comparison.

	Gene	**Validation Summary**	**t-Statistic Rank**
1.	26-kDa cell surface protein TAPA-1	□	885
2.	Id1	□	1405
3.	Cleavage and polyadenylation specificity factor	×	304
4.	Interferon-inducible protein 9-27	◊	92
5.	Nonspecific crossreacting antigen	♦	214
6.	cAMP response element regulatory protein (CREB2)	□	707
7.	MHC class I HLA-Bw58	×	1932
8.	Translational initiation factor 2 gamma subunit	×	104
9.	Splicing factor (CC1.4)	□	482
10.	Nucleolar protein (B23)	◊	7
11.	Lactate dehydrogenase-A (LDH-A)	◊	466
12.	Guanine nucleotide-binding protein G(OLF), alpha subunit	□	730
13.	LI-cadherin	◊	1482
14.	Lysozyme	♦	131
15.	Prolyl 4-hydroxylase beta-subunit and disulfide isomerase (P4HB)	◊	373
16.	Eukaryotic initiation factor 4AII	□	1197
17.	HLA class I histocompatibility antigen, A-26(A-10) A*2601 alpha	×	962
18.	Interferon-inducible protein 1-8D	□	322
19.	Very long chain acyl-CoA dehydrogenase	×	1760
20.	Dipeptidase	♦	744
21.	Heat shock 27 KD protein	□	667
22.	Tyrosine-protein kinase receptor TIE-1 precursor	◊□	618
23.	Mitochondrial import receptor MOM38	×	1994
24.	Mitochondrial matrix protein P1 precursor	□	3
25.	Eukaryotic initiation factor EIF-4A homolog	□	401

6. *cAMP response element regulatory protein (CREB2)*
   Also known as activating transcription factor 2 (ATF-2), CREB2 binds to cAMP response element (CRE) either as a homo-dimer, or as a hetero-dimer in conjunction with activator proteins (AP1), such as Jun, fos and ATF/CREB families, which regulate transcription in response to extracellular signals and have a decisive role in cell proliferation, tumorigenesis and apoptosis [50]. ATF-2 mRNA is implicated in several types of human cancers, such as gastric, colon, pancreatic, and esophageal cancers [98]. It has also been implicated in driving the expression of vascular endothelial growth factor (VEGF) under endoplasmic reticulum stress, which could further promote tumor angiogenesis.

7. *MHC class I HLA-Bw58*
   No link found.

8. *Translational initiation factor 2 gamma subunit*
   No link found.

9. *Splicing factor (CC1.4)*
   The protein encoded by this gene is an RNA binding protein, which is found in the nucleus and co-localizes with the core spliceosomal protein. Studies of a murine protein with high sequence similarity to this protein suggest that this protein may act as a transcriptional co-activator for JUN/AP1, which have been shown to play a dominant role in the oncogenic ras-induced transformation of human carcinoma cells [107].

10. *Nucleolar protein (B23)*
    Nucleophosmin (B23) is involved in ribosome biogenesis, and interacts with tumor suppressor proteins p53 and Rb [61]. Levels of nucleophosmin have been reported to be up-regulated in many tumor types [77].

11. *Lactate dehydrogenase-A (LDH-A)*
    Lactate dehydrogenase (LDH) levels have been correlated with poor prognosis and with resistance to chemotherapy and radiotherapy in various cancers. LDH is over-expressed in colorectal cancer [78], and has also been implicated in mediating c-myc-induced transformation from normal to cancer cells [93].

12. *Guanine nucleotide-binding protein G(OLF), alpha subunit*
    Persistent activation of the G-protein (olf) has been shown to exert convergent signals through the rho kinase pathway to promote cellular invasion and survival in solid tumors during towards metastasis [88].

13. *LI-cadherin*

    Over-expression of LI-cadherin has been implicated in lymph node metastasis of gastrointestinal cancer, which is closely related to colon cancer [58].

14. *Lysozyme*

    Colonic epithelium can produce lysozyme, and its expression is up-regulated in the dysplastic epithelium in adenomas and in invasive cancer cells [108].

15. *Prolyl 4-hydroxylase beta-subunit and disulfide isomerase (P4HB)*

    This protein possesses two different enzymatic functions depending on whether it is present in cells in monomer form (disulfide isomerase) or in the prolyl 4-hydroxylase tetramer form [80]. Interestingly, the expression of prolyl-hydroxylase was shown to suppress hypoxia inducible factor-1-alpha activation and inhibit angiogenesis and growth of colon carcinoma [34].

16. *Eukaryotic initiation factor 4AII*

    Eukaryotic translation initiation factor, eIF4A, exists as a complex with cyclin-dependent kinases (CDKs). The CDK-eIF4A complex is abundant in actively proliferating and growing cells, but is absent from cells that have ceased dividing, indicating that this interaction could underlie the molecular mechanism linking cell proliferation with translational control, which is altered in cancer progression. Interestingly, the CDK-eIF4A complex contains kinase activity that is sensitive to the CDK-specific inhibitor roscovitine, suggesting that this may be a lead compound for the treatment of colon cancer [51].

17. *HLA class I histocompatibility antigen, A-26(A-10) A*2601 alpha*

18. *Interferon-inducible protein 1-8D*

    Interestingly, both interferon-inducible protein 1-8D and 9-27 (see 4 above) have been postulated to mediate the link between interferon- and radiation-induced cell death [20]. Levels of the former are also up-regulated in tumor cells following the suppression of bcr-abl synthesis by siRNAs or tyrosine kinase activity by Glivec, a novel anti-cancer drug [109]. These studies indicate that this protein could be an interesting therapeutic target for inducing tumor cell death.

19. *Very long chain acyl-CoA dehydrogenase*

    No link found.

20. *Dipeptidase*

    Dipeptidase 1 has been used as a marker for colon cancer [74].

21. *Heat shock 27 KD protein*
    Low molecular weight stress proteins such as heat shock protein 27 (hsp27) have been implicated in cellular processes potentially related to malignant transformation from normal to cancer cells [75]. Furthermore, increased expression of hsp27 has been shown to enhance the tumorigenicity of immunogenic colon carcinoma [39].

22. *Tyrosine-protein kinase receptor TIE-1 precursor*
    Protein tyrosine kinases (PTKs) are a major class of proto-oncogenes that are involved in tumor progression and angiogenesis. Positive immunohistochemical staining for tie-1 was observed in gastric adenocarcinoma tissues [67]. Furthermore, clinico-pathological studies have indicated that tie-1 kinase expression is inversely correlated with patients' survival, indicating that tie-1 inhibitors could have major implications in colon cancer.

23. *Mitochondrial import receptor MOM38*
    No link found.

24. *Mitochondrial matrix protein P1 precursor*
    Also known as heat shock protein 60 (hsp60), it belongs to a group of proteins that typically modulate the cellular response to stress but are also implicated in the cell cycle, cell proliferation and differentiation. Altered expression of HSP has been reported for nearly all classes of tumors, and hsp60 specifically has been shown to be over-expressed in colon cancer [52].

25. *Eukaryotic initiation factor EIF-4A homolog*
    See 16 above.

6.4 Discussion

We have proposed a novel approach for the problem of ranking genes by their relevance to a given disease based on gene expression data. Our method is based on formulating the problem as a bipartite ranking problem, and exploits existing biological knowledge, in the form of training examples, to directly address the problem of ranking relevant genes higher than irrelevant ones.

The benefit of our approach over previous methods is clear from our results. For example, the KIAA0220 gene, which is ranked 1st in our list of genes relevant to leukemia and which appears to be a very promising lead as a potential therapeutic target, is placed close to the bottom – at position 6628 – in a ranking based on the t-statistic (see Table 6.4). A similar observation holds for several of our other top-ranking genes. Such genes are unlikely to be pulled

out by traditional methods; yet, based on a few training genes that exemplify the notion of biological relevance to leukemia, genes such as KIAA0220 are easily discovered by our method.

We note again that although we have compared our approach to the t-statistic based ranking method, in many cases, such ranking methods may not even be applicable. In particular, one may want to include biological samples corresponding to various kinds of experiments pertaining to the disease under study; if the samples cannot be divided into two natural classes, traditional methods such as that based on the t-statistic cannot be applied.

Perhaps the most exciting aspect of our results is that, using only markers as training genes, several potential therapeutic targets are discovered by the learned ranking. While the question of whether it is actually possible to develop drugs based on these potential targets can be answered only through biological and chemical experiments, the possibility of using machine learning methods to point to such targets suggests an important role for machine learning in post-genome medical research.

References

[1] S. Agarwal, T. Graepel, R. Herbrich, S. Har-Peled, and D. Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, 2005.

[2] S. Agarwal, T. Graepel, R. Herbrich, and D. Roth. A large deviation bound for the area under the ROC curve. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[3] S. Agarwal, S. Har-Peled, and D. Roth. A uniform convergence bound for the area under the ROC curve. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[4] S. Agarwal and P. Niyogi. Stability and generalization of bipartite ranking algorithms. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

[5] S. Agarwal and D. Roth. Learnability of bipartite ranking functions. In *Proceedings of the 18th Annual Conference on Learning Theory*, 2005.

[6] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[7] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997.

[8] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the USA*, 96:6745–6750, 1999.

[9] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[10] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[11] A. Amsterdam, K. C. Sadler, K. Lai, S. Farrington, R. T. Bronson, J. A. Lees, and N. Hopkins. Many ribosomal protein genes are cancer genes in zebrafish. *PloS Biology*, 2:690, 2004.

[12] M. Anthony and P. Bartlett. *Learning in Neural Networks: Theoretical Foundations*. Cambridge University Press, 1999.

[13] P. L. Bartlett, P. M. Long, and R. C. Williamson. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452, 1996.

[14] Z. W. Birnbaum and O. M. Klose. Bounds for the variance of the Mann-Whitney statistic. *Annals of Mathematical Statistics*, 38, 1957.

[15] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[16] C. Boucheix, G. H. Duc, C. Jasmin, and E. Rubinstein. Tetraspanins and malignancy. *Expert Reviews in Molecular Medicine*, pages 1–17, 2001.

[17] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[18] R. C. Buck. Partition of space. *American Mathematical Monthly*, 50:2541–544, 1943.

[19] C. Carrion, M. A. de Madariaga, and J. C. Domingo. In vitro cytotoxic study of immunoliposomal doxorubicin targeted to human CD34+ leukemic cells. *Life Sciences*, 75(3):313–328, 2004.

[20] E. Clave, E. D. Carosella, E. Gluckman, and G. Socie. Radiation-enhanced expression of interferon-inducible genes in the kg1a primitive hematopoietic cell line. *Leukemia*, 11(3):114–119, 1997.

[21] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[23] C. Cortes and M. Mohri. AUC optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[24] C. Cortes and M. Mohri. Confidence intervals for the area under the ROC curve. In *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[25] R. S. Cotran, V. Kumar, and S. L. Robbins. *Robbins Pathologic Basis of Disease*. W. B. Saunders Company, 4th edition, 1989.

[26] K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems 14*, pages 641–647. MIT Press, 2002.

[27] D. Dantzig. On the consistency and power of Wilcoxon's two sample test. In *Koninklijke Nederlandse Akademie van Weterschappen, Series A*, volume 54, 1915.

[28] V. H. de la Peña and E. Giné. *Decoupling: From Dependence to Independence*. Springer-Verlag, New York, 1999.

[29] G. A. Deblandre, O. P. Marinx, S. S. Evans, S. Majjaj, O. Leo, D. Caput, G. A. Huez, and M. G. Wathelet. Expression cloning of an interferon-inducible 17-kDa membrane protein implicated in the control of cell growth. *Journal of Biological Chemistry*, 270(40):23860–23866, 1995.

[30] L. Devroye. Exponential inequalities in nonparametric estimation. In G. Roussas, editor, *Nonparametric Functional Estimation and Related Topics*, NATO ASI Series, pages 31–44. Kluwer Academic Publishers, 1991.

[31] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition.* Springer-Verlag, New York, 1996.

[32] L. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979.

[33] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman and Hall, 1993.

[34] N. Erez, M. Milyavsky, R. Eilam, I. Shats, N. Goldfinger, and V. Rotter. Expression of prolyl-hydroxylase-1 (PHD1/EGLN2) suppresses hypoxia inducible factor-1 alpha activation and inhibits tumor growth. *Cancer Research*, 63(24):8777–8783, 2003.

[35] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

[36] B. Falini, C. Mecucci, E. Tiacci, M. Alcalay, R. Rosati, L. Pasqualucci, R. L. Starza, D. Diverio, E. Colombo, A. Santucci, B. Bigerna, R. Pacini, A. Pucciarini, A. Liso, M. Vignetti, P. Fazi, N. Meani, V. Pettirossi, G. Saglio, F. M. F. Lo-Coco, P. G. Pelicci, and M. F. Martelli. Cytoplasmic nucleophosmin in acute myelogenous leukemia with a normal karyotype. *New England Journal of Medicine*, 352(3):254–266, 2005.

[37] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[38] Y. Freund and R. 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, 1997.

[39] C. Garrido, A. Fromentin, B. Bonnotte, N. Favre, M. Moutet, A. P. Arrigo, P. Mehlen, and E. Solary. Heat shock protein 27 enhances the tumorigenicity of immunogenic rat colon carcinoma cell clones. *Cancer Research*, 58(23):5495–5499, 1998.

[40] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[41] J. D. Griffin, R. Davis, D. A. Nelson, F. R. Davey, R. J. Mayer, C. Schiffer, O. R. McIntyre, and C. D. Bloomfield. Use of surface marker analysis to predict outcome of adult acute myeloblastic leukemia. *Blood*, 68:1232–1241, 1986.

[42] I. Guyon, J. Weston, S. Barnhill, and V. N. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

[43] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.

[44] J. G. Hardman, L. E. Limbird, and A. G. Gilman. *Goodman & Gilman's The Pharmacological Basis of Therapeutics*. McGraw-Hill Professional, 10th edition, 2001.

[45] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100:78–150, 1992.

[46] D. Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz, 1999.

[47] R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Learning preference relations for information retrieval. In *Proceedings of the ICML-1998 Workshop on Text Categorization and Machine Learning*, 1998.

[48] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. *Advances in Large Margin Classifiers*, pages 115–132, 2000.

[49] S. I. Hill, H. Zaragoza, R. Herbrich, and P. J. W. Rayner. Average precision and the problem of generalisation. In *Proceedings of the ACM SIGIR Workshop on Mathematical and Formal Methods in Information Retrieval*, 2002.

[50] S. Huguier, J. Baguet, S. Perez, H. Dam, and M. Castellazzi. Transcription factor ATF-2 cooperates with v-Jun to promote growth factor independent proliferation in vitro and tumor formation in vivo. *Molecular and Cellular Biology*, 18:7020–7029, 1998.

[51] A. P. Hutchins, G. R. Roberts, C. W. Lloyd, and J. H. Doonan. In vivo interaction between CDKA and eIF4A: a possible mechanism linking translation and cell proliferation. *FEBS Letters*, 556(1-3):91–94, 2004.

[52] C. Jolly and R. I. Morimoto. Role of the heat shock response and molecular chaperones in oncogenesis and cell death. *Journal of the National Cancer Institute*, 92:1564–1572, 2000.

[53] Y. Kanda, T. Hamaki, R. Yamamoto, A. Chizuka, M. Suguro, T. Matsuyama, N. Takezako, A. Miwa, M. Kami, H. Hirai, and A. Togawa. Clinical significance of CD34 expression in response to therapy of patients with acute myeloid leukemia: an overview of 2483 patients from 22 studies. *Cancer*, 88(11):2529–2533, 2000.

[54] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[55] M. Kearns and D. Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11:1427–1453, 1999.

[56] M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.

[57] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

[58] S. Ko, K.-M. Chu, J. M. Luk, B. W. Wong, S.-T. Yuen, S.-Y. Leung, and J. Wong. Overexpression of LI-cadherin in gastric cancer is associated with lymph node metastasis. *Biochemical and Biophysical Research Communications*, 319(2):562–568, 2004.

[59] B. Krishnapuram, L. Carin, and A. J. Hartemink. Joint classifier and feature optimization for cancer diagnosis using gene expression data. In *Proceedings of the 7th Annual Conference on Research in Computational Molecular Biology*, pages 167–175, 2003.

[60] O. I. Kulaeva, S. Draghici, L. Tang, J. M. Kraniak, S. J. Land, and M. A. Tainsky. Epigenetic silencing of multiple interferon pathway genes after cellular immortalization. *Oncogene*, 22(26):4118–4127, 2003.

[61] S. Kurki, K. Peltonen, L. Latonen, T. M. Kiviharju, P. M. Ojala, D. Meek, and M. Laiho. Nucleolar protein NPM interacts with HDM2 and protects tumor suppressor protein p53 from HDM2-mediated degradation. *Cancer Cell*, 5(5):465–475, 2004.

[62] S. Kutin and P. Niyogi. The interaction of stability and weakness in AdaBoost. Technical Report TR-2001-30, Computer Science Department, University of Chicago, 2001.

[63] S. Kutin and P. Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, 2002.

[64] L. L. Lanier, J. P. Allison, and J. H. Phillips. Correlation of cell surface antigen expression on human thymocytes by multi-color flow cytometric analysis: implications for differentiation. *The Journal of Immunology*, 137:2501–2507, 1986.

[65] T. Lavabre-Bertrand, C. Duperray, C. Brunet, P. Poncelet, C. Exbrayat, P. Bourquard, C. Lavabre-Bertrand, J. Brochier, M. Navarro, and G. Janossy. Quantification of CD24 and CD45 antigens in parallel allows a precise determination of B-cell maturation stages: relevance for the study of B-cell neoplasias. *Leukemia*, 8(3):402–208, 1994.

[66] E. L. Lehmann. *Nonparametrics: Statistical Methods Based on Ranks*. Holden-Day, San Francisco, California, 1975.

[67] W.-C. Lin, A. F.-Y. Li, C.-W. Chi, W.-W. Chung, C. L. Huang, W.-Y. Lui, H.-J. Kung, and C.-W. Wu. tie-1 protein tyrosine kinase: A novel independent prognostic marker for gastric cancer. *Clinical Cancer Research*, 5(7):1745–1751, 1999.

[68] J. Luo, B. D. Manning, and L. C. Cantley. Targeting the PI3K-Akt pathway in human cancer: Rationale and promise. *Cancer Cell*, 4(4):257–262, 2003.

[69] D. Lyden, A. Z. Young, D. Zagzag, W. Yan, W. Gerald, R. O'Reilly, B. L. Bader, R. O. Hynes, Y. Zhuang, K. Manova, and R. Benezra. Id1 and Id3 are required for neurogenesis, angiogenesis and vascularization of tumour xenografts. *Nature*, 401(6754):670–677, 1999.

[70] A. Marchetti, F. Buttitta, S. Pellegrini, G. Bertacca, and R. Callahan. Reduced expression of INT-6/eIF3 -p48 in human tumors. *International Journal of Oncology*, 18(1):175–179, 2001.

[71] A. Marie-Cardine, H. Kirchgessner, C. Eckerskorn, S. C. Meuer, and B. Schraven. Human T lymphocyte activation induces tyrosine phosphorylation of alpha-tubulin and its association with the SH2 domain of the p59fyn protein tyrosine kinase. *European Journal of Immunology*, 25(12):3290–3297, 1995.

[72] J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, New York, 2002.

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

[74] C. M. McIver, J. M. Lloyd, P. J. Hewett, and J. E. Hardingham. Dipeptidase 1: a candidate tumor-specific molecular marker in colorectal carcinoma. *Cancer Letters*, 209(1):67–74, 2004.

[75] A. Michils, M. Redivo, V. Z. de Beyl, V. de Maertelaer, D. Jacobovitz, P. Rocmans, and J. Duchateau. Increased expression of high but not low molecular weight heat shock proteins in resectable lung carcinoma. *Lung Cancer*, 33(1):59–67, 2001.

[76] R. Mohle, M. Schittenhelm, C. Failenschmid, F. Bautz, K. Kratz-Albers, H. Serve, W. Brugger, and L. Kanz. Functional response of leukaemic blasts to stromal cell-derived factor-1 correlates with preferential expression of the chemokine receptor CXCR4 in acute myelomonocytic and lymphoblastic leukaemia. *British Journal of Haematology*, 110(3):563–572, 2000.

[77] Y. Nozawa, N. V. Belzen, A. C. J. Van Der Made, W. N. M. Dinjens, and F. T. Bosman. Expression of nucleophosmin/B23 in normal and neoplastic colorectal mucosa. *Journal of Pathology*, 178(1):48–52, 1996.

[78] S. Ono. Studies on carcinoembryonic antigen (CEA), lactate dehydrogenase (LDH), and LDH isozymes in the tissue of colorectal carcinoma. *Nippon Geka Gakkai Zasshi*, 84(4):336–348, 1983.

[79] C. Patry, L. Bouchard, P. Labrecque, D. Gendron, B. Lemieux, J. Toutant, E. Lapointe, R. Wellinger, and B. Chabot. Small interfering RNA-mediated reduction in heterogeneous nuclear ribonucleoparticule A1/A2 proteins induces apoptosis in human cancer cells but not in normal mortal cell lines. *Cancer Research*, 63(22):7679–7688, 2003.

[80] T. Pihlajaniemi, T. Helaakoski, K. Tasanen, R. Myllyla, M. L. Huhtala, J. Koivu, and K. I. Kivirikko. Molecular cloning of the beta-subunit of human prolyl 4-hydroxylase. this subunit and protein disulphide isomerase are products of the same gene. *EMBO Journal*, 6(3):643–649, 1987.

[81] L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35(4):965–984, 1988.

[82] T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi. General conditions for predictivity in learning theory. *Nature*, 428:419–422, 2004.

[83] D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.

[84] P. Ponka. Cell biology of heme. *American Journal of the Medical Sciences*, 318(4):241, 1999.

[85] S. Rajaram, A. Garg, X. S. Zhou, and T. S. Huang. Classification approach towards ranking and sorting problems. In *Proceedings of the European Conference on Machine Learning*, pages 301–312, 2003.

[86] A. Rakotomamonjy. SVMs and area under ROC curves. Technical report, PSI- INSA de Rouen, 2004.

[87] S. Ramakrishnan and L. L. Houston. Prevention of growth of leukemia cells in mice by monoclonal antibodies directed against Thy 1.1 antigen disulfide linked to two ribosomal inhibitors: pokeweed antiviral protein or ricin A chain. *Cancer Research*, 44(4):1398–1404, 1984.

[88] K. Régnauld, Q.-D. Nguyen, L. Vakaet, E. Bruyneel, J.-M. Launay, T. Endo, M. Mareel, C. Gespach, and S. Emami. G-protein alpha(olf) subunit promotes cellular invasion, survival, and neuroendocrine differentiation in digestive and urogenital epithelial cells. *Oncogene*, 21(25):4020–4031, 2002.

[89] S. Rosset. Model selection via the AUC. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[90] C. Sato, M. Miyaki, S. Oikawa, H. Nakazato, and G. Kosaki. Differential expression of carcinoembryonic antigen and nonspecific crossreacting antigen genes in human colon adenocarcinomas and normal colon mucosa. *Japanese Journal of Cancer Research*, 79:433–437, 1988.

[91] R. E. Schapire. The boosting approach to machine learning: An overview. In *MSRI Workshop on Nonlinear Estimation and Classification*, 2002.

[92] M. P. Scolnik, R. Morilla, M. M. de E. de Bracco, D. Catovsky, and E. Matutes. CD34 and CD117 are overexpressed in AML and may be valuable to detect minimal residual disease. *Leukemia Research*, 26(7):615–619, 2002.

[93] H. Shim, C. Dolde, B. C. Lewis, C.-S. Wu, G. Dang, R. A. Jungmann, R. Dalla-Favera, and C. V. Dang. c-Myc transactivation of LDH-A: Implications for tumor metabolism and growth. *Proceedings of the National Academy of Sciences of the USA*, 94:6658–6663, 1997.

[94] B. K. Shin, H. Wang, A. M. Yim, F. L. Naour, F. Brichory, J. H. Jang, R. Zhao, E. Puravs, J. Tra, C. W. Michael, D. E. Misek, and S. M. Hanash. Global profiling of the cell surface proteome of cancer cells uncovers an abundance of proteins with chaperone function. *Journal of Biological Chemistry*, 278(9):7607–7616, 2003.

[95] W. Shoji, Y. Ohmori, and M. Obinata. Sialoglycoprotein c-Myc selectively regulates the latent period and erythroid-specific genes in murine erythroleukemia cell differentiation. *Japanese Journal of Cancer Research*, 84(8):885–892, 1993.

[96] P. M. Shoolingin-Jordan, S. Al-Daihan, D. Alexeev, R. L. Baxter, S. S. Bottomley, I. D. Kahari, I. Roy, M. Sarwar, L. Sawyer, and S. F. Wang. 5-Aminolevulinic acid synthase: mechanism, mutations and medicine. *Biochimica et Biophysica Acta*, 1647(1-2):361–366, 2003.

[97] Y. Su, T. M. Murali, V. Pavlovic, M. Schaffer, and S. Kasif. RankGene: Identification of diagnostic genes based on expression data. *Bioinformatics*, 19(12):1578–1579, 2003.

[98] J. Takeda, T. Maekawa, T. Sudo, Y. Seino, H. Imura, N. S. N, C. Tanaka, and S. Ishii. Expression of CREBP1 transcriptional regulator binding to the cyclic AMP response element in central nervous system, regenerating liver, and human tumors. *Oncogene*, 6:1009–1014, 1991.

[99] S. Tavor, I. Petit, S. Porozov, A. Avigdor, A. Dar, L. Leider-Trejo, N. Shemtov, V. Deutsch, E. Naparstek, A. Nagler, and T. Lapidot. CXCR4 regulates migration and development of human acute myelogenous leukemia stem cells in transplanted NOD/SCID mice. *Cancer Research*, 64:2817–2824, 2004.

[100] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, pages 1134–1142, 1984.

[101] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

[102] V. N. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

[103] A. R. Villalobos-Arambula, J. C. Aguilar-Luna, A. Esparza, F. J. Perea, R. de Loza, A. Hernandez-Cordova, and B. Ibarra. Fetal hemoglobin and the gamma G/gamma A chain ratio in children with acute lymphoblastic leukemia L1 and L2. *Sangre (Barc)*, 38(1):31–35, 1993.

[104] T. Voss, H. Ahorn, P. Haberl, H. Dohner, and K. Wilgenbus. Correlation of clinical data with proteomics profiles in 24 patients with B-cell chronic lymphocytic leukemia. *International Journal of Cancer*, 91(2):180–186, 2001.

[105] D. A. Weidner and J. P. Sommadossi. 3'-Azido-3'-deoxythymidine inhibits globin gene transcription in butyric acid-induced K-562 human leukemia cells. *Molecular Pharmacology*, 38:797–804, 1990.

[106] J. W. Wilson, R. W. Deed, T. Inoue, M. Balzi, A. Becciolini, P. Faraoni, C. S. Potten, and J. D. Norton. Expression of Id helix-loop- helix proteins in colorectal adenocarcinoma correlates with p53 expression and mitotic index. *Cancer Research*, 61:8803–10, 2001.

[107] L. Xiao and W. Lang. A dominant role for the c-Jun NH2-terminal kinase in oncogenic ras-induced morphologic transformation of human lung carcinoma cells. *Cancer Research*, 60:400–408, 2000.

[108] S. T. Yuen, M. P. Wong, L. P. Chung, S. Y. Chan, N. Cheung, J. Ho, and S. Y. Leung. Up-regulation of lysozyme production in colonic adenomas and adenocarcinomas. *Histopathology*, 32(2):126–132, 1998.

[109] Z. Zhelev, R. Bakalova, H. Ohba, A. Ewis, M. Ishikawa, Y. Shinohara, and Y. Baba. Suppression of bcr-abl synthesis by siRNAs or tyrosine kinase activity by Glivec alters different oncogenes, apoptotic/antiapoptotic genes and cell proliferation factors (microarray study). *FEBS Letters*, 570(1-3):195–204, 2004.

Author's Biography

Shivani Agarwal was born in Maun, Botswana, on April 8, 1978. She received the B.Sc. degree with honours in mathematics from the University of Delhi, India, in 1998; the B.A. degree with honours in computer science from the University of Cambridge, U.K., in 2000; and the M.S. degree in computer science from the University of Illinois, Urbana-Champaign, in 2002. Her awards include the *Prof. P.L. Bhatnagar Memorial Prize* for the best student in mathematics at St. Stephen's College, University of Delhi, in 1998; the *Jawaharlal Nehru Memorial Trust Cambridge Scholarship* for studies at the University of Cambridge during 1998-2000; a Senior Scholarship at Trinity College, University of Cambridge, during 1999-2000; the *C.L. & Jane W-S. Liu Award* for exceptional research promise at the Department of Computer Science, University of Illinois, in 2003; and an Excellent Teaching Assistant Award at the Department of Computer Science, University of Illinois, in 2004.