Learnability of Bipartite Ranking Functions

Shivani Agarwal and Dan Roth

Department of Computer Science,
University of Illinois at Urbana-Champaign,
201 N. Goodwin Avenue, Urbana, IL 61801, USA
{sagarwal, danr}@cs.uiuc.edu

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. We define a model of learnability for ranking functions in a particular setting of the ranking problem known as the bipartite ranking problem, and derive a number of results in this model. Our first main result 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, which measure the richness of a ranking function class in the same way as do the standard VC-dimension related shatter coefficients (growth function) for classes of classification functions, do not grow too quickly. Our second main result 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. Finally, we investigate questions of the computational complexity of learning ranking functions.

1 Introduction

Two decades ago, Valiant [1] proposed a theory of learnability for binary classification functions defined on Boolean domains. His 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 (learning of binary-valued functions) and regression (learning of real-valued functions), two of the most well-studied problems in machine learning. Recently, a new learning problem, namely that of *ranking*, has gained attention in the machine learning community [2, 3, 4, 5]. In ranking, one learns a real-valued function that assigns scores to instances, but the scores themselves do not matter; instead, what is important is the relative ranking of instances induced by those scores. This problem is distinct from both classification and regression, and it is natural to ask whether a similar theoretical understanding can be developed for this problem. This paper constitutes a first step in that direction.

P. Auer and R. Meir (Eds.): COLT 2005, LNAI 3559, pp. 16–31, 2005.

1.1 Previous Results

In the binary classification problem, the learner is given a finite sequence of labeled training examples $\underline{z} = ((x_1, y_1), \ldots, (x_m, y_m))$, where the x_i are instances in some instance space X and the y_i are labels in $Y = \{-1, 1\}$, and the goal is to learn a binary-valued function $h : X \rightarrow Y$ that predicts accurately labels of future instances. In the PAC model, a learning algorithm for a class $\mathcal{H}$ of binary classification functions on X is a function $L : \bigcup_{m=1}^{\infty} (X \times Y)^m \rightarrow \mathcal{H}$ 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 X and any target function $t \in \mathcal{H}$, given a random training sample $\underline{z} = ((x_1, t(x_1)), \ldots, (x_m, t(x_m)))$ 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 $h = L(\underline{z})$ output by L 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 L. A class $\mathcal{H}$ is said to be learnable if there is a learning algorithm for $\mathcal{H}$.

In a classic paper, Blumer et al. [6] showed that the PAC 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 [7] 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 $\mathcal{H}$; 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 [6, 8, 9], often referred to as the 'agnostic' model. In this general model, examples are generated according to an arbitrary joint distribution $\mathcal{D}$ over $X \times \{-1, 1\}$, and a learning algorithm is required to output with high probability a hypothesis $h \in \mathcal{H}$ with prediction error $\mathbf{P}_{(x,y) \sim \mathcal{D}}\{h(x) \neq y\}$ 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.

The regression problem is similar to the classification problem, except that the labels y_i in this case come from $Y = \mathbb{R}$ or $Y = [a, b]$ for some $a, b \in \mathbb{R}$, and the goal is to learn a real-valued function $f : X \rightarrow Y$ that approximates well labels of future instances. An analogous theory of learnability has been developed for this problem, starting with the work of Haussler [8] 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 [10] 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 [11] introduced a new measure of the richness of a real-valued function class known now as the fat-shattering dimension. It was then shown [11, 12, 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.

1.2 Our Results

In the bipartite ranking problem [5, 14], described in detail in Section 2, the learner is given a sequence of 'positive' training examples $\underline{x}^+ = (x_1^+, \ldots, x_m^+)$ and a sequence of 'negative' training examples $\underline{x}^- = (x_1^-, \ldots, x_n^-)$, the x_i^+ and x_j^- being instances in some instance space X, and the goal is to learn a real-valued ranking function $f : X \rightarrow \mathbb{R}$ that ranks future positive instances higher than negative ones, *i.e.*, that assigns higher values to positive instances than to negative ones. We define a model of learnability for ranking functions in the setting of the bipartite ranking problem, and derive a number of results in this model. Our first main result 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 [14], which measure the richness of a ranking function class in the same way as do the standard VC-dimension related shatter coefficients for classes of classification functions, do not grow too quickly. As in the case of classification and regression, the proof of this result makes use of a uniform convergence result of [14] 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 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. 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.

1.3 Organization

We describe the bipartite ranking problem in greater detail in Section 2, and formulate our model of learnability for ranking functions in the setting of this problem in

Section 3. A sufficient condition for learnability in this model is derived in Section 4, and a necessary condition in Section 5. We consider the computational complexity of learning ranking functions in Section 6.

2 The Bipartite Ranking Problem

In the bipartite ranking problem [5, 14], the learner is given a training sample $(\underline{x}^+, \underline{x}^-)$ consisting of a sequence of 'positive' training examples $\underline{x}^+ = (x_1^+, \ldots, x_m^+)$ and a sequence of 'negative' training examples $\underline{x}^- = (x_1^-, \ldots, x_n^-)$, the x_i^+ and x_j^- being instances in some instance space X, and the goal is to learn a real-valued ranking function $f : X \rightarrow \mathbb{R}$ that ranks future positive instances higher than negative ones, *i.e.*, that assigns higher values to positive instances than to negative ones. Such problems arise, 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 given to the learner 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 such that relevant documents are ranked higher than irrelevant ones.

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

$$R_{\mathcal{D}_+,\mathcal{D}_-}(f) = \mathbf{E}_{x^+ \sim \mathcal{D}_+, x^- \sim \mathcal{D}_-} \left\{ \mathbf{I}_{\{f(x^+) < f(x^-)\}} + \frac{1}{2} \mathbf{I}_{\{f(x^+) = f(x^-)\}} \right\}, \qquad (1)$$

where $\mathbf{I}_{\{\cdot\}}$ denotes the indicator variable whose value is one if its argument is true and zero otherwise. 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. A related quantity is the *empirical ranking error* of f with respect to a sample $(\underline{x}^+, \underline{x}^-) \in X^m \times X^n$, denoted by $\hat{R}_{\underline{x}^+,\underline{x}^-}(f)$ and defined as follows:

$$\hat{R}_{\underline{x}^+,\underline{x}^-}(f) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left\{ \mathbf{I}_{\{f(x_i^+) < f(x_j^-)\}} + \frac{1}{2} \mathbf{I}_{\{f(x_i^+) = f(x_j^-)\}} \right\}. \qquad (2)$$

This is simply the fraction of positive-negative pairs in $(\underline{x}^+, \underline{x}^-)$ that are ranked incorrectly by f, assuming again that ties are broken uniformly at random.

Although the bipartite ranking problem shares similarities with the binary classification problem, it should be noted that the two problems are in fact distinct. In particular, it is possible for binary functions obtained by thresholding different real-valued functions to have the same classification errors, while the ranking errors of the real-valued functions differ significantly. For a detailed discussion of this distinction, see [15, 14][1].

[1] In [15, 14], the performance of a ranking function is measured in terms of the area under the ROC curve (AUC); this quantity is simply equal to one minus the empirical ranking error.

3 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

$$R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) = \inf_{g \in \mathcal{F}} R_{\mathcal{D}_+,\mathcal{D}_-}(g)\,. \tag{3}$$

We formalize this idea below, following closely the notation and terminology of Anthony and Bartlett [16]. In what follows, $\mathbb{Q}$ denotes the set of rationals and $\mathbb{N}$ the set of positive integers.

Definition 1 (Learnability). *Let $\mathcal{F}$ be a class of real-valued ranking functions on X. A* learning algorithm *L for $\mathcal{F}$ is a function $L : \left(\bigcup_{m=1}^{\infty} X^m\right) \times \left(\bigcup_{n=1}^{\infty} X^n\right) \rightarrow \mathcal{F}$ 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 X,*

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

The smallest such integer $M(\epsilon,\delta,\rho)$ is called the sample complexity *of L, denoted $M_L(\epsilon,\delta,\rho)$. 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 [16], 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 [16], we refer to this as the *restricted* model.

Definition 2 (Learnability in Restricted Model). *Let $\mathcal{F}$ be a class of real-valued ranking functions on X. A* learning algorithm *L for $\mathcal{F}$ in the restricted model is a function $L : \left(\bigcup_{m=1}^{\infty} X^m\right) \times \left(\bigcup_{n=1}^{\infty} X^n\right) \rightarrow \mathcal{F}$ 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 X for which there is a target function $t \in \mathcal{F}$ such that $R_{\mathcal{D}_+,\mathcal{D}_-}(t) = 0$,*

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

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

Clearly, if a class of ranking functions $\mathcal{F}$ is learnable, then $\mathcal{F}$ is learnable in the restricted model. Note that learnability of $\mathcal{F}$ in the restricted model is equivalent to learnability of the class of classification functions $\mathcal{H} = \big\{h : X \to \{-1,1\} \mid h(x) = \theta(f(x)+\tau)$ for some $f \in \mathcal{F}, \tau \in \mathbb{R}\big\}$, where $\theta(u) = 1$ for $u > 0$ and $\theta(u) = -1$ for $u \leq 0$, in the restricted (PAC) model for classification. However, this equivalence does *not* hold in the standard (agnostic) model.

4 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 [14] of $\mathcal{F}$ do not grow too quickly, and obtain an upper bound on the sample complexity of such an algorithm.

Definition 3 (Bipartite Rank Matrix [14]). *Let $f : X \to \mathbb{R}$ be a ranking function on X, let $m, n \in \mathbb{N}$, and let $\underline{x} = (x_1, \ldots, x_m) \in X^m$, $\underline{x}' = (x'_1, \ldots, x'_n) \in X^n$. The* bipartite rank matrix *of f with respect to $\underline{x}, \underline{x}'$, denoted by $\mathcal{B}_f(\underline{x}, \underline{x}')$, is defined to be the matrix in $\{0, 1/2, 1\}^{m \times n}$ whose (i,j)-th element is given by*

$$[\mathcal{B}_f(\underline{x}, \underline{x}')]_{ij} = \mathbf{I}_{\{f(x_i) > f(x'_j)\}} + \frac{1}{2}\mathbf{I}_{\{f(x_i) = f(x'_j)\}}$$

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

Definition 4 (Bipartite Rank-Shatter Coefficient [14]). *Let $\mathcal{F}$ be a class of real-valued functions on X, and let $m, n \in \mathbb{N}$. The (m,n)*-th bipartite rank-shatter coefficient *of $\mathcal{F}$, denoted by $r(\mathcal{F}, m, n)$, is defined as follows:*

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

Definition 5 (Empirical Error Minimization (EEM) Algorithm). *Let $\mathcal{F}$ be a class of ranking functions on X. Define an* empirical error minimization (EEM) algorithm *for $\mathcal{F}$ to be any function $L : \left(\bigcup_{m=1}^{\infty} X^m\right) \times \left(\bigcup_{n=1}^{\infty} X^n\right) \to \mathcal{F}$ with the property that for any $m, n \in \mathbb{N}$ and any $(\underline{x}^+, \underline{x}^-) \in X^m \times X^n$,*

$$\hat{R}_{\underline{x}^+, \underline{x}^-}(L(\underline{x}^+, \underline{x}^-)) = \min_{g \in \mathcal{F}} \hat{R}_{\underline{x}^+, \underline{x}^-}(g) .$$

Theorem 1. *Let $\mathcal{F}$ be a class of ranking functions on X, and let L 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 L is a learning algorithm for $\mathcal{F}$ with sample complexity*

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

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

The proof of this result makes use of the following uniform convergence result for the ranking error given in [14][2]:

Theorem 2 ([14]). *Let $\mathcal{F}$ be a class of ranking functions on X, and let $m, n \in \mathbb{N}$. Then for any distributions $\mathcal{D}_+, \mathcal{D}_-$ on X and for any $\epsilon > 0$,*

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

Proof (of Theorem 1). It can be shown using standard techniques [16] that for any $m, n \in \mathbb{N}$, any $(\underline{x}^+, \underline{x}^-) \in X^m \times X^n$ and any distributions $\mathcal{D}_+, \mathcal{D}_-$ on X,

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

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 $m, n \in \mathbb{N}$. Let $\rho \in (0, 1) \cup \mathbb{Q}$ and $\epsilon, \delta \in (0, 1)$, and let $\mathcal{D}_+, \mathcal{D}_-$ be any distributions on 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

$$\begin{aligned}
&\mathbf{P}_{\underline{x}^+ \sim \mathcal{D}_+^m, \underline{x}^- \sim \mathcal{D}_-^n} \left\{ R_{\mathcal{D}_+, \mathcal{D}_-}(L(\underline{x}^+, \underline{x}^-)) - R^*_{\mathcal{D}_+, \mathcal{D}_-}(\mathcal{F}) \;\geq\; \epsilon \right\} && (4)\\
&\quad\leq \mathbf{P}_{\underline{x}^+ \sim \mathcal{D}_+^m, \underline{x}^- \sim \mathcal{D}_-^n} \left\{ \sup_{f \in \mathcal{F}} \left| \hat{R}_{\underline{x}^+, \underline{x}^-}(f) - R_{\mathcal{D}_+, \mathcal{D}_-}(f) \right| \;\geq\; \epsilon/2 \right\} \\
&\quad\leq 4 \cdot r(\mathcal{F}, 2\rho M, 2(1-\rho)M) \cdot e^{-\rho(1-\rho)M\epsilon^2/32} && \text{(by Theorem 2)}\\
&\quad\leq 4 \cdot c_1 (4\rho(1-\rho)M^2)^{c_2} \cdot e^{-\rho(1-\rho)M\epsilon^2/32} .
\end{aligned}$$

Therefore, to make the probability in Eq. (4) smaller than δ, it is sufficient if

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

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

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

Using this and simplifying terms, we get 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) smaller than δ. The result then follows from the definition of sample complexity (Definition 1). □

[2] The uniform convergence result in [14] is given for the area under the ROC curve (AUC); as mentioned previously, this quantity is simply equal to one minus the empirical ranking error.

Notice that 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. Similar observations regarding the role of the skew ρ in ranking have been made in different contexts in [15, 14]. Theorem 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 1 (Finite function classes). Let $\mathcal{F}$ be a finite class of ranking functions on some instance space X. Then $r(\mathcal{F}, m, n) \leq |\mathcal{F}|$ for all $m, n \in \mathbb{N}$. Thus we have from Theorem 1 that $\mathcal{F}$ is learnable; in particular, taking $c_1 = |\mathcal{F}|$, $c_2 = 0$, we have that any EEM algorithm L for $\mathcal{F}$ is a learning algorithm for $\mathcal{F}$ with sample complexity[3]

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

Example 2 (Linear ranking functions). Let $\mathcal{F}_{\text{lin}(d)}$ be the class of linear ranking functions on $\mathbb{R}^d$. Then it can be shown [14] that $r(\mathcal{F}_{\text{lin}(d)}, m, n) \leq (2emn/d)^d$ for all $m, n \in \mathbb{N}$. Thus we have from Theorem 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 L for $\mathcal{F}_{\text{lin}(d)}$ is a learning algorithm for $\mathcal{F}_{\text{lin}(d)}$ with sample complexity

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

Example 3 (Polynomial ranking functions). Let $q \in \mathbb{N}$, and let $\mathcal{F}_{\text{poly}(d,q)}$ be the class of polynomial ranking functions on R^d with degree less than or equal to q. Then it can be shown [14] that $r(\mathcal{F}_{\text{poly}(d,q)}, m, n) \leq (2emn/C(d,q))^{C(d,q)}$ 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 1 that $\mathcal{F}_{\text{poly}(d,q)}$ 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 L for $\mathcal{F}_{\text{poly}(d,q)}$ is a learning algorithm for $\mathcal{F}_{\text{poly}(d,q)}$ with sample complexity

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

5 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.

[3] It is in fact possible to obtain a slightly tighter upper bound in this case using a different uniform convergence result of [14] for finite function classes.

Definition 6 (Rank-Shattering). *Let $\mathcal{F}$ be a class of real-valued functions on 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 X. For each $i \in \{1, \ldots, r\}$, $b \in \{0,1\}^r$, define*

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

We say that $\mathcal{F}$ rank-shatters *S iff 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 7 (Rank Dimension). *Let $\mathcal{F}$ be a class of real-valued functions on X. Define the* rank dimension *of $\mathcal{F}$, denoted by* rank-dim($\mathcal{F}$)*, to be the largest positive integer r for which there exists a set of r pairs of instances in X that is rank-shattered by $\mathcal{F}$.*

Theorem 3. *Let $\mathcal{F}$ be a class of ranking functions on X with* rank-dim$(\mathcal{F}) = r$*. Then for any function $L : \left(\bigcup_{m=1}^{\infty} X^m\right) \times \left(\bigcup_{n=1}^{\infty} X^n\right) \to \mathcal{F}$, any $m, n \in \mathbb{N}$ such that $m + n \geq 2r$, and any $\epsilon > 0$, there exist distributions $\mathcal{D}_+, \mathcal{D}_-$ on X such that*

$$\mathbf{E}_{\underline{x}^+ \sim \mathcal{D}_+^m, \underline{x}^- \sim \mathcal{D}_-^n}\left\{ R_{\mathcal{D}_+,\mathcal{D}_-}(L(\underline{x}^+, \underline{x}^-)) - R^*_{\mathcal{D}_+,\mathcal{D}_-}(\mathcal{F}) \right\}$$
$$\geq \frac{1}{2^{10}}\sqrt{\frac{r}{m+n}}\left(1 - \sqrt{1 - e^{-(2m/(m+n)+1)}}\right)^2 \left(1 - \sqrt{1 - e^{-(2n/(m+n)+1)}}\right)^2.$$

Proof (sketch). 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 L 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 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 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 6, it can be verified that for any $f : X \to \mathbb{R}$,

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

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 $i, j \in \{1, \ldots, r\}$, $f_b(w_i^{b+}) > f_b(w_j^{b-})$. From the above equation this gives

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

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

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

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

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

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, [16–Chapter 5]); the details are considerably more involved and are omitted for lack of space (see [17] for complete details). Denoting the uniform distribution over $\{0,1\}^r$ by $\mathcal{U}$, what we get is that for any $\alpha > 0$,

$$\begin{aligned}&\mathbf{E}_{b\sim\mathcal{U}}\left\{\mathbf{E}_{\underline{x}^+\sim\mathcal{D}_{b+}^m,\underline{x}^-\sim\mathcal{D}_{b-}^n}\left\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{\underline{x}}) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\right\}\right\}\\ &\geq \frac{\alpha}{2^{10}}\left(1-\sqrt{1-e^{-(2m/r+1)\alpha^2/(1-\alpha^2)}}\right)^2\left(1-\sqrt{1-e^{-(2n/r+1)\alpha^2/(1-\alpha^2)}}\right)^2.\end{aligned}$$

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

$$\begin{aligned}&\mathbf{E}_{b\sim\mathcal{U}}\left\{\mathbf{E}_{\underline{x}^+\sim\mathcal{D}_{b+}^m,\underline{x}^-\sim\mathcal{D}_{b-}^n}\left\{R_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(f_{\underline{x}}) - R^*_{\mathcal{D}_{b+},\mathcal{D}_{b-}}(\mathcal{F})\right\}\right\}\\ &\geq \frac{1}{2^{10}}\sqrt{\frac{r}{m+n}}\left(1-\sqrt{1-e^{-(2m/(m+n)+1)}}\right)^2\left(1-\sqrt{1-e^{-(2n/(m+n)+1)}}\right)^2. \qquad \square\end{aligned}$$

Corollary 1. *Let $\mathcal{F}$ be a class of ranking functions on X with* rank-dim($\mathcal{F}$) $= r$*, and let L be any learning algorithm for $\mathcal{F}$. Then L has sample complexity*

$$M_L(\epsilon,\delta,\rho) \geq \frac{r}{2^{20}(\epsilon+\delta)^2}\left(1-\sqrt{1-e^{-(2\rho+1)}}\right)^4\left(1-\sqrt{1-e^{-(2(1-\rho)+1)}}\right)^4.$$

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

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

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 X,

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

Theorem 3 then implies that

$$\epsilon + \delta \geq \frac{1}{2^{10}} \sqrt{\frac{r}{M}} \left(1 - \sqrt{1 - e^{-(2\rho+1)}}\right)^2 \left(1 - \sqrt{1 - e^{-(2(1-\rho)+1)}}\right)^2 .$$

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

Proof. This follows directly from Corollary 1. □

Example 4. Let $\mathcal{F}$ be the class of all ranking functions $f : \mathbb{R} \rightarrow \mathbb{R}$ 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 2, $\mathcal{F}$ is not learnable.

Remark 1. We note that since the distributions constructed in the proof of Theorem 3 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 2.

6 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 [16, 6], we formalize this by defining learning algorithms for *graded* function classes. For each $d \in \mathbb{N}$, let X_d be a subset of $\mathbb{R}^d$, and let $\mathcal{F}_d$ be a set of ranking functions on 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 $L : \bigcup_{d=1}^{\infty} \left(\left(\bigcup_{m=1}^{\infty} X_d^m \right) \times \left(\bigcup_{n=1}^{\infty} X_d^n \right) \right) \rightarrow \mathcal{F}$ such that if $(\underline{x}^+, \underline{x}^-) \in X_d^m \times X_d^n$, then $L(\underline{x}^+, \underline{x}^-) \in \mathcal{F}_d$, and for each d, L is a learning algorithm for $\mathcal{F}_d$ (in the sense of Definition 1). Assuming that learning algorithms are computable functions, we can now ask how the computational complexity of a learning algorithm L for a graded class of ranking functions $\mathcal{F} = \bigcup \mathcal{F}_d$ grows with d.

Definition 8 (Efficient Learnability). *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions and let L be a learning algorithm for $\mathcal{F}$. We say that L is* efficient *if*

(i) the worst-case time complexity $T_L(m, n, d)$ of L on samples $(\underline{x}^+, \underline{x}^-) \in X_d^m \times X_d^n$ is polynomial[4] in m, n and d, and

(ii) the sample complexity $M_L(\epsilon, \delta, \rho, d)$ of L 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 and 5 can be extended to efficient learnability as follows.

Definition 9 (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 $(\underline{x}^+, \underline{x}^-) \in X_d^m \times X_d^n$, and in time polynomial in m, n and d, returns a ranking function $f \in \mathcal{F}_d$ such that $\hat{R}_{\underline{x}^+, \underline{x}^-}(f) = \min_{g \in \mathcal{F}_d} \hat{R}_{\underline{x}^+, \underline{x}^-}(g)$.*

Theorem 4. *Let $\mathcal{F} = \bigcup \mathcal{F}_d$ be a graded class of ranking functions, and suppose that there exist functions $c_1 : \mathbb{N} \rightarrow \mathbb{R}^+$, $c_2 : \mathbb{N} \rightarrow \mathbb{R}^+ \cup \{0\}$ 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 L is an efficient EEM algorithm for $\mathcal{F}$. Then

(i) by Theorem 1, L is a learning algorithm for $\mathcal{F}_d$ for each d and therefore a learning algorithm for $\mathcal{F}$,

(ii) by Definition 9, the time complexity $T_L(m, n, d)$ of L on $\mathcal{F}_d$ is polynomial in m, n and d, and

(iii) by Theorem 1, the sample complexity $M_L(\epsilon, \delta, \rho, d)$ of L 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, L is an efficient learning algorithm for $\mathcal{F}$. □

Theorem 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* rank-dim($\mathcal{F}_d$) *is polynomial in d.*

[4] In the logarithmic cost model of computation [18], the time complexity is also allowed to depend polynomially on the number of bits required to represent the input.

Proof. This follows directly from Definition 8 and Corollary 1. □

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 [16], 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 similar to that for classification; we omit the details.

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

Theorem 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 for $\mathcal{F}$-FIT, i.e., $\mathcal{F}$-FIT is in* RP.

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 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 Theorem 6. 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 $\mathcal{H}_{\text{lin}(d)} = \big\{h : \mathbb{R}^d \rightarrow \{-1, 0, 1\} \mid h(\mathbf{x}) = \text{sign}(\sum_{l=1}^d w_l x_l + \theta) \text{ for some } \mathbf{w} \in \mathbb{R}^d, \theta \in \mathbb{R}\big\}$. Given a function $h \in \mathcal{H}_{\text{lin}(d)}$ and a sample $\underline{z} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)) \in (\mathbb{R}^d \times \{-1, 1\})^m$, define the *empirical error* of h with respect to $\underline{z}$, denoted by $\hat{\text{er}}_{\underline{z}}(h)$, as follows:

$$\hat{\text{er}}_{\underline{z}}(h) = \frac{1}{m} \sum_{i=1}^m \Big\{ \mathbf{I}_{\{h(\mathbf{x}_i) \neq 0\}} \mathbf{I}_{\{h(\mathbf{x}_i) \neq y_i\}} + \frac{1}{2} \mathbf{I}_{\{h(\mathbf{x}_i) = 0\}} \Big\} .$$

Let $\mathcal{H}_{\text{lin}} = \bigcup \mathcal{H}_{\text{lin}(d)}$, and define the following decision problem associated with $\mathcal{H}_{\text{lin}}$:

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

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 [16], 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 $\underline{z} = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)) \in (\mathbb{R}^d \times \{-1, 1\})^m$, $k' \in \{1, \ldots, m\}$ be an instance of $\mathcal{H}_{\text{lin}}$-FIT. We construct from $\underline{z}, k'$ an instance $(\underline{x}^+, \underline{x}^-) \in (\mathbb{R}^{d+1})^m \times (\mathbb{R}^{d+1}), k \in \{1, \ldots, m\}$ 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 $\underline{x}^+ = (x_1^+, \ldots, x_m^+)$, $\underline{x}^- = (x_1^-)$, and $k = k'$. We claim that there exists $h \in \mathcal{H}_{\text{lin}(d)}$ with $\hat{\text{er}}_{\underline{z}}(h) \leq k'/m$ if and only if there exists $f \in \mathcal{F}_{\text{lin}(d+1)}$ with $\hat{R}_{\underline{x}^+,\underline{x}^-}(f) \leq k/m$.

First, suppose there exists $h \in \mathcal{H}_{\text{lin}(d)}$ with $\hat{\text{er}}_{\underline{z}}(h) \leq k'/m$, 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, $f \in \mathcal{F}_{\text{lin}(d+1)}$, and it can be verified that $\hat{R}_{\underline{x}^+,\underline{x}^-}(f) = \hat{\text{er}}_{\underline{z}}(h) \leq k'/m = k/m$. Conversely, suppose there exists $f \in \mathcal{F}_{\text{lin}(d+1)}$ with $\hat{R}_{\underline{x}^+,\underline{x}^-}(f) \leq k/m$, 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, $h \in \mathcal{H}_{\text{lin}(d)}$, and it can be verified that $\hat{\text{er}}_{\underline{z}}(h) = \hat{R}_{\underline{x}^+,\underline{x}^-}(f) \leq k/m = k'/m$.

Since the time required to construct the instance $(\underline{x}^+, \underline{x}^-), k$ from $\underline{z}, k'$ is polynomial in the size of $\underline{z}, k'$, 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 9 and Theorem 4 to the restricted model case.

Definition 10 (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 L such that, given any sample $(\underline{x}^+, \underline{x}^-) \in X_d^m \times X_d^n$ for which there exists a target function $t \in \mathcal{F}_d$ satisfying $\hat{R}_{\underline{x}^+,\underline{x}^-}(t) = 0$, L halts in time polynomial in m, n and d and returns a ranking function $f \in \mathcal{F}_d$ such that $\hat{R}_{\underline{x}^+,\underline{x}^-}(f) = 0$.*

Theorem 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 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 (sketch). As discussed in Example 2 (Section 4), $r(\mathcal{F}_{\text{lin}(d)}, m, n) \leq (2emn/d)^d$ for all $d, m, n \in \mathbb{N}$. Therefore, by Theorem 8, it suffices to show the existence of an efficient consistent-hypothesis-finder for $\mathcal{F}_{\text{lin}}$. This can be done by formulating a linear program such that, given a training sample $(\underline{x}^+, \underline{x}^-) \in (\mathbb{R}^d)^m \times (\mathbb{R}^d)^n$ for which there exists a target function $t \in \mathcal{F}_{\text{lin}(d)}$ satisfying $\hat{R}_{\underline{x}^+,\underline{x}^-}(t) = 0$, the solution of the linear program gives a ranking function $f \in \mathcal{F}_{\text{lin}(d)}$ such that $\hat{R}_{\underline{x}^+,\underline{x}^-}(f) = 0$ (see [17] for details). Solving the linear program using a polynomial-time linear programming algorithm such as Karmarkar's [19] then constitutes an efficient consistent-hypothesis-finder for $\mathcal{F}_{\text{lin}}$. □

Remark 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 3. In the above proof, we could also have used a linear program that finds a classification function $h \in \mathcal{H}_{\mathrm{lin}(d)}$ of the form $h(\mathbf{x}) = \mathrm{sign}(\sum_{l=1}^{d} w_l x_l + \theta)$ 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 10. *Let $\mathcal{F}^b_{\mathrm{lin}} = \bigcup \mathcal{F}^b_{\mathrm{lin}(d)}$, where $\mathcal{F}^b_{\mathrm{lin}(d)}$ is the class of linear ranking functions on $\{0,1\}^d$. If* RP $\neq$ NP*, then $\mathcal{F}^b_{\mathrm{lin}}$ is not efficiently learnable in the restricted model.*

Proof (sketch). Let, if possible, $\mathcal{F}^b_{\mathrm{lin}}$ be efficiently learnable in the restricted model. Then there is an efficient randomized consistent-hypothesis-finder $\mathcal{A}$ for $\mathcal{F}^b_{\mathrm{lin}}$ (see [16, 17]). Clearly, $\mathcal{A}$ can be used to construct an efficient randomized consistent-hypothesis-finder for $\mathcal{H}^b_{\mathrm{lin}} = \bigcup \mathcal{H}^b_{\mathrm{lin}(d)}$, where $\mathcal{H}^b_{\mathrm{lin}(d)}$ is the class of Boolean threshold functions on $\{0,1\}^d$. This, in turn, implies the existence of an efficient learning algorithm for $\mathcal{H}^b_{\mathrm{lin}}$ in the restricted (PAC) model (see [16]). Since the problem of learning Boolean threshold functions in the PAC model is known to be NP-hard [20], this implies RP = NP. Thus, if RP $\neq$ NP, then $\mathcal{F}^b_{\mathrm{lin}}$ 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 $\mathcal{F}_{\mathrm{poly}(q)} = \bigcup \mathcal{F}_{\mathrm{poly}(d,q)}$, where $\mathcal{F}_{\mathrm{poly}(d,q)}$ 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 $\mathcal{F}^b_{\mathrm{poly}(q)} = \bigcup \mathcal{F}^b_{\mathrm{poly}(d,q)}$, where $\mathcal{F}^b_{\mathrm{poly}(d,q)}$ 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.

7 Conclusion and Open Questions

Our goal in this paper has been to initiate a formal study of learnability for 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, can the rank dimension be related to previous quantities (such as the VC-dimension or pseudo-dimension), or is it a fundamentally new quantity? So far, we have not been able to find a relation to earlier dimensions. Third, 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?

Acknowledgments

We would like to express warm thanks to Sariel Har-Peled for many valuable discussions. The exposition in this paper is influenced in large parts by the excellent text of Anthony and Bartlett [16]. This research was supported in part by NSF ITR grants IIS 00-85980 and IIS 00-85836 and a grant from the ONR-TRECC program.

References

1. Valiant, L.G.: A theory of the learnable. Communications of the ACM (1984) 1134–1142
2. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. Journal of Artificial Intelligence Research **10** (1999) 243–270
3. Herbrich, R., Graepel, T., Obermayer, K.: Large margin rank boundaries for ordinal regression. Advances in Large Margin Classifiers (2000) 115–132
4. Crammer, K., Singer, Y.: Pranking with ranking. In Dietterich, T.G., Becker, S., Ghahramani, Z., eds.: Advances in Neural Information Processing Systems 14, MIT Press (2002) 641–647
5. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An efficient boosting algorithm for combining preferences. Journal of Machine Learning Research **4** (2003) 933–969
6. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.K.: Learnability and the Vapnik-Chervonenkis dimension. Journal of the ACM **36** (1989) 929–965
7. Vapnik, V.N., Chervonenkis, A.: On the uniform convergence of relative frequencies of events to their probabilities. Theory of Probability and its Applications **16** (1971) 264–280
8. Haussler, D.: Decision theoretic generalizations of the PAC model for neural net and other learning applications. Information and Computation **100** (1992) 78–150
9. Kearns, M.J., Schapire, R.E., Sellie, L.M.: Toward efficient agnostic learning. Machine Learning **17** (1994) 115–141
10. Pollard, D.: Convergence of Stochastic Processes. Springer-Verlag (1984)
11. Kearns, M.J., Schapire, R.E.: Efficient distribution-free learning of probabilistic concepts. Journal of Computer and System Sciences **48** (1994) 464–497
12. Alon, N., Ben-David, S., Cesa-Bianchi, N., , Haussler, D.: Scale-sensitive dimensions, uniform convergence, and learnability. Journal of the ACM **44** (1997) 615–631
13. Bartlett, P.L., Long, P.M., Williamson, R.C.: Fat-shattering and the learnability of real-valued functions. Journal of Computer and System Sciences **52** (1996) 434–452
14. Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., Roth, D.: Generalization bounds for the area under the ROC curve. Journal of Machine Learning Research (2005) 393–425
15. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. In: Advances in Neural Information Processing Systems 16, MIT Press (2004)
16. Anthony, M., Bartlett, P.L.: Learning in Neural Networks: Theoretical Foundations. Cambridge University Press (1999)
17. Agarwal, S.: A Study of the Bipartite Ranking Problem in Machine Learning. PhD thesis, University of Illinois at Urbana-Champaign (2005)
18. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley (1974)
19. Karmarkar, N.: A new polynomial-time algorithm for linear programming. Combinatorica **4** (1984) 373–395
20. Pitt, L., Valiant, L.G.: Computational limitations on learning from examples. Journal of the ACM **35** (1988) 965–984