UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

THEORY-GUIDED ALGORITHM DESIGN FOR SCALABLE MACHINE LEARNING

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

Doctor of Philosophy

By

YITING CAO
Norman, Oklahoma
May 2023

THEORY-GUIDED ALGORITHM DESIGN
FOR SCALABLE MACHINE LEARNING

A DISSERTATION APPROVED FOR
THE SCHOOL OF COMPUTER SCIENCE

BY THE COMMITTEE CONSISTING OF

Dr. Chao Lan

Dr. Dean Hougen

Dr. Dimitris Diochnos

Dr. Christian Remling

Abstract

My thesis focuses on designing scalable machine learning algorithms leveraging theoretical advances in mathematics. In particular, I investigate two directions where scalability plays an important role: fair machine learning and randomized feature representations. In fair machine learning, my research concentrates on achieving individual fairness in the single model and decoupled model settings with minimum data labeling budgets. For randomized feature representations, I propose a model-agnostic framework for designing computationally efficient randomized machine learning algorithms with provable performance guarantees, which demonstrates that it is not necessary for individual models to be weakly trained before they are optimally ensembled. Furthermore, I also contribute to the scalable estimation of Kernel matrix spectral norm. Specifically, I propose to apply sketching techniques to efficiently estimate the spectral norm, theoretically derive the estimation error and empirically demonstrate the estimation efficiency in a time-constrained setting.

Acknowledgments

First and foremost, I want to thank my advisor Dr. Chao Lan for always believing in my potential and consistently supporting my career choices, for example, consulting another faculty for opinion on certain research topics, having summer internship at Quora, and graduating early. Next, I want to thank my committee members for helpful discussions during our meetings. I also want to thank the School of Computer Science at University of Oklahoma for providing support throughout my years here. Finally, I want to thank my parents for supporting my educational journey. I will not become who I am today without them.

All my papers mentioned in this thesis are joint work with Dr. Chao Lan.

Contents

List of Figures

Chapter 1

Introduction

The 21^{st} century has seen a significant growth in data volume thanks to the internet. More importantly, the emergence of larger storage media and increased performance of computer processors also contribute to the recent prosperity of machine learning research. While the aforementioned advances make most machine learning models practical in industrial applications, some nice techniques fail to see the light of day due to their poor scalability. That is, such algorithms quickly become cost-prohibitive as the size of data, typically the number of samples, grows. Hence they remain unknown to most machine learning practitioners despite being considered interesting by the theory community. To improve practicality of those techniques, numerous research has been done on improving the scalability of machine learning algorithms, and my efforts are a part of that.

My thesis focuses on designing scalable machine learning algorithms leveraging theoretical advances in mathematics. In particular, I investigate two directions where scalability plays an important role: fair machine learning and randomized feature representations. Next, I will give a very brief introduction to these fields. More details on related work will be included in later sections.

1.1 Fair Machine Learning

As machine learning is increasingly applied to different aspects of our daily life, some of those aspects inevitably affect decision-making, directly or indirectly. Those decision-making are frequently life-changing, Eg. deciding whether or not to issue a credit card to someone, or whether someone should get early release from prison. Social activist organizations have found that some of these machine learning systems exacerbate discrimination in our society. A famous and influential example of such finding is ProPublica's report on the COMPAS software being racially biased. The COMPAS software predicts recidivism, and it is used by many US courts. Despite

social awareness of existing unfairness in machine learning applications, solving this problem is an uphill battle. To start with, fairness as a social concept is hard to quantify. There is still no concensus on which fairness criteria is the best, but it is widely accepted that fairness criterion fall into two categories: individual fairness and group fairness. Individual fairness requires model predictions be similar on similar individuals, while group fairness needs overall model predictions to be close between groups. My research [CL22a, CL22b] concentrates on achieving individual fairness in the single model and decoupled model settings.

1.2 Randomized Feature Representations

Randomization has long been the method of choice for reducing computational cost. It is a powerful tool for trade-off between accuracy and computational cost. Well-known applications of randomized algorithm in machine learning include random fourier features for scaling up kernel machines [RR08], random projection for classfication and regression [AV06], randomized low-rank matrix factorization for collaborative filtering [DM16], random forest, dropout for deep learning, stochastic gradient descent etc. One common trait for all these established methods is that their design are quite model-specific [RR08, Bre01, PPS94] and their application areas are a bit narrow. In [CL22c], I propose a model-agnostic framework for designing randomized machine learning algorithms. It turns out that it is not necessary for individual models to be weakly trained before they are ensembled.

While [CL22c] suggest new ways for learning more efficiently through randomization, my research also contribute to performing classical machine learning operations more efficiently. Namely, Chapter 7 improves the scalability of spectral norm computation on non-sparse matrices by applying sketching methods. Spectral norm is crucial for spectral learning, a subfield of machine learning that concerns solving an optimization problem which involves regularization with a spectral penalty term. Spectral learning has application in collaborative filtering, multi-task learning, etc [AM05, AMP10]. More specifically, spectral norm regularization is found powerful under various settings in matrix completion, graph embeddings, adversarial learning, and computer vision [MHT10, SCS+15, BMCM19, RKH20, ZHX+21]. Hence, improving the scalability of spectral norm computation in Chapter 7 is a huge step forward for many machine learning applications.

1.3 Outline

A reader should skip the Preliminaries chapter and refer back to it when the fundamental concepts are mentioned in later chapters. In Part I, I will introduce my work on active learning for individual fairness [CL22a, CL22b], where [CL22a] examines the single model setting and [CL22b] consider the case for decoupled model. In Part II, I will explain my research on designing randomized machine learning methods for scalability [CL22c] and Chapter 7. [CL22c] relaxes the weakly-learned assumption for traditional ensemble models, and Chapter 7 finds a faster, more accurate approximation of spectral norm for non-sparse symmetric matrices in the low computational budget setting. Part I and Part II include slightly modified versions of [CL22a, CL22b, CL22c] and Chapter 7 is a paper submission currently under review.

Chapter 2

Preliminaries

2.1 Tools from Matrix Analysis and High Dimensional Probability

Definition 2.1.1 (Subgaussian)**.** Let X be a random variable. X is subgaussian if for some $K_0 > 0$ the tails of X satisfy

$$P[|X| \geq t] \leq 2\exp(-t^2/K_0^2) \text{ for all } t \geq 0. \tag{2.1}$$

This is Proposition 2.5.2 in [Ver18].

Lemma 2.1.2 (Weyl's Inequality)**.** *For any symmetric matrices S and T with the same dimensions, we have*

$$\max_i |\lambda_i(S) - \lambda_i(T)| \leq \|S - T\|_2. \tag{2.2}$$

This is Theorem 4.5.3 in [Ver18].

Lemma 2.1.3 (Matrix Bernstein's Inequality)**.** *Let $X_1, \ldots, X_D$ be independent mean zero, $n \times n$ symmetric random matrices, such that $\|X_i\| \leq C$ almost surely for all i. Then, for every $t \geq 0$, we have*

$$P[|\lambda_1\big(\sum_{i=1}^{D} X_i\big)| \geq t] \leq 2n\exp(-\frac{t^2/2}{\sigma^2 + Ct/3}). \tag{2.3}$$

Here $\sigma^2 = \|\sum_{i=1}^{D} EX_i^2\|$ is the norm of the matrix variance of the sum.

This is Theorem 5.4.1 in [Ver18].

We state a version of Cauchy's Interlacing Theorem.

Lemma 2.1.4 (Cauchy's Interlacing Theorem)**.** *Assume the eigenvalues are in ascending order. Suppose that $A \in M_n$ is Hermitian and has the block form*

$$A = \begin{bmatrix} B & C \\ C^* & D \end{bmatrix} \tag{2.4}$$

where $B \in M_m$, $C \in M_{m,n-m}$, and $D \in M_{n-m}$. Then for each $1 \leq k \leq m$,

$$\lambda_k(A) \leq \lambda_k(B) \leq \lambda_{k+n-m}(A). \tag{2.5}$$

When $m = n - 1$,

$$\lambda_1(A) \leq \lambda_1(B) \leq \lambda_2(A) \leq \ldots \leq \lambda_{n-1}(B) \leq \lambda_n(A). \tag{2.6}$$

It is immediately clear from here that $\lambda_{\max}(A) \geq \lambda_{\max}(A')$ whenever A' is a submatrix of A so $\|A\| \geq |a_{ij}|$ for all i, j.

Lemma 2.1.5. *For any $d \times n$ matrix Z,*

$$s_k(Z) = \sqrt{\lambda_k(Z^T Z)}. \tag{2.7}$$

This is immediate from the singular value decomposition of Z and spectral decomposition of $Z^T Z$, $Z^T Z = V^* \Sigma U^* U \Sigma V^* = V \Sigma^2 V^*$.

2.2 Tools from Previous Research

Lemma 2.2.1 (Exponentially fast convergence of RFF)**.** *For any fixed pair of points x, y, their random fourier features satisfy*

$$P[|z(x)^T z(y) - k(x, y)| \geq \varepsilon] \leq 2 \exp(-D\varepsilon^2/4). \tag{2.8}$$

This comes from the paragraph above Claim 1 of [RR07]. By Definition 2.1.1, this means that the random variable $z(x)^T z(y) - k(x, y)$ (wrt randomness of RFF) is subgaussian with $K_0 = \frac{2}{\sqrt{D}}$.

Lemma 2.2.2 (uniform convergence of RFF (Claim 1 [RR07]))**.** *Let M be a compact subset of R^d with diameter $diam(M)$. Then for the mapping z defined in Algorithm 5, we have*

$$P[\sup_{x,y \in M} |z(x)^T z(y) - k(x, y)| \geq \varepsilon] \tag{2.9}$$

$$\leq 2^8 (\frac{\sigma_p diam(M)}{\varepsilon})^2 \exp(-\frac{D\varepsilon^2}{4(d+2)}), \tag{2.10}$$

where $\sigma_p^2 = 2d\gamma$ for the spherical gaussian kernel $k(x, y) = \exp(-\gamma\|x - y\|^2)$.

Repeating the argument to obtain Lemma 2.2.2 from Lemma 2.2.1 will be useful in the proof of Theorem 7.4.2.

Lemma 2.2.3 (Theorem 3 [DMC05])**.** *Suppose G is an $n \times n$ symmetric positive semi-definite matrix, let $k \leq c$ be a rank parameter, and let $\tilde{G}_k =$*

$CW_K^+C^T$ be constructed from Algorithm 6 by sampling c columns of G with probabilities $\{p_i\}_{i=1}^n$ such that

$$p_i = \frac{G_{ii}^2}{\sum_{i=1}^n G_{ii}^2}. \tag{2.11}$$

Let $r = rank(W)$ and let G_k be the best rank-k approximation to G. In addition, let $\varepsilon > 0$. If $c \geq \frac{4(1+\sqrt{8\log(1/\delta)})^2}{\varepsilon^2}$, then with probability at least $1-\delta$,

$$\|G - \tilde{G}_k\|_2 \leq \|G - G_k\|_2 + \varepsilon \sum_{i=1}^n G_{ii}^2. \tag{2.12}$$

Part I

Active Fair Learning

Chapter 3

Background on Active Fair Learning

3.1 Introduction

Fairness of machine learning models is a critical societal concern these days. It may be surprising that machine learning algorithms lead to discriminatory decisions against minority people. However, many case studies [FPCDG16, CW18] confirm this unfortunate truth. Soon after this problem is exposed, researchers have been trying hard to mitigate this issue [CR18, MMS+21, PS22]. Some exploratory studies established that fairness metrics generally falls into two categories, group fairness and individual fairness. Generally speaking, group fairness is achieved when the model output has minimal disparity across different groups, while individual fairness demands comparable predictions on comparable individuals. The following study will focus on the latter.

Lipschitz condition of a prediction model is originally proposed as an individual fairness metric [DHP+12]. Later, [YR18] proposed a relaxed version called approximate metric-fairness which is a probabilistic and almost Lipschitz condition. Other individual fairness metrics are proposed to adapt to different situations and [Ilv20, MYBS20] introduces how to design them. Even though individual and group fairness are usually treated separately, [ZWS+13, SMKR19] explain how to combine individual and group fairness. The limited-resource setting is previously explored in [KRR18, BJW20], before my active fair learning study. To compare the active setting with the passive counterpart, the analysis on sample complexity for achieving individual fairness in [BL19, SCM20] is used.

My research emphasizes obtaining a more efficient sample complexity for achieving individual fairness. This is done by adapting active learning tricks to serve individual fairness goals. Given a bias budget ε, existing methods [YR18, BL19, SCM20] give a $O(\frac{1}{\varepsilon^2})$ sample complexity. The following

sections will show that this can be improved to $O(\log \frac{1}{\varepsilon})$ through a variation of active learning. To clear the way for further discussions on algorithm design and analysis, I first introduce a new form of approximate metric-fairness (AMF) based on uniform continuity and prove its equivalence to the original form leveraging an established connection between almost Lipschitz continuity and uniform continuity [Van91]. Next, I will present a passive AMF learner and show its generalization performance in achieving individual fairness. From there, I will explain the design of an active AMF learner which selects samples whose neighboring predictions are very different from its own. I will show that with high probability, this learner only takes $O(\log \frac{1}{\varepsilon})$ sample labels to bound unfairness with ε under proper conditions. This analysis assumes boundedness of a meticulously defined counter approximate metric fairness coefficient and an example calculation will be shown to clarify.

The proposed active AMF algorithm [CL22a] in Chapter 4 is experimented on three real world datasets. It is observed to improve individual fairness of both linear and non-linear models faster than both its passive counterpart and canonical active learning, and such improvement is achieved while maintaining similar accuracy, achieving a more productive fairness-accuracy trade-off.

3.2 Related Work

3.2.1 Fair Learning

Machine learning is increasingly used in today's world including applications that is closely related to healthcare and hiring decisions. Giving fair treatment for every human being is vital to maintaining social mobility. Unfortunately, a lot of machine learning algorithms are only exacerbating the existing racial and gender-based tensions [FPCDG16, CW18]. In an effort to ameliorate the situation, researchers are motivated to find algorithmic improvements that would achieve better fairness as measured by common fairness metrics [GHZGW16, AIK18, GHZGW18, RY18, MOS20]. It is widely known that such fairness metrics generally belongs to two classes, individual fairness and group fairness. The following research will address individual fairness. After [YR18] relaxed the individual fairness criterion based on Lipschitz condition [DHP$^+$12], its probabilistic and almost Lipschitz condition becomes the new go-to metric for researchers and many later studies [KRR18, BL19, BJW20] apply it to their evaluation. The active fair learning proposed as follows is also built on AMF. However, we adopt a new and provably equivalent criteria based on uniform continuity instead of almost Lipschitz.

Some research in individual fairness focus on finding a proper similarity

metric for individual samples [Ilv20, MYBS20]. In the following study, I will assume a metric is given and emphasize how to achieve individual fairness *efficiently* through active learning. To my knowledge, research on individual fairness before [CL22a] consider the passive setting where training data is arbitrarily labeled. This leads to a sample complexity of $O(\frac{1}{\varepsilon^2})$ when the unfairness budget is ε [YR18, BL19, SCM20]. The following discussion concerns the active case, where which samples to label are judiciously chosen. Such selection improves the sample complexity to $O(\log \frac{1}{\varepsilon})$. This sample complexity reduction is a significant improvement over the existing methods.

3.2.2 Active Learning

Active learning is a popular and established subfield of machine learning that provides cost-reducing alternatives when labels on the training data are expensive to obtain [Set09, AKG$^+$14, H$^+$14]. It cautiously select certain data samples to label instead of labeling as many as possible arbitrarily. The existing selection process differ to cater to different needs. The uncertainty-based procedure labels data with uncertain model outputs, and the query-by-committee procedure labels data with disparate predictions from a committee of models. Active learning has shown success in achieving better accuracy more effectively than arbitrary labeling in the passive setting [TCM99, WLR$^+$03, Liu04, HJZL06, AZL06, ZH13]. To quantify its efficiency, active labeling achieves ε error in $O(\log \frac{1}{\varepsilon})$ while arbitrary labeling do that in $O(\frac{1}{\varepsilon})$ [Das05, Han07, BHV10].

Another notable trend in the literature is that active learning is usually designed for classification but very few are for regression [BRK07, SN09, CZZ13, YK10]. The common procedure for regression is greedy sampling for data labeling. It labels the data samples that are the most different from what's already labeled in terms of both features and label. In any case, these existing active learning algorithms improve accuracy for traditional learners while the following research, inspired by disagreement-based active learning [WLH19], improves individual fairness for AMF learners.

3.2.3 Fairness in Active Learning

Fairness in active learning is not researched until recently. They generally either use active labeling [AAT20, SDI20] or adaptive sampling [AAK$^+$20, SFGJ21]. [CL22a] in Chapter 4 is also active labeling, but it improves individual fairness for AMF learners while [AAT20, SDI20] aims at achieving group fairness for standard learners. Besides, [CL22a] is the first work that shows active learning can improve the sample complexity for individual fairness to $O(\log \frac{1}{\varepsilon})$. [CL22b] in Chapter 5 focus on the decoupled setting as brought up by [DIKL18]. It is the first to achieve increasing (instead of

slower decreasing) fairness during active sample selection while maintaining comparable accuracy.

Chapter 4

Active Approximately Metric-Fair (AMF) Learning

4.1 Active AMF Learning

4.1.1 Approximate Metric-Fairness (AMF)

A new version of approximate metric-fairness (AMF) is defined as follows and equivalence to the canonical version is proven. Next, the analogous passive AMF learner is shown and its generalization guarantee proven. The following discussion in [CL22a] concentrate on the regression problem.

Let X be an instance space equipped with a metric d and distribution D. Let H be a class of models defined on X. The original form of AMF [YR18] is defined based on almost Lipschitz continuity, as follows.

Definition 4.1.1. A model $h \in H$ is said to be (ε, β) approximately metric-fair with respect to d and D if

$$\Pr_{x,x' \sim D}\{|h(x) - h(x')| > d(x, x') + \beta\} \leq \varepsilon. \tag{4.1}$$

To facilitate algorithm design and analysis, [CL22a] propose the following new form of AMF based on uniform continuity.

Definition 4.1.2. A model $h \in H$ is said to be $(\alpha, \beta, \varepsilon)$ approximately metric-fair with respect to d and D if

$$\Pr_{x,x' \sim D}\{d(x, x') \leq \alpha,\ |h(x) - h(x')| > \beta\} \leq \varepsilon. \tag{4.2}$$

Intuitively, the new form models individual fairness by stating that if two individuals x and x' are similar (in a sense that $d(x, x') \leq \alpha$), then their predictions should be similar (in a sense that $|h(x) - h(x')| > \beta$) with high probability.

The following theorem suggests the two forms are equivalent, and is inspired by an interesting discovery that uniform continuity is *almost* Lipschitz [Van91].

Theorem 4.1.3. *Fix any $\alpha, \beta > 0$. Any model with a convex domain is (ε, β) approximately metric-fair with respect to d and D if it is $(\alpha, \beta, \varepsilon)$ approximately metric-fair with respect to metric $d' = \frac{\alpha}{2\beta} \cdot d$ and D, and only if it is $(\alpha, 3\beta, \varepsilon)$ approximately metric-fair with respect to d' and D.*

Proof. Let h be a model with a convex domain. Define two sets $\Psi_1(\beta) = \{(x, x') \mid |h(x) - h(x')| \leq d(x, x') + \beta\}$ and $\Psi_2(\alpha, \beta) = \{(x, x') \mid d'(x, x') \leq \alpha \Rightarrow |h(x) - h(x')| \leq \beta\}$, where '$\Rightarrow$' means 'imply'. We first prove

$$\Psi_2(\alpha, \beta) \subseteq \Psi_1(\beta) \subseteq \Psi_2(\alpha, 3\beta). \tag{4.3}$$

The left relation holds because, for any β, if there exists an α such that $d'(x, x') \leq \alpha$ implies $|h(x) - h(x')| \leq \beta$, then [Van91, Theorem 1] implies that $|h(x) - h(x')| \leq d(x, x') + \beta$, where $d(x, x') = \frac{2\beta}{\alpha} d'(x, x')$.

The right relation follows from the fact that, if $|h(x) - h(x')| \leq d(x, x') + \beta = \frac{2\beta}{\alpha} d'(x, x') + \beta$, then $d'(x, x') \leq \alpha$ implies $|h(x) - h(x')| \leq \frac{2\beta}{\alpha} \cdot \alpha + \beta = 3\beta$.

Then, by contrapositive, (4.3) implies

$$\tilde{\Psi}_2(\alpha, \beta) \supseteq \tilde{\Psi}_1(\beta) \supseteq \tilde{\Psi}_2(\alpha, 3\beta), \tag{4.4}$$

where $\tilde{\Psi}$ denotes the complement of Ψ. This further implies $\Pr\{\tilde{\Psi}_2(\alpha, \beta)\} \geq \Pr\{\tilde{\Psi}_1(\beta)\} \geq \Pr\{\tilde{\Psi}_2(\alpha, 3\beta)\}$, and the theorem follows by the two definitions of fairness. □

Theorem 4.1.3 shows one form of AMF converges to the other as β decreases, which establishes an equivalence between them. It also suggests one can achieve one form of AMF through the other. In the rest of this chapter, AMF learners will be designed and analyzed based on Definition 4.1.2. For conciseness, the subscripts in $\Pr_{x,x' \sim D}$ will be omitted whenever they are clear from the context.

Next, a passive AMF learner based on Definition 4.1.2 is described and its generalization guarantee is proven.

To facilitate discussion, define the fairness measure

$$\Delta_{\alpha,\beta}(h) = \Pr\{d(x, x') \leq \alpha, |h(x) - h(x')| > \beta\}. \tag{4.5}$$

Then h is said to be $(\alpha, \beta, \varepsilon)$-AMF if $\Delta_{\alpha,\beta}(h) \leq \varepsilon$.

Let S be a sample of $X \times X$ with cardinality m. An estimate of the probability $\Delta_{\alpha,\beta}(h)$ on sample S is

$$\Delta_{\alpha,\beta}(h; S) = \frac{1}{m} \sum_{(x,x') \in S} \mathbb{I}\{d(x, x') \leq \alpha, \; |h(x) - h(x')| > \beta\}, \tag{4.6}$$

where $\mathbb{I}$ is an indicator function.

It is natural for AMF learning to find a model h with small $\Delta_{\alpha,\beta}(h;S)$ and hope this could generalize to a small $\Delta_{\alpha,\beta}(h)$. This chapter focus on a realizable case where H contains perfect AMF models that satisfy $\Delta_{\alpha,\beta}(h)=0$. Based on this, the passive AMF learner is defined as follows.

Definition 4.1.4. Given a hypothesis class H, a loss function ℓ and a labeled training set $L=\{(x_1,y_n),\ldots,(x_n,y_n)\}$ where x_i is the i_{th} instance and y_i is its label, an AMF learner returns a model $h\in H$ by solving

$$\min_{h\in H}\ \frac{1}{n}\sum_{i=1}^{n}\ell(h(x_i),y_i),\quad \text{s.t. } \Delta_{\alpha,\beta}(h;S)=0, \tag{4.7}$$

where $S=\{(x_i,x_j)\}_{i,j=1,\ldots,n}$.

One can show the above AMF learner has a similar generalization guarantee as in [YR18] based on the following lemma. Let $\mathcal{R}_m(\cdot)$ denote the Rademacher complexity of some hypothesis class for sample size m.

Lemma 4.1.5. *Fix any $t,\beta>0$. Let $F: X\times X\to\mathbb{R}$ be a hypothesis class induced from H such that $\forall f\in F$, $f(x,x')=\tau_\beta^t(|h(x)-h(x')|)$ where $\tau_\beta^t(z)$ is a piecewise model outputting 1 if $z>\beta+\frac{1}{t}$, outputting 0 if $z\le\beta$ and $t(z-\beta)$ otherwise. Then $\mathcal{R}_m(F)\le 8t\cdot\mathcal{R}_m(H)$.*

Proof Sketch. Repeatedly apply the Rademacher complexity property of composite function with Lipschitz condition e.g. [BM02, Theorem 12] on τ_β^t and *abs*. See the supplementary material at the end of this chapter for details. □

Based on the above, one can prove the proposed AMF learner has generalization guarantee based on an assumption that instances are sampled i.i.d.. The results is as follows.

Theorem 4.1.6. *Fix any $\alpha,\beta,t>0$. Suppose $\mathcal{R}_m(H)\in O(1/\sqrt{m})$. Any model $h\in H$ returned by the AMF learner satisfies $\Delta_{\alpha,\beta+1/t}(h)\le\varepsilon$ with probability at least $1-\delta$ if $m\ge\frac{1}{\varepsilon^2}\left(16tc+\sqrt{\frac{1}{2}\log\frac{1}{\delta}}\right)$, where m is the number of $(x,x')\in S$ satisfying $d(x,x')\le\alpha$ and c is a constant inherited from $O(1/\sqrt{m})$.*

Proof Sketch. The main challenge in our analysis is an extra $d(x,x')\le\alpha$ term that cannot be directly removed using the Rademacher complexity property as in Lemma 4.1.5. To tackle this, I introduce $V=\{(x,x')\in S; d(x,x')\le\alpha\}$.

I will first transform the analysis of joint event $|h(a)-h(b)|>\beta$ and $d(a,b)\le$ to an analysis of single event $|h(a)-h(b)|>\beta$ by narrowing the

domain to V. Then, I derive a generalization bound for the single event by first relaxing its indicator function to the piecewise function defined in Lemma 4.1.5, then applying the standard generalization argument with $\mathcal{R}_m(F)$ e.g., [MRT18], and finally connecting $\mathcal{R}_m(F)$ to $\mathcal{R}_m(H)$ using Lemma 4.1.5. At the end, I transform the result for the single event back to a result for the joint event which completes the proof. See the supplementary material for details. □

Theorem 4.4.2 implies one can achieve $(\alpha, \beta, \varepsilon)$ AMF with $O(\frac{1}{\varepsilon^2})$ randomly labeled instances, which is consistent with the sample complexity in [YR18]. Constant c depends on the hypothesis class e.g., if H is the set of linear models with proper constraints, I can set c to the maximum norm of the instance [SSBD14]; if H is the set of kernel machines with proper constraints, I can set c to the product of kernel function bound and gram matrix trace [MRT18].

In the theorem, variable t is the slope of a Lipschitz function introduced to approximate the indicator function. Its impact on the error bound is interesting twofold. A smaller t leads to a weaker fairness guarantee, in a sense that $\Delta_{\alpha,\beta+1/t'} \leq \varepsilon$ implies $\Delta_{\alpha,\beta+1/t} \leq \varepsilon$ whenever $t' \leq t$. But it also leads to higher sample efficiency, in a sense that a smaller t implies smaller m suffices for the generalization guarantee.

Algorithm 1 Active AMF Learning

Input: an initial labeled training set L, an unlabeled set U, a hypothesis class H, number k.

1: **while** stopping criterion is not met **do**

2: Learn a model $h \in H$ based on sample L using the AMF learner in Definition 4.1.4.

3: Pick an i.i.d. sample of k instances $u \in U$ satisfying

$$\exists u' \in L,\ d(u, u') \leq \alpha,\ |h(u) - h(u')| > \beta. \tag{4.8}$$

4: Label the selected instances. Then add them to sample L, and remove them from sample U.

5: **end while**

Output: model h.

4.1.2 Sample Complexity of Active AMF Learning

In this section, [CL22a] propose an active AMF learner based on Definition 4.1.4 and derive its sample complexity.

The key idea is to label instances that are fairly close to their neighbors but receive fairly different predictions from some hypothesis. [CL22a]

characterize such instances using a set

$$\begin{aligned}\mathcal{C}_{\alpha,\beta}(H) = \{(x, x') \in X \times X; \exists h \in H, \\ d(x, x') \leq \alpha,\ |h(x) - h(x')| > \beta\}.\end{aligned} \tag{4.9}$$

Next, [CL22a] design a counter AMF coefficient, which will be used to derive the complexity.

Definition 4.1.7. The counter (α, β) AMF coefficient with respect to a hypothesis class H is

$$\xi_{\alpha,\beta} = \sup_{r>0} \frac{\Pr\{(x, x') \in \mathcal{C}_{\alpha,\beta}(\mathcal{B}_{\alpha,\beta}(r))\}}{r}, \tag{4.10}$$

where $\mathcal{B}_{\alpha,\beta}(r) = \{h \in H; \Delta_{\alpha,\beta}(h) \leq r\}$ is the set of hypotheses that are (α, β, r) AMF.

Intuitively, the coefficient measures the largest volume of instance pairs that do not contribute to the fairness achievable in a hypothesis class. One could expect it to be smaller if hypotheses are more fair. For conciseness, I will omit the subscripts in $\xi_{\alpha,\beta}$ whenever they are clear from the context.

The proposed active AMF learner is shown in Algorithm 1. In each round, it trains model h on the labeled set using the AMF learner, and then labels instances that are close to the training data but receive different predictions from h. It is clear that all labeled instances fall in $\mathcal{C}_{\alpha,\beta}(H)$. The fairness coefficients α, β are assumed preset by the problem, and one can stop labeling when a desired AMF degree is achieved.

Our following theorem shows that, under proper conditions, Algorithm 1 can return a model satisfying $(\alpha, \beta, \varepsilon)$ AMF through $O(\log \frac{1}{\varepsilon})$ labeling with high probability.

Theorem 4.1.8. *Fix any $\alpha, \beta > 0$. If the counter (α, β) AMF coefficient w.r.t. H is bounded, then with probability at least $1 - \delta$, any $h \in H$ returned by Algorithm 1 satisfies $\Delta_{\alpha,\beta}(h) \leq \varepsilon$ after $O(\log \frac{1}{\varepsilon})$ labeling.*

Proof Sketch. Let $V_q = \{h \in H; \Delta_{\alpha,\beta}(h; S_q) = 0\}$ be the set of 'perfect' AMF models at the end of Q rounds of labeling. The goal of our analysis is to show that, if one label $k = \frac{1}{4\xi^2}\left(32c/\beta + \sqrt{\frac{1}{2}\log\frac{1}{\delta'}}\right)$ instances in each round, then by the generalization bound in Theorem 4.4.2, there is

$$\Pr\{\mathcal{C}_{\alpha,\beta}(V_{q+1})\} \leq \frac{1}{2}\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}. \tag{4.11}$$

with high probability. This implies $Q = \log_2 \frac{1}{\varepsilon}$ rounds of labeling, which means $Q_k = O(\log \frac{1}{\varepsilon})$ total labeling suffices to achieve $\Pr\{\mathcal{C}_{\alpha,\beta}(V_{q+1})\} \leq \varepsilon$. Since $\Delta_{\alpha,\beta}(h) \leq \Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}$ for any $h \in V_q$ by definition, the theorem is proved.

Let & be logic 'AND' and define event

$$I_\alpha^\beta(x, x'; h) := d(x, x') \leq \alpha \ \& \ |h(x) - h(x')| > \beta. \tag{4.12}$$

A key to prove (4.11) is to split the domain of $\Delta_{\alpha,\beta}(h) = \Pr\{I_\alpha^\beta(x, x'; h)\}$ for any $h \in V_{q+1}$ into $(x, x') \in \mathcal{C}_{\alpha,\beta}(V_q)$ and $(x, x') \notin \mathcal{C}_{\alpha,\beta}(V_q)$. Probability on the second subdomain is zero, and probability on the first subdomain can be bounded using Theorem 4.4.2 conditioned on the fact that all labeled instances fall in $\mathcal{C}_{\alpha,\beta}(V_q)$. That bound is smaller than $\frac{1}{2\xi}$ by our choice of k and the definition of ξ, therefore implying $V_{q+1} \subseteq \mathcal{B}\left(\frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2\xi}\right)$ and thus $\Pr\{\mathcal{C}_{\alpha,\beta}(V_{q+1})\} \leq \Pr\left\{\mathcal{C}_{\alpha,\beta}\left(\mathcal{B}_{\alpha,\beta}\left(\frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2\xi}\right)\right)\right\} \leq \xi \cdot \frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2\xi} = \frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2}$, where the second inequality is by definition. This proves (4.11) and thus the theorem. □

The proof of Theorem 4.1.8 also illuminates the key for Algorithm 1 to reduce labeled instances is in Step 3, where one label u if $(u, u') \in \mathcal{C}_{\alpha,\beta}(V_q)$ because only such pair can be used to further rule out hypotheses in V_q and shrink $\mathcal{C}_{\alpha,\beta}(V_q)$, which guarantees the shrinkage of $\Delta_{\alpha,\beta}(h)$.

An implicit assumption of the derived sample complexity is that, the unlabeled set contains at least one instance satisfying (4.8) per epoch until convergence. This is similar to the analysis of disagreement-based active learning [H+14], which assumes at least one unlabeled instance is disagreed by the committee models per epoch. From a practical perspective, when no valid instance is found, one could train another model or randomly label one instance and proceed to the next epoch.

The time complexity for Algorithm 1 to find an instance satisfying (4.8). In a centralized computing environment, the complexity is $O(|U||L|)$, where $|U|$ is the size of unlabeled set and $|L|$ is the size of labeled set. Typically $|L| \ll |U|$. This is higher than the complexity of uncertainty-based strategy which is typically $O(|U|)$, but more comparable to the complexity of query-by-committee which is typically $O(|U|t)$ for t committee models. In a distributed computing environment, the complexity can be reduced to $O(|U|)$ if the evaluations of an instance $u \in U$ with all $u' \in L$ can be parallelized. Nonetheless, how to make selection more efficient remains an open challenge.

4.1.3 The Counter AMF Coefficient

An important factor in our analysis is the counter AMF coefficient. [CL22a] give an example on how to calcualte it.

Example 4.1.9. Fix $\alpha, \beta, B > 0$. Let $h_w(x) = w \cdot x$ be a 1-dimensional linear hypothesis defined on $[-B, B]$, and define $H = \{h_w; w \geq 0\}$. Assume

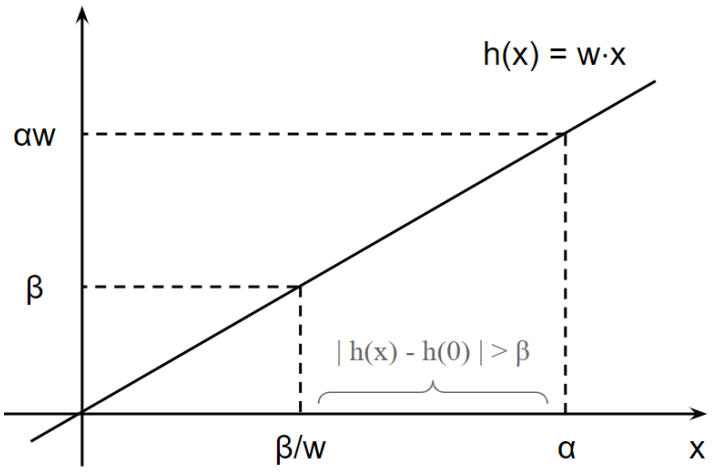


Figure 4.1: Visualization of $\Pr_x(I_\alpha^\beta(x, 0; h))$.

instances are uniformly distributed on $[-B, B]$. If $B > \alpha$, then

$$\Delta_{\alpha,\beta}(h) = \begin{cases} 0, & \text{if } w < \frac{\beta}{\alpha} \\ 1 - \frac{\beta}{\alpha w} + \frac{\beta}{Bw} \ln \frac{\beta}{\alpha w}, & \text{if } w \geq \frac{\beta}{\alpha} \end{cases}, \tag{4.13}$$

and the counter (α, β) AMF coefficient w.r.t. to H is 1.

Proof. The roadmap of this proof is as follows. First derive $\Delta_{\alpha,\beta}(h_w)$ through case study and show it is non-decreasing with respect to w. Based on this, argue the probability in (4.10) is equivalent to

$$\mathbb{P}_* := \Pr\{|h_*(x) - h_*(x')| > \beta, d(x, x') \leq \alpha\}, \tag{4.14}$$

where h_* is the model satisfying $\Delta_{\alpha,\beta}(h_*) = r$, hence $\xi = \sup_{r>0} \mathbb{P}_*/r = \sup_{r>0} r/r = 1$.

Now the detailed proof is shown. For conciseness, I will write h for h_w but with the mind that each h is associated with a w. Also, recall the event notation $I_\alpha^\beta(x, x'; h) := d(x, x') \leq \alpha \;\&\; |h(x) - h(x')| > \beta$.

Step 1: Characterize $\Delta_{\alpha,\beta}(h)$ for any $h \in H$.

Fix any h. Consider two cases.

(i) If $\alpha w < \beta$, simple geometric analysis shows that event $I_\alpha^\beta(x, x'; h)$ is always false so $\Pr_{x,x'}\{I_\alpha^\beta(x, x'; h)\} = 0$.

(ii) If $\alpha w \geq \beta$ (which implies $w \neq 0$), then $\alpha \geq \beta/w$. In this case, we

can properly partition the domain and have

$$\begin{aligned}
&\Pr_{x,x'}\{I_\alpha^\beta(x,x';h)\} \\
&= \mathbb{E}_{x'\in[-B,B]}\left[\Pr_x\{I_\alpha^\beta(x,x';h)\}\right] \\
&= 2\,\mathbb{E}_{x'\in[0,B]}\left[\Pr_x\{I_\alpha^\beta(x,x';h)\}\right] \\
&= 2\int_{x'\in[0,B-\alpha]} \Pr_x\{I_\alpha^\beta(x,x';h)\}\cdot p(x') \\
&\quad + 2\int_{x'\in(B-\alpha,B-\frac{\beta}{w}]} \Pr_x\{I_\alpha^\beta(x,x';h)\}\cdot p(x') \\
&\quad + 2\int_{x'\in(B-\frac{\beta}{w},B]} \Pr_x\{I_\alpha^\beta(x,x';h)\}\cdot p(x'),
\end{aligned} \tag{4.15}$$

where $\Pr_x\{I_\alpha^\beta(x,x';h)\}$ is the probability defined for x with a fixed x'. In (4.15), the first equality is by definition, and the second equality is by the observation that $\Pr_x\{I_\alpha^\beta(x,x';h)\}$ is symmetric on $[-B,B]$ (which will become more clear in later analysis). Note that $p(x')=\frac{1}{2B}$.

Now each integral is studied separately.

(ii.a) If $x'\in[0,B-\alpha]$, we can show

$$\Pr_x\{I_\alpha^\beta(x,x';h)\} = 1-\frac{\beta}{w\alpha}. \tag{4.16}$$

To verify this, let us first fix $x'=0$ and identify the set of x in $[-\alpha,\alpha]$ that makes event $I_\alpha^\beta(x,0;h)$ true. This case is illustrated in Figure 4.1. We see all targeted x fall in $[\beta/w,\alpha]$ and (by symmetry) in $[-\alpha,-\beta/w]$. This implies $\Pr_x\{I_\alpha^\beta(x,0;h)\}=\frac{2\cdot(\alpha-\beta/w)}{2\cdot\alpha}=1-\frac{\beta}{\alpha w}$. Since h is linear, the above result applies to all $x'\in[0,B-\alpha]$, which implies (4.16) and thus the first integral equals to $(1-\frac{\beta}{w\alpha})(1-\frac{\alpha}{B})$. Note it is non-negative since $\alpha w\geq\beta$ and $B>\alpha$.

(ii.b) If $x'\in(B-\alpha,B-\frac{\beta}{w}]$, we can show

$$\Pr_x\{I_\alpha^\beta(x,x';h)\} = 1-\frac{\beta}{w(B-x')}. \tag{4.17}$$

We can verify this in a similar way as in (ii.a), with additional shift of the origin to x' and constraint $x\leq B$. Then, geometric analysis suggests all targeted x fall in $[x'+\frac{\beta}{w},B]$ (shorter than interval $[\beta/w,\alpha]$ in Figure 4.1) and thus $\Pr_x\{I_\alpha^\beta(x,x';h)\}=\frac{2(B-x'-\beta/w)}{2(B-x')}$. This implies the second integral is $\frac{1}{B}(\alpha-\frac{\beta}{w}(1-\ln\frac{\beta}{w\alpha}))$. Note it is non-negative as (4.17) is non-negative by the domain of x'.

(ii.c) If $x'\in(B-\frac{\beta}{w},B]$, it is easy to see no (x',x) makes event $I_\alpha^\beta(x,x';h)$ true so $\Pr_x\{I_\alpha^\beta(x,x';h)\}=0$. Then the third integral is zero.

Plugging the integrals of (ii.a), (ii.b) and (ii.c) back to (4.15), and combining results of cases (i) and (ii) gives (4.13).

Step 2: Show $\Delta_{\alpha,\beta}(h)$ is non-decreasing w.r.t. w.

All one need to show is $\Delta_{\alpha,\beta}(h)$ is non-negative and non-decreasing when $w \geq \frac{\beta}{\alpha}$. The first property is guaranteed since all integrals in (4.15) are non-negative. To see the second property, take derivative $\frac{\partial \Delta_{\alpha,\beta}(h)}{\partial w} = \frac{\beta\,(B+\alpha(\ln\frac{\alpha w}{\beta}-1))}{w^2\,\alpha\,B}$. Since $w \geq \frac{\beta}{\alpha}$ and $B > \alpha$, one can easily show the derivative is bigger than zero and hence $\Delta_{\alpha,\beta}(h)$ is non-decreasing.

Step 3: Equivalent Probability.

Let $h_* = w_* x$ be the model satisfying $\Delta_{\alpha,\beta}(h_*) = r$. It is not hard to show it exists for every $r \in [0, 1)$ based on (4.13). Then, results of Step 1 and Step 2 suggest $\mathcal{B}_{\alpha,\beta}(r)$ is the set of linear models satisfying $w \leq w_*$, which implies

$$\Pr\{\mathcal{C}_{\alpha,\beta}(\mathcal{B}_{\alpha,\beta}(r))\} = \Pr\{I_\alpha^\beta(x, x'; h_*)\}. \tag{4.18}$$

To verify this, one first show every $(x, x') \in \mathcal{C}_{\alpha,\beta}(\mathcal{B}_{\alpha,\beta}(r))$ makes event $I_\alpha^\beta(x, x'; h_*)$ true. This is true because, for any x, x' with $d(x, x') \leq \alpha$, if there exists an $w \leq w_*$ such that $|wx - wx'| > \beta$, then $|w_* x - w_* x'| \geq |wx - wx'| > \beta$. One can then show every (x, x') that makes event $I_\alpha^\beta(x, x'; h_*)$ true is also in $\mathcal{C}_{\alpha,\beta}(\mathcal{B}_{\alpha,\beta}(r))$. This is true since h_* exists.

The equivalence implies $\xi = \sup_{r>0} \frac{\Pr\{I_\alpha^\beta(x,x';h_*)\}}{r} = \sup_{r>0} \frac{r}{r} = 1$. The proof is completed. □

4.2 Experiments

4.2.1 Implementation Issues

In this section, three implementation issues are discussed.

The first issue is related to the AMF Learner in Definition 4.1.4. Directly solving (4.7) is not easy since $\Delta_{\alpha,\beta}(h)$ is non-convex. [CL22a] propose to approximate the solution by solving

$$\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i) + \lambda\, \tilde{\Delta}_{\alpha,\beta}(h; S), \tag{4.19}$$

instead, where λ is a regularization coefficient and

$$\tilde{\Delta}_{\alpha,\beta}(h; S) = \frac{1}{n^2 \beta^2} \sum_{i,j=1}^{n} M_{ij} \cdot |h(x_i) - h(x_j)|^2, \tag{4.20}$$

with M being an n-by-n matrix whose entries are defined as $M_{ij} = \mathbb{I}\{d(x_i, d_j) \leq \alpha\}$. Such approximation can be justified by the following relation, which implies that minimizing $\tilde{\Delta}_{\alpha,\beta}(h; S)$ also minimizes $\Delta_{\alpha,\beta}(h; S)$.

Lemma 4.2.1. *Fix any $\alpha, \beta > 0$. Then $\Delta_{\alpha,\beta}(h; S) \leq \tilde{\Delta}_{\alpha,\beta}(h; S)$ for any $h \in S$ and sample S.*

In practice, the approximate learner (4.19) may not always return a model with zero bias on training data. In this case, the proposed algorithm remains applicable and sample-efficient on fairness. There are two possible theoretical explanations on the maintained efficiency. First, if the bias is sufficiently small e.g., $\Delta_{\alpha,\beta}(h; S) \in O(\varepsilon)$, then the passive bound in Theorem 4.4.2 can be extended to $\Delta_{\alpha,\beta}(h) \in O(\varepsilon)$. Plugging this back to Theorem 4.1.8, one can obtain a similar complexity with an additional constant factor. Second, one may borrow ideas from agnostic active learning e.g., [DHM07, BBL09] and develop a new complexity for the non-realizable case (i.e., when h has zero bias). These possible extensions are left for future study.

The second implementation issue is related to the base model. [CL22a] propose to implement a linear model and a kernel regression model approximated by Random Fourier Feature [RR07] called 'rff model'.

For the linear model, if instances $x_1, \ldots, x_n \in \mathbb{R}^p$, one can show $\tilde{\Delta}_{\alpha,\beta}(h; S) = \frac{2}{n^2\beta^2} \cdot h^T[x](D - M)[x]^T h$, where $[x]$ is an n-by-p matrix with the i_{th} row being x_i^T. Further, if squared loss is used, then solution to (4.19) is

$$h = ([x](I - \frac{2\lambda}{n\beta^2}(D - M))[x]^T)^{-1}([x][y]), \tag{4.21}$$

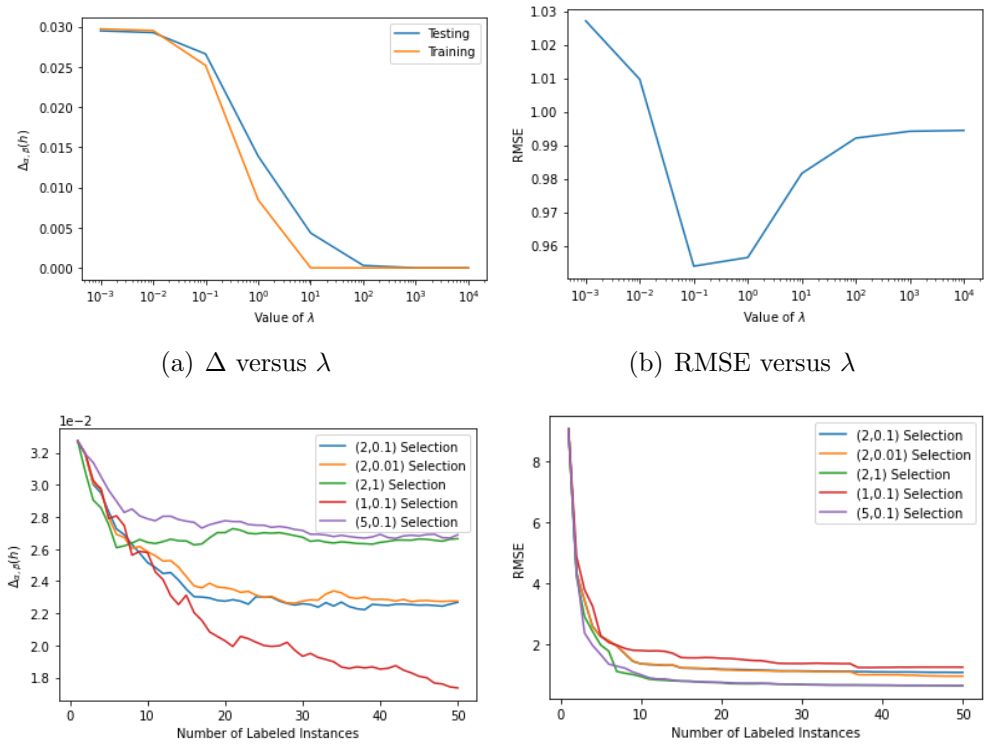
where $[y] \in \mathbb{R}^n$ is a vector with the i_{th} entry being y_i and D is an n-by-n diagonal matrix with $D_{ii} = \sum_{j=1}^{n} M_{ij}$.

For the rff model, one can first calculate random features [RR07] and then train a linear model based on them using the AMF learner. Note random features are only used to approximate the prediction model, and $d(x, x')$ is still measured using the original features.

The last issue is related to active learning. Given a labeled training set L and an unlabeled set U, the proposed active AMF learner labels a candidate instance u if there exists $u' \in L$ satisfying $d(u, u') \leq \alpha$ and $|h(u) - h(u')| > \beta$. In principle, one can also pair u with instances in U, as long as the labeled instances fall in $\mathcal{C}_{\alpha,\beta}(V_q)$. In practice, pairing u with instances in L is often more efficient (since the label set is often way smaller than the unlabeled set), and leads to slightly better performance as one observe in experiments.

4.2.2 Empirical Results

[CL22a] experiment on three real-world data sets. The Insurance data set[data] has individual medical costs billed by health insurance company, and the task is to predict the cost based other attributes. The Life data



(a) Δ versus λ

(b) RMSE versus λ

(c) Δ vs (α, β) Selection

(d) RMSE vs (α, β) Selection

Figure 4.2: Results of Sensitivity Analysis

set[datb] has the life expectancy in different countries, and the task is to predict the expectancy. [CL22a] also use a data set collected from public resources. It contains the COVID death rates of 3142 counties in United States and the task is to predict the rate based on other attributes including population density, obesity rate, smoking rate, diabetes rate, elderly population and vaccine rate. To learn more data sets used to evaluate algorithmic fairness, interested readers may look into [LQRI+22].

[CL22a] encode categorical features by dummy variables, address missing data using mean imputation and standardize all features. For higher numerical stability, re-scale the labels are re-scaled: on the Insurance data set, divide the medical cost which varies from 4k to 40k by 10k; on the Life data set, divide the life expectancy which varies from 40 to 90 by 100; on the COVID data set, multiply the death rate which varies from 0 to 0.01 by 100.

Each data set is arbitrarily split into an initial training set (assumed labeled), an unlabeled set (for query) and a testing set. Size of the initial training set is chosen as follows: for the linear base model, it is the feature number on the Insurance and Life data sets, and twice that number on the COVID data set; for the rff base model, it is half of the random feature

number. Size of the testing data is 25% of the total data size. The remaining data are treated as unlabeled.

On a data set, [CL22a] run an active learner for 20 random trials and report the average model performance on the testing sets. Model bias is measured by $\Delta_{\alpha,\beta}(h; S_n)$ defined in (4.6), with (α, β) set to (2, 0.1) on Insurance, (10, 0.2) on Life and (1.5, 0.001) on COVID. [CL22a] also experiment with other fairness coefficients and observe similar comparative performance. Model error is measured by the root-mean-squared-error.

[CL22a] evaluate the proposed active labeling strategy on the linear base model and rff base model respectively, and compare its performance with the following three strategies.

– *Random*: It randomly selects instances to label.

– *Query-by-Committee (QBC)*: It labels instances which receive the largest prediction variance from a committee of models. Following [BRK07], [CL22a] construct a committee of five models and train each one using a bootstrap sample of the training data, with sample size equals to the training set size divided by the committee size.

– *Uncertainty*: It labels instances which are most different from the training data in both feature space and label space [WLH19]. To our knowledge, this is a state-of-the-art active labeling method for regression model.

– *Cluster*: It is a clustering based baseline method that relies on the distance between instances. It first identifies the top m uncertain instances in the candidate pool using the above method, then runs k-means clustering to identify their k centers, and finally labels the identified instances.

For the metric-fair learner, its regularization coefficient λ is chosen to strike a good balance between fairness and accuracy. For the linear base model, λ is set to 1 on Insurance and Life and 0.1 on COVID; for the rff base model, λ is set to 1 on Insurance, 5 on Life and 0.5 on COVID.

For the rff base model, the random features that approximate Gaussian kernel [RR07] are generated. The random feature number is set to 100 on Insurance, 400 on Life and 200 on COVID. The gamma coefficient is set to 1e-4 on Insurance, 1e-9 on Life and 1e-2 on COVID. In practice, one can observe these configurations lead to good and stable performance of active metric-fair learning. For the clustering based baseline method, set $m = 10$ and $k = 3$ as they give consistently good performance (except k is set to 10 for linear model on the Life dataset).

Results of the experimented strategies on both base models across three data sets are shown in Figure 4.3.

In Figure 4.3 (a-f), one can see the proposed active AMF learner reduces model bias more efficiently than other learners, which empirically verifies its efficient sample complexity. One might notice it achieves almost zero bias in all cases, supporting our assumption on the realizable case. (And

note this is not achieved at the cost of significantly deteriorating accuracy, as explained in the next paragraph.) There seems no consistent pattern on the efficiency of other learners. One may notice QBC and uncertainty are often less efficient than random, implying the importance of (efficiently) achieving individual fairness by design, as presented in this study.

In Figure 4.3 (g-l), it is not surprising to see that uncertainty based labeling reduces error faster than other strategies. Comparatively, the proposed active AMF learner manages to achieve a comparable reduction rate, suggesting its has an efficient fairness-accuracy trade-off.

Sensitivity analysis on the proposed strategy is also performed. Results are presented in Figure 4.2. Figures 4.2 (a-b) show the performance versus regularization coefficient λ. One can see both training and testing δ decrease as λ increases. This suggests the metric-fair learner can effectively reduce bias and the reduction is generalizable, which supports Theorem 4.4.2. One can also see model error first decreases and then increases, exhibiting an overfitting phenomenon.

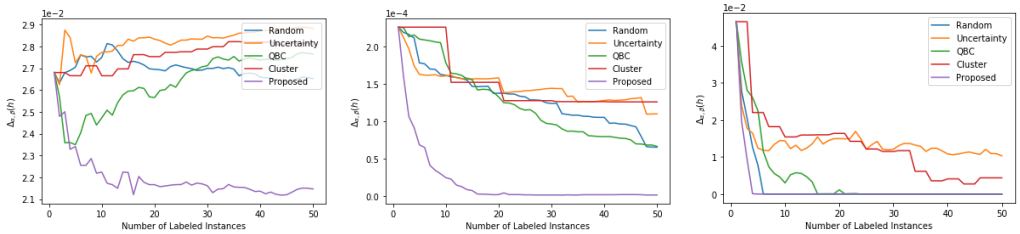
Figures 4.2 (c-d) show the performance versus different choices of (α, β) when selecting instances in Step 3 of Algorithm 1. (But all δ's are evaluated based on the same (α, β) for fair comparison.) One can see using smaller α to select instances leads to faster convergence of δ but more slowly convergence of RMSE. There seems no clear pattern on the impact of β. Overall, it is shown that one can balance fairness and accuracy of the proposed strategy through adjusting α.

4.3 Conclusion

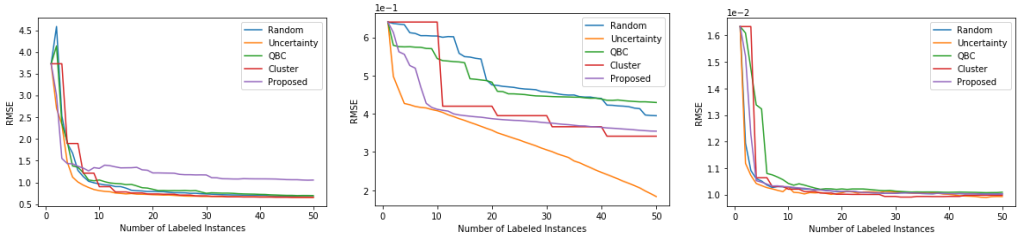
[CL22a] propose the first active approximate metric-fair (AMF) learner and prove it can achieve an ε bias budget by labeling only $O(\log \frac{1}{\varepsilon})$ instances. To my knowledge, this result is a first and substantial improvement over the existing $O(\frac{1}{\varepsilon^2})$ sample complexity for achieving individual fairness by the passive learners. Through extensive experiments across three public data sets, [CL22a] show the proposed active AMF learner improves fairness of two regression models more efficiently than its passive counterpart as well as state-of-the-art active learners, while being able to maintain comparable accuracy. Another contribution of this study is to present a provably equivalent form of AMF based on uniform continuity instead of the existing almost Lipschitz.

4.4 Supplementary Material

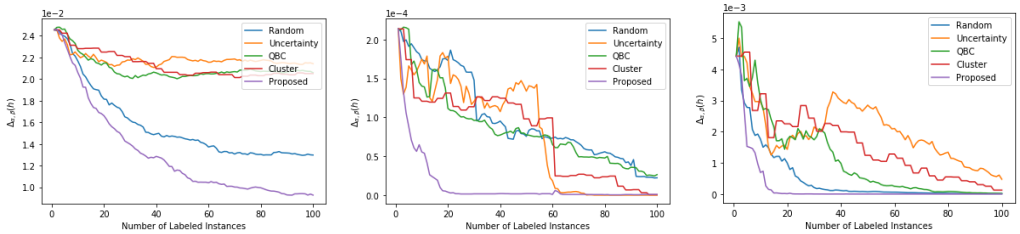
Lemma 4.4.1 (Lemma 3.5)**.** *Fix any $t, \beta > 0$. Let $F : X \times X \to \mathbb{R}$ be a hypothesis class induced from H such that $\forall f \in F$, $f(x, x') = \tau_\beta^t(|h(x) -$*



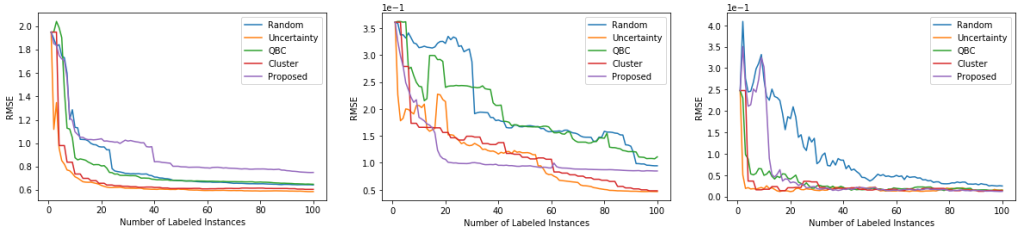
(a) Linear Insurance (b) Linear Life (c) Linear COVID



(d) Linear Insurance (e) Linear Life (f) Linear COVID



(g) RFF Insurance (h) RFF Life (i) RFF COVID



(j) RFF Insurance (k) RFF Life (l) RFF COVID

Figure 4.3: Bias and RMSE of Different Active Labeling Strategies for Linear and RFF model on Three Data Sets

$h(x')|$) *where* $\tau_\beta^t(z)$ *is a piecewise model outputting* 1 *if* $z > \beta + \frac{1}{t}$, *outputting* 0 *if* $z \leq \beta$ *and* $t(z - \beta)$ *otherwise. Then* $\mathcal{R}_m(F) \leq 8t \cdot \mathcal{R}_m(H)$.

Proof. Let $G : X \times X \to \mathbb{R}$ be the set of functions induced from h and defined as $\forall g \in G$, $g(a, b) = h(a) - h(b)$. Let *abs* be the absolute function. Then $f(a, b) = \tau_\beta^t \circ abs \circ g(a, b)$ and one can write, accordingly,

$$F = \tau_\beta^t \circ abs \circ G. \tag{4.22}$$

One first show $\mathcal{R}_m(F) \leq \mathcal{R}_m(G)$. This is true because

$$\mathcal{R}_m(F) = \mathcal{R}_m(\tau_\beta^t \circ abs \circ G) \leq 2t \cdot \mathcal{R}_m(abs \circ G) \leq 4t \cdot \mathcal{R}_m(G), \tag{4.23}$$

where both inequalities are by the property of Rademacher complexity for composite function with one component being Lipschitz continuous e.g., [BM02, Theorem 12] and the facts that τ_β^t and abs are both Lipschitz with constants t and 1 respectively.

One then show $\mathcal{R}_m(G) \leq 2 \cdot \mathcal{R}_m(H)$. This is true because

$$\begin{aligned}
\mathcal{R}_m(G) &= \mathbb{E}_{\{(a_i,b_i)\}}\mathbb{E}_\sigma \sup_{g\in G} \frac{1}{m}\sum_{i=1}^m \sigma_i g(a_i,b_i) \\
&= \mathbb{E}_{\{(a_i,b_i)\}}\mathbb{E}_\sigma \sup_{g\in G} \frac{1}{m}\sum_{i=1}^m \sigma_i [h(a_i) - h(b_i)] \\
&\leq \mathbb{E}_{\{(a_i,b_i)\}}\mathbb{E}_\sigma \sup_{g\in G} \frac{1}{m}\sum_{i=1}^m \sigma_i h(a_i) + \mathbb{E}_{\{(a_i,b_i)\}}\mathbb{E}_\sigma \sup_{g\in G} \frac{1}{m}\sum_{i=1}^m \sigma_i h(b_i) \\
&= 2 \cdot \mathbb{E}_{\{(a_i,b_i)\}}\mathbb{E}_\sigma \sup_{g\in G} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i) \\
&= 2 \cdot \mathcal{R}_m(H),
\end{aligned} \tag{4.24}$$

where the third equality is based on the fact that σ_i is uniform in {-1, 1} so the expectation with respect to σ_i is the same as the expectation with respect to $-\sigma_i$.

Combining (4.23) and (4.24) proves the lemma. □

Theorem 4.4.2 (Theorem 3.6)**.** *Fix any $\alpha, \beta, t > 0$. Suppose $\mathcal{R}_m(H) \in O(1/\sqrt{m})$. Any model $h \in H$ returned by the AMF learner satisfies $\Delta_{\alpha,\beta+1/t}(h) \leq \varepsilon$ with probability at least $1-\delta$ if $m \geq \frac{1}{\varepsilon^2}\left(16tc + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\right)$, where m is the number of $(x, x') \in S$ satisfying $d(x, x') \leq \alpha$ and c is a constant inherited from $O(1/\sqrt{m})$.*

Proof. To facilitate discussion, define two functions

$$\tau_\beta(z) = \begin{cases} 1, & \text{if } z > \beta \\ 0, & \text{if } z \leq \beta \end{cases}, \tag{4.25}$$

and

$$\tau_\beta^t(z) = \begin{cases} 1, & \text{if } z > \beta + \frac{1}{t} \\ t(z-\beta), & \text{if } \beta < z \leq \beta + \frac{1}{t} \\ 0, & \text{if } z \leq \beta \end{cases}. \tag{4.26}$$

By definition, one have

$$\tau_{\beta+\frac{1}{t}}(z) \leq \tau_\beta^t(z) \leq \tau_\beta(z). \tag{4.27}$$

Recall $S = \{(x_i, x_j)\}_{i,j=1,\ldots,n}$. Let S_α be a subset of S defined as

$$S_\alpha = \{(a,b) \in S \mid d(a,b) \leq \alpha\}. \tag{4.28}$$

Suppose the size of S_α is m. Then,

$$\begin{aligned}
\Delta_{\alpha,\beta}(h;S) &= \frac{1}{n^2}\sum_{i,,j=1}^{n} \mathbb{I}\{|h(x_i)-h(x_j)| > \beta,\ d(x_i,x_j) \leq \alpha\} \\
&= \frac{m}{n^2}\cdot\frac{1}{m}\sum_{(a,b)\in S_\alpha} \mathbb{I}\{|h(a)-h(b)| > \beta\} \\
&= \frac{m}{n^2}\cdot\frac{1}{m}\sum_{(a,b)\in S} \tau_\beta(|h(a)-h(b)|).
\end{aligned} \tag{4.29}$$

Recall $F : X \times X \to \mathbb{R}$ is the set of functions induced from τ_β^t and defined as $\forall f \in F,\ f(a,b) = \tau_\beta^t(|h(a)-h(b)|)$. One have that, with probability at least $1-\delta$,

$$\begin{aligned}
\frac{1}{m}\sum_{(a,b)\in S} \tau_\beta(|h(a)-h(b)|) &\geq \frac{1}{m}\sum_{(a,b)\in S} \tau_\beta^t(|h(a)-h(b)|) \\
&\geq \mathbb{E}[\tau_\beta^t(|h(a)-h(b)|) \mid d(a,b)\leq\alpha] - 2\mathcal{R}_m(F) - \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \\
&\geq \mathbb{E}[\tau_{\beta+\frac{1}{t}}(|h(a)-h(b)|) \mid d(a,b)\leq\alpha] - 16t\mathcal{R}_m(H) - \sqrt{\frac{\log\frac{1}{\delta}}{2m}} \\
&\geq \mathbb{E}[\tau_{\beta+\frac{1}{t}}(|h(a)-h(b)|) \mid d(a,b)\leq\alpha] - \frac{1}{\sqrt{m}}\left(16tc + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\right).
\end{aligned} \tag{4.30}$$

where for some constant c. In (4.30), the first inequality is by (4.27); the second one is by standard generalization bound[1] with Rademacher complexity e.g. [MRT18, Theorem 3.3] conditioned on $d(a,b) \leq \alpha$; the third one is by (4.27) and Lemma 4.1.5; and the last one holds since $\mathcal{R}_m \in O(1/\sqrt{m})$. Note the expectation of $(a,b) \in S_\alpha$ in $\mathcal{R}_m \in O(1/\sqrt{m})$ is

[1]Here one can follow [YR18] and treat S_α as an i.i.d. sample. If it is not, one can either add an additional constraint that no two pairs in S_α share the same instance so it can be viewed as an i.i.d. sample, or apply a generalization error bound on non-i.i.d. sample e.g. [MR08]. In either case, the order of our result remains the same.

also conditioned on $d(a,b) \leq \alpha$, and one always assume $\mathcal{R}_m \in O(1/\sqrt{m})$ w.r.t. any data proper distribution.

Combining (4.29) and (4.30), one can see $\Delta_{\alpha,\beta}(h;S) = 0$ implies

$$\mathbb{E}[\tau_{\beta+\frac{1}{t}}(|h(a)-h(b)|) \mid d(a,b) \leq \alpha] \leq \frac{1}{m}\left(16tc + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\right). \tag{4.31}$$

Further, one can show

$$\Delta_{\alpha,\beta+\frac{1}{t}}(h) \leq \mathbb{E}[\tau_{\beta+\frac{1}{t}}(|h(a)-h(b)|) \mid d(a,b) \leq \alpha], \tag{4.32}$$

because

$$\begin{aligned}
\Delta_{\alpha,\beta+\frac{1}{t}}(h) &= \int_{(a,b)\in X\times X} \mathbb{I}\{|h(a)-h(b)| > \beta + 1/t\} \cdot \mathbb{I}\{d(a,b) \leq \alpha\} \cdot p(a,b) \\
&\leq \int_{(a,b)\in X\times X} \mathbb{I}\{|h(a)-h(b)| > \beta + 1/t\} \cdot p(a,b) \\
&\leq \int_{(a,b)\in X\times X} \mathbb{I}\{|h(a)-h(b)| > \beta + 1/t\} \cdot p(a,b \mid d(a,b) \leq \alpha) \\
&= \mathbb{E}[\tau_{\beta+\frac{1}{t}}(|h(a)-h(b)|) \mid d(a,b) \leq \alpha].
\end{aligned} \tag{4.33}$$

Combining (4.31) and (4.32), and upper bounding the RHS of (4.31) by ε implies that $\Delta_{\alpha,\beta+\frac{1}{t}}(h) \leq \varepsilon$ whenever

$$m \geq \frac{1}{\varepsilon^2}\left(16tc + \sqrt{\frac{1}{2}\log\frac{1}{\delta}}\right). \tag{4.34}$$

The theorem is proved. □

Theorem 4.4.3 (Theorem 4.2)**.** *Fix any $\alpha, \beta > 0$. Suppose $\mathcal{R}_m(H) \in O(1/\sqrt{m})$ and the counter (α,β) AMF coefficient w.r.t. H is bounded. Then, with probability at least $1-\delta$, any $h \in H$ returned by Algorithm 1 satisfies $\Delta_{\alpha,\beta}(h) \leq \varepsilon$ after $O(\log\frac{1}{\varepsilon})$ labeling.*

Proof. Suppose one have performed q rounds of labeling. Let L_q be the updated training set and S_q be the associated set of instance pairs in Definition 4.1.4. Define

$$V_q = \{h \in H; \Delta_{\alpha,\beta}(h;S_q) = 0\}. \tag{4.35}$$

Consider labeling m instances in round $q+1$. First, note that all labeled instances fall in $\mathcal{C}_{\alpha,\beta}(V_q)$ and thus will add to S_q at least m pairs of (x,x') satisfying $d(x,x') \leq \alpha$. Then, by Theorem 4.4.2 and setting $t = 1/\beta$, if

$m \geq \frac{1}{4\xi^2}\left(32c/\beta + \sqrt{\frac{1}{2}\log\frac{1}{\delta'}}\right)$, with probability at least $1-\delta'$, any $h \in V_{q+1}$ satisfies

$$\Delta_{\alpha,\beta}(h) \leq 1/(2\xi). \tag{4.36}$$

Let & be logic 'AND' and define event

$$I_\alpha^\beta(x, x'; h) := d(x, x') \leq \alpha \;\&\; |h(x) - h(x')| > \beta. \tag{4.37}$$

Then, with probability at least $1-\delta'$, any $h \in V_{q+1}$ satisfies

$$\begin{aligned}
\Pr\{I_\alpha^\beta(x, x'; h)\} &= \Pr\{I_\alpha^\beta(x, x'; h) \,\&\, (x, x') \in \mathcal{C}_{\alpha,\beta}(V_q)\} \\
&\quad + \Pr\{I_\alpha^\beta(x, x'; h) \,\&\, (x, x') \notin \mathcal{C}_{\alpha,\beta}(V_q)\} \\
&= \Pr\{I_\alpha^\beta(x, x'; h) \,\&\, (x, x') \in \mathcal{C}_{\alpha,\beta}(V_q)\} \\
&= \Pr\{I_\alpha^\beta(x, x'; h) \mid (x, x') \in \mathcal{C}_{\alpha,\beta}(V_q)\} \cdot \Pr\{(x, x') \in \mathcal{C}_{\alpha,\beta}(V_q)\} \\
&\leq \frac{\Pr\{(x, x') \in \mathcal{C}_{\alpha,\beta}(V_q)\}}{2\xi},
\end{aligned} \tag{4.38}$$

where the second equality is by the fact that $\Pr\{I_\alpha^\beta(x, x'; h) \;\&\; (x, x') \notin \mathcal{C}_{\alpha,\beta}(V_q)\} \leq \Pr\{I_\alpha^\beta(x, x'; h) \;\&\; (x, x') \notin \mathcal{C}_{\alpha,\beta}(V_{q+1})\} = 0$, and the inequality is by (4.36) conditioned on an additional fact that all labeled instances fall in $\mathcal{C}_{\alpha,\beta}(V_{q+1})$. For conciseness, one will write $\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}$ for $\Pr\{(x, x') \in \mathcal{C}_{\alpha,\beta}(V_q)\}$.

Result in (4.38) implies $V_{q+1} \subseteq \mathcal{B}\left(\frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2\xi}\right)$ and

$$\begin{aligned}
\Pr\{\mathcal{C}_{\alpha,\beta}(V_{q+1})\} &\leq \Pr\left\{\mathcal{C}_{\alpha,\beta}\left(\mathcal{B}_{\alpha,\beta}\left(\frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2\xi}\right)\right)\right\} \\
&\leq \xi \cdot \frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2\xi} = \frac{\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}}{2},
\end{aligned} \tag{4.39}$$

where the first inequality is by the definition of ξ. This result means $\Pr\{\mathcal{C}_{\alpha,\beta}(V_q)\}$ is halved after each round of labeling. Therefore, after $Q := \log_2\frac{1}{\varepsilon}$ rounds of labeling,

$$\Delta_{\alpha,\beta}(h) \leq \Pr\{\mathcal{C}_{\alpha,\beta}(V_Q)\} \leq \varepsilon, \tag{4.40}$$

with probability at least $1-Q\delta'$; where the left inequality is by definition. By then, the total number of labeled instances is $\log_2\frac{1}{\varepsilon}\cdot\frac{1}{4\xi^2}\left(32c/\beta + \sqrt{\frac{1}{2}\log\frac{1}{\delta'}}\right)$. Setting $\delta = Q\delta'$ and plugging $\delta' = \delta/Q$ in completes the proof. □

Example 4.4.4 (Example 4.4)**.** Fix $\alpha, \beta > 0$. Let $h_z(x) = 1_{x>z}$ be a threshold function defined on $[0, 1]$. Let $H = \{h_z; \alpha \leq z \leq 1-\alpha\}$. Assume points are uniformly distributed in $[0, 1]$. Then, $\Delta_{\alpha,\beta}(h) = \frac{\alpha(1-\alpha^2)}{1-2\alpha}$ and $\xi \leq \frac{1}{2}$.

Proof. One can apply the same proof strategy and show $\Delta_{\alpha,\beta}(h) = \frac{\alpha(1-\alpha^2)}{1-2\alpha} \geq \alpha(1+\alpha)$. (For two points to be within α distance apart, the probability is $(1-2\alpha)2\alpha + 2\alpha(\alpha+\frac{\alpha}{2}) = 1-\alpha^2$; for z to land in a certain α-length interval within its whole range of length $1-2\alpha$, the probability is $\alpha/(1-2\alpha)$), $\mathcal{C}_{\alpha,\beta}(\mathcal{B}_{\alpha,\beta}(r)) = [z-\frac{\alpha}{2}, z] \times [z, z+\frac{\alpha}{2}] \cup [z, z+\frac{\alpha}{2}] \times [z-\frac{\alpha}{2}, z]$, so $\Pr[\mathcal{C}_{\alpha,\beta}(\mathcal{B}_{\alpha,\beta}(r))] = \frac{\alpha^2}{2} \leq \frac{r}{2}$ (because $\alpha+\alpha^2 \leq r$ implies $\alpha^2 \leq r$). Hence $\xi \leq \frac{1}{2}$. □

Lemma 4.4.5 (Lemma 5.1)**.** *Fix any $\alpha, \beta > 0$. One have $\Delta_{\alpha,\beta}(h; S) \leq \tilde{\Delta}_{\alpha,\beta}(h; S)$ for any $h \in S$ and sample S.*

Proof. Since $\mathbb{I}_{x \geq t} \leq \frac{x}{t}$ for any $x, t \geq 0$, one have

$$\begin{aligned}
&\mathbb{I}\{d(x_i, x_j) \leq \alpha,\, |h(x_i) - h(x_j)| \geq \beta\} \\
&\quad = \mathbb{I}\{d(x_i, x_j) \leq \alpha\} \cdot \mathbb{I}\{|h(x_i) - h(x_j)|^2 \geq \beta^2\} \\
&\quad \leq \frac{1}{\beta^2} \cdot \mathbb{I}\{d(x_i, x_j) \leq \alpha\} \cdot |h(x_i) - h(x_j)|^2 \qquad (4.41) \\
&\quad = \frac{1}{\beta^2} \cdot M_{ij} \cdot |h(x_i) - h(x_j)|^2.
\end{aligned}$$

Plugging this back to (4.6) proves the lemma. □

Chapter 5

Fairness-Aware Active Learning for Decoupled Model

5.1 Proposed Algorithm

In this section, [CL22b] propose a disagreement-based fairness-aware active learning strategy (D-FA^{2}L) for decoupled model.

Let (x, s, y) be an instance, where x is the vector of non-protected features, s is a binary protected feature, and y is a binary label. A decoupled model is a pair of models (h_0, h_1) such that h_i is applied on instances with $s = i$. A standard decoupled supervised learner $\mathcal{A}_d$ trains model h_i from instances with $s = i$ using any standard supervised learning technique.

[CL22b] focus on the classification problem, but assume model h_i will output the posterior probability as $h_i(x) = \Pr\{y = 1 \mid x\}$ instead of directly outputting the binary class – one can obtain the latter by thresholding model output as $\mathbb{1}_{h(x)>0.5}$, where $\mathbb{1}$ is an indicator function.

The basic idea of the proposed strategy is to query label for instance u that receives significantly different predictions from the decoupled models h_0 and h_1, i.e., $|h_1(u) \neq h_0(u)| > \alpha$ for some preset threshold α. This is motivated by the classic disagreement-based active labeling strategy [H$^+$14], and can be viewed as its extension from single model to decoupled model. Detailed connection and difference are discussed later.

The proposed D-FA^{2}L process is presented in Algorithm 2. In practice, one can stop labeling when a preset number of labels are queried or a desired model performance is achieved.

5.2 Theoretical Analysis

In this section, [CL22b] theoretically analyze D-FA^{2}L. The main insight is that model unfairness is bounded by a notion of α-distance, which can

Algorithm 2 Disagreement-based FA^{2}L Algorithm (D-FA^{2}L)

Input: an initial labeled training set L, a pool of unlabeled data set U, a hypothesis class H, a standard decoupled supervised learner $\mathcal{A}_d$, hyper-parameters α, k.

a decoupled model (h_0, h_1).

1: Train $(h_0, h_1) \in H \times H$ from L using $\mathcal{A}_d$.
2: **while** Stopping criterion is not met **do**
3: Independently and uniformly select k instances $u \in U$ satisfying $|h_0(u) - h_1(u)| > \alpha$
4: Label the selected instances, add them to L, and remove them from U.
5: Retrain (h_0, h_1) from L using $\mathcal{A}_d$.
6: **end while**

be efficiently reduced by D-FA^{2}L under proper conditions. In the rest of this section, we first introduce a set of definitions, then present the main theoretical results, and finally discuss the impact on model accuracy.

5.2.1 Notations and Definitions

Recall (x, s, y) denotes an arbitrary instance. Let (X, S, Y) be the random variable from which (x, s, y) is sampled. Let μ_* be the joint sampling probability distribution on (X, S) that induces $\mu(x) = \mathbb{P}\{X = x\}$ and $\mu_s(x) = \mathbb{P}\{X = x \mid S = s\}$.

Let H be a hypothesis class from which the decoupled models are learned. Recall each model $h \in H$ is defined as

$$h(x) = \mathbb{P}\{Y = 1 \mid X = x,\ h\}, \tag{5.1}$$

where randomness comes from the prediction uncertainty of h on a fixed x. The predicted label of x is $\mathbb{1}_{h(x)>0.5}$.

Recall (h_0, h_1) denotes a decoupled model. [CL22b] evaluate its fairness based on a popular notion called disparate impact [BS16]. Specifically, for any (h_0, h_1), define

$$\texttt{DI}(h_0, h_1) = \frac{\mathbb{P}\{h_0(x) > 0.5 \mid S = 0\}}{\mathbb{P}\{h_1(x) > 0.5 \mid S = 1\}}. \tag{5.2}$$

If $\texttt{DI}(h_0, h_1)$ is closer to 1, then (h_0, h_1) is more fair.

[CL22b] analyze how D-FA^{2}L helps to learn a model with small $\texttt{DI}(h_0, h_1)$. Define the α-distance of (h_0, h_1) as

$$d_\alpha(h_0, h_1) = \mathbb{P}\{|h_0(x) - h_1(x)| > \alpha\}, \tag{5.3}$$

where randomness comes from the uncertainty of x w.r.t. μ. An estimate of the distance on a sample J is

$$d_\alpha(h_0, h_1; J) = \frac{1}{|J|} \sum_{x \in J} \mathbb{1}\{|h_0(x) - h_1(x)| > \alpha\}. \tag{5.4}$$

[CL22b] assume the decoupled model returned by a standard learner $\mathcal{A}_d$ has the following form of large margin.

Definition 5.2.1. (h_0, h_1) has an α-margin of γ if

$$\mathbb{P}\{|h_0(x) - h_1(x)| > \alpha \mid \mathbb{1}_{h_0(x)>0.5} \neq \mathbb{1}_{h_1(x)>0.5}\} \geq \gamma. \tag{5.5}$$

Intuitively, a large α-margin means the decoupled model makes confident prediction on most instances. This is a reasonable assumption e.g., it is not hard to show a standard large margin learner that achieves hard margin α on h_0 and h_1 promises $\gamma = 1$, if one treat $|h_i(x) - 0.5|$ as the margin.

This analysis will also involve the distance between the data distributions of the two groups, defined as follows.

Definition 5.2.2. The relative total variation distance between distributions μ_0 and μ_1 on any c-weighted data set is

$$\lambda_c := \max_{\mathcal{Q} \in \Omega_c} \frac{|\mu_0(\mathcal{Q}) - \mu_1(\mathcal{Q})|}{|\mu_0(\mathcal{Q}) + \mu_1(\mathcal{Q})|}, \tag{5.6}$$

where $\Omega_c = \{T \subseteq X; \mu_0(T) + \mu_1(T) \geq c\}$.

One can see that λ_c is small if the two distributions are similar, and the following is an example.

Example 5.2.3. If $X = \mathbb{R}$ and $\mu_i = N(\theta_i, \sigma^2)$, then $\lambda_c \leq \frac{|\theta_0 - \theta_1|}{2\sigma c}$.

Proof. By the total variation distance bound between two Gaussian [DMR18, Theorem 1.3], one have $\max_{\mathcal{Q}} |\mu_0(\mathcal{Q}) - \mu_1(\mathcal{Q})| \leq \frac{|\theta_0 - \theta_1|}{2\sigma}$. The rest follows by the definition of λ_c. □

Finally, inspection suggests that D-FA^{2}L is essentially selecting data in a special region that helps the learner to reduce model unfairness efficiently. This region is defined as follows.

Definition 5.2.4. Let $V \subseteq H \times H$ be any hypothesis space of the decoupled model. The α-controversial region of V is

$$\mathcal{C}_\alpha(V) = \{x \in X;\ \exists (h_0, h_1) \in V,\ |h_0(x) - h_1(x)| > \alpha\}. \tag{5.7}$$

The α-controversial coefficient is

$$\xi_\alpha = \sup_{\delta > 0} \frac{\mathbb{P}_{x \sim \mu}\{\mathcal{C}_\alpha(\Sigma_{\alpha,\delta})\}}{\delta}, \tag{5.8}$$

where $\Sigma_{\alpha,\delta} = \{(h_0, h_1) \in H \times H;\ d_\alpha(h_0, h_1) \leq \delta\}$.

[CL22b] assume ξ_α is bounded. The following is an example.

5.2.2 Main Theoretical Results

The first result shows that, the disparity of any decoupled model with large α-margin is 'squeezed' by their α-distance.

Theorem 5.2.5. *Let $\alpha, \gamma, c > 0$. Any model (h_0, h_1) with an α-margin of γ and $\mathbb{P}\{h_1(x) > 0.5 \mid S = 1\} \geq c$ satisfies*

$$\frac{1-\lambda_c}{1+\lambda_c} - \frac{d_\alpha(h_0,h_1)}{c\gamma p_0} \leq \texttt{DI}(h_0,h_1) \leq \frac{1+\lambda_c}{1-\lambda_c} + \frac{d_\alpha(h_0,h_1)}{c\gamma p_0}, \tag{5.9}$$

where $p_0 = \mathbb{P}\{S = 0\}$.

Proof. To facilitate discussion, let us assume w.l.o.g. that X is finite so all μ_i's are probability mass functions. Write

$$\mathbb{P}\{h_i(x) > 0.5 \mid S = i\} = \sum_{x \in X} I_i^x \mu_i^x, \tag{5.10}$$

where $I_i^x = \mathbb{1}_{h_i(x)>0.5}$ and $\mu_i^x = \mu_i(x)$.

[CL22b] first bound $\sum_x I_0^x \mu_0^x - \sum_x I_1^x \mu_1^x$, which equals to

$$\left(\sum_x I_0^x \mu_0^x - \sum_x I_1^x \mu_0^x\right) + \left(\sum_x I_1^x \mu_0^x - \sum_x I_1^x \mu_1^x\right). \tag{5.11}$$

The 1st parenthesized term equals to

$$\begin{aligned} \sum_x (I_0^x - I_1^x)\, \mu_0^x &\leq \sum_x \mathbb{1}_{I_0^x \neq I_1^x} \cdot \mu_0^x \\ &\leq \frac{1}{p_0} \sum_x \mathbb{1}_{I_0^x \neq I_1^x} \cdot \mu(x), \end{aligned} \tag{5.12}$$

where $p_0 = \mathbb{P}\{S = 0\}$. The first inequality is by case-studying I_0^x and I_1^x so that $I_0^x - I_1^x \leq \mathbb{1}_{I_0^x \neq I_1^x}$ for any x, and the second inequality is by definition so that $\mu_0(x) = \mathbb{P}\{X = x, S = 0\}/p_0 \leq \mu(x)/p_0$. Note the rightmost sum is $\mathbb{P}\{I_0^x \neq I_1^x\}$ and

$$\begin{aligned} &\mathbb{P}\{I_0^x \neq I_1^x\} \\ &= \mathbb{P}\{I_0^x \neq I_1^x,\, |h_0(x) - h_1(x)| > \alpha\} \\ &\qquad + \mathbb{P}\{I_0^x \neq I_1^x,\, |h_0(x) - h_1(x)| \leq \alpha\} \\ &\leq (1/\gamma) \cdot \mathbb{P}\{I_0^x \neq I_1^x,\, |h_0(x) - h_1(x)| > \alpha\} \\ &\leq (1/\gamma) \cdot d_\alpha(h_0, h_1), \end{aligned} \tag{5.13}$$

where the first inequality is due to the α-margin assumption.

For the 2nd parenthesized term in (5.11), since $\sum_x I_1^x \mu_1^x \geq c$,

$$\sum_x I_1^x \mu_0^x - \sum_x I_1^x \mu_1^x \leq \left(\frac{1+\lambda_c}{1-\lambda_c} - 1\right) \sum_x I_1^x \mu_1^x. \tag{5.14}$$

Putting all back to (5.11), one have

$$\sum_x I_0^x \mu_0^x - \left(\frac{1+\lambda_c}{1-\lambda_c}\right) \sum_x I_1^x \mu_1^x \ \leq \ \frac{d_\alpha(h_0, h_1)}{\gamma p_0}. \tag{5.15}$$

By symmetric arguments, bounding $\sum_x I_1^x \mu_1^x - \sum_x I_0^x \mu_0^x$ gives

$$\sum_x I_0^x \mu_0^x - \left(\frac{1-\lambda_c}{1+\lambda_c}\right) \sum_x I_1^x \mu_1^x \ \geq \ -\frac{d_\alpha(h_0, h_1)}{\gamma p_0}. \tag{5.16}$$

Combining (5.15), (5.16) and $\sum_x I_1^x \mu_1^x \geq c$ proves the theorem. □

The main implication of Theorem 5.2.5 is one can reduce the unfairness of a decoupled model by reducing its α-distance.

The next theoretical result shows D-FA^{2}L can efficiently reduce the α-distance under four assumptions: **(A1)** H has a Rademacher complexity of $\mathcal{R}_m(H) \in O(1/\sqrt{m})$, **(A2)** instances are selected i.i.d. in $\mathcal{C}_\alpha(V)$, **(A3)** $d_{\alpha/2}(h_0, h_1; J) \leq 1/4\xi_\alpha$ and **(A4)** ξ_α is bounded. Assumptions (A1, A2) are common e.g., [BM02]. (A3) is supported by our empirical observation that $\hat{d}_{\frac{\alpha}{2}}(h_0, h_1)$ often drops to a small value.

The second theoretical results is stated as follows.

Theorem 5.2.6. *Under (A1, A2, A3, A4), with probably at least $1-t$, D-FA^{2}L returns a model with $d_\alpha(h_0, h_1) \leq \epsilon$ after labeling*

$$O\left(\frac{\xi_\alpha^2 \cdot \log_{1/2} \epsilon}{\alpha^2} \cdot \left(32 + \sqrt{2 \log \frac{4 \log_{1/2} \epsilon}{t}}\right)^2\right) \tag{5.17}$$

instances.

Proof. Let $V_j \subseteq H \times H$ be the hypothesis space satisfying (A3) on the training set before the j_{th} round of labeling. Note that$V_0 \supseteq V_1 \supseteq V_2 \ldots$. Based on (A1, A2), with probability at least $1 - t'$, any decoupled model $(h_0, h_1) \in V_{j+1}$ satisfies

$$d_\alpha(h_0, h_1) \leq \hat{d}_{\frac{\alpha}{2}}(h_0, h_1; J) + \frac{K\left(32 + \sqrt{2\log(4/t')}\right)}{\alpha\sqrt{|L|}}. \tag{5.18}$$

The proof of (5.18) is similar to [YR18, Theorem 4.1], relying on standard Rademacher arguments plus a Lipschitz approximation of the 0-1 loss function. Its proof is deferred to Section 5.5.

Combining (5.18) and (A4), one can see if D-FA^{2}L labels

$$N = \frac{K\xi_\alpha^2}{\alpha^2}\left(32 + \sqrt{2\log\frac{4}{t'}}\right)^2 \tag{5.19}$$

instances in $C_\alpha(V_j)$ in the $(j+1)_{th}$ round of labeling[1], then with probability at least $1-t'$,

$$\mathbb{P}\{|h_0(x) - h_1(x)| > \alpha \mid x \in C_\alpha(V_j)\} \leq 1/(2\xi_\alpha), \tag{5.20}$$

for any $(h_0, h_1) \in V_{j+1}$. This further implies

$$\begin{aligned}
&\mathbb{P}\{|h_0(x) - h_1(x)| > \alpha\} \\
&= \mathbb{P}\{|h_0(x) - h_1(x)| > \alpha \mid x \in C_\alpha(V_j)\} \cdot \mathbb{P}\{x \in C_\alpha(V_j)\} \\
&\quad + \mathbb{P}\{|h_0(x) - h_1(x)| > \alpha \mid x \notin C_\alpha(V_j)\} \cdot \mathbb{P}\{x \notin C_\alpha(V_j)\} \\
&= \mathbb{P}\{|h_0(x) - h_1(x)| > \alpha \mid x \in C_\alpha(V_j)\} \cdot \mathbb{P}\{x \in C_\alpha(V_j)\} \\
&\leq \frac{\mathbb{P}\{x \in C_\alpha(V_j)\}}{2\xi_\alpha} \\
&:= r_\alpha,
\end{aligned} \tag{5.21}$$

where the second equality is by the definition of $\mathcal{C}_\alpha(V)$ so that $\mathbb{P}\{|h_0(x) - h_1(x)| > \alpha \mid x \notin C_\alpha(V_j)\} = 0$. This result implies any $(h_0, h_1) \in V_{j+1}$ falls in Σ_{α,r_α}. Thus $V_{j+1} \subseteq \Sigma_{\alpha,r_\alpha}$ and

$$\begin{aligned}
\frac{\Pr\{x \in C_\alpha(V_{j+1})\}}{\xi_\alpha} &\leq \frac{\Pr\{x \in C_\alpha(\Sigma_{\alpha,r_\alpha})\}}{\xi_\alpha} \\
&\leq r_\alpha \\
&= \frac{\Pr\{x \in C_\alpha(V_j)\}}{2\xi_\alpha},
\end{aligned} \tag{5.22}$$

where the second inequality is by the definition of ξ_α.

Result (5.22) implies that $\Pr\{C_\alpha(V_{j+1})\} \leq \frac{1}{2}\Pr\{C_\alpha(V_j)\}$ with probability at least $1-t'$. Then, after M rounds of labeling, $\mathbb{P}\{C_\alpha(V_M)\} \leq (1/2)^M$ with probability at least $1 - Mt'$. Setting $(1/2)^M = \epsilon$ gives $M = \log_{1/2}\epsilon$, and setting $Mt' = t$ gives $t' = t/M = t/\log_{1/2}\epsilon$. Putting these back to (5.19), one have that $d_\alpha(h_0, h_1) \leq \epsilon$ with probability at least $1-t$ after labeling $M \cdot N$ instances. This proves the theorem. □

Theorem 5.2.6 implies D-FA^{2}L can reduce the α-distance and the reduction is efficient under proper conditions. Combining this with Theorem 5.2.5 explains how D-FA^{2}L achieves fairness.

[1]This is obtained by setting last term in (5.18) to $1/4\xi_\alpha$ and solving for $|L|$.

5.2.3 Impact of D-FA^{2}L on Model Accuracy

In the previous section, it is shown that data training data labeled by D-FA^{2}L helps to improve model fairness. In this section, their impact on model accuracy is discussed.

Define the disagreement region w.r.t. model (h_0, h_1) as

$$\mathcal{D}(h_0, h_1) = \{x \in X; \mathbb{1}_{h_0(x)>0.5} \neq \mathbb{1}_{h_1(x)>0.5}\}, \tag{5.23}$$

and the controversial region w.r.t. model (h_0, h_1) as

$$\mathcal{C}_\alpha(h_0, h_1) = \{x \in X; |h_0(x) - h_1(x)| > \alpha\}. \tag{5.24}$$

In D-FA^{2}L, data are selected by model (h_0, h_1) uniformly at random from $\mathcal{C}_\alpha(h_0, h_1)$. On the other hand, the active learning literature says training data help to improve accuracy if selected from the following disagreement region [H$^+$14, Section 2]

$$DIS(H) = \cup_{(h_0,h_1)\in V}\ \mathcal{D}(h_0, h_1), \tag{5.25}$$

where $V = H \times H$. This leads to the idea that if $\mathcal{C}_\alpha(h_0, h_1)$ overlaps with $DIS(H)$ to a proper degree, then data selected by D-FA^{2}L could help improve accuracy with proper probability. In the following paragraphs, [CL22b] examine this insight through a toy example.

Figure 5.1(a) shows a population represented by their predictions received from a fixed decoupled model (h_0, h_1). Each point $(a, b) \in [0, 1] \times [0, 1]$ represents an instance (or, the set of instances) receiving $h_0(x) = a$ and $h_1(x) = b$. Through simple geometric arguments, one can show $\mathcal{C}_\alpha(h_0, h_1)$ is union of the upper-left and lower-right isosceles right triangles with leg length $1 - \alpha$, and $\mathcal{D}(h_0, h_1)$ and is union of the upper-left and lower-right squares with side length 0.5. Their overlapping region is union of the two shaded regions.

If $\alpha < 0.5$, simple geometric arguments shows the relative area of the overlapping region is

$$\frac{|\mathcal{C}_\alpha(h_0, h_1) \cap \mathcal{D}(h_0, h_1)|}{|\mathcal{C}_\alpha(h_0, h_1)|} = \frac{0.5 - \alpha^2}{(1 - \alpha)^2}, \tag{5.26}$$

where $|\cdot|$ denotes the area of a region. Apparently, this area is larger if α is larger, implying a larger overlapping between $\mathcal{C}_\alpha(h_0, h_1)$ and $DIS(V)$. This further implies D-FA^{2}L could improve accuracy more efficiently if α is large.

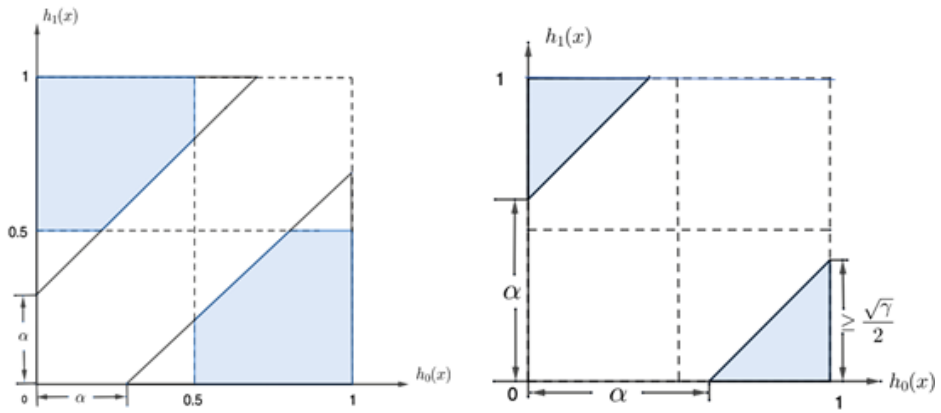
If $\alpha \geq 0.5$, the overlapping region is show in Figure 5.1(b). In this case, the shaded region always has

$$\frac{|\mathcal{C}_\alpha(h_0, h_1) \cap \mathcal{D}(h_0, h_1)|}{|\mathcal{C}_\alpha(h_0, h_1)|} = 1, \tag{5.27}$$

which implies

$$\mathcal{C}_\alpha(h_0, h_1) \subseteq \mathcal{D}(h_0, h_1) \subseteq DIS(H). \tag{5.28}$$

This implies D-FA^{2}L could improve accuracy efficiently.



(a) $\alpha < 0.5$ (b) $\alpha > 0.5$

Figure 5.1: Visualization of Disagreement and Controversial Region under Different α Values

5.3 Experiments

5.3.1 Data Preparation

[CL22b] experiment on two public data sets. The COMPAS data set contains 6172 instances described by 12 numerical attributes. Attribute 'Two-Year-Recidivism' is treated as label, and 'African-American' treated as the protected attribute.

The Crime and Community data set contains 1994 instances described by 128 attributes. Attribute 'crime rate' is treated as the label, and 'percentage of African American' combined with 'percentage of African American police' is treated as the protected attribute. The label is binarized so that a community with crime rate larger than 0.37 is considered as a high crime community, and is otherwise is considered as a low crime community. The protected attribute is binarized so that a community with less than 50% percentage sum of African American (AA) and AA police is considered a minority AA community, and otherwise it is considered as a non-minority AA community.

On each data set, 0.2% of the data is arbitrarily selected as the initial labeled training set, 25% of the remaining data is selected as the testing set, and the rest data is treated as the pool of unlabeled data. The average performance of each experimented method is then reported over 20 random trials. To increase numerical stability of decoupled supervised learning, in each trial, features of each group is standardized based on its selected training and pool data.

5.3.2 Experiment Design

[CL22b] evaluate the performance of D-FA^{2}L and compare it with other methods. Since existing AL techniques are mostly designed for single model, we do not find any directly comparable active labeling strategy for decoupled model. Nonetheless, we adapt both classic and state-of-the-art methods and come up with the following three baseline strategies.

- Random: Data are selected uniformly at random.
- Uncertainty: It is an adaption of the classic uncertainty-based AL strategy for decoupled model. Data with the highest prediction uncertainty are selected, where the uncertainty of instance x with protected attribute $S = i$ is measured by

$$uncertainty(x) = \frac{1}{|h_i(x) - 0.5|}. \tag{5.29}$$

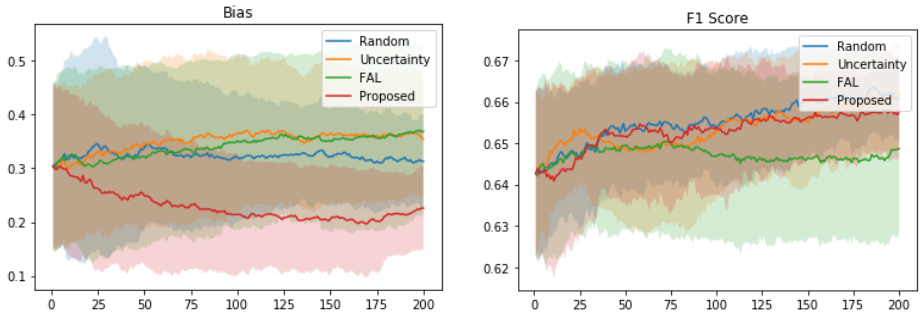
- FAL: It is an adaption of the state-of-the-art fairness-aware active labeling strategy [AAT20] for decoupled model. First, the top m data with the highest prediction entropy are picked, where the entropy of an instance x with $S = i$ is evaluated based on $h_i(x)$. Then, among these data, the ones that could maximally reduce model unfairness are selected, where the reduction of instance with $S = i$ is measured based on h_i.

Logistic regression is used as the base model for all methods, and its regularization coefficient is set to 0.1 as it gives the best performance on initial training set. For D-FA^{2}L, [CL22b] set $\alpha = 0.2$ on the COMPAS data set and $\alpha = 0.7$ on the Community and Crime data set. α can be selected as something slightly lower than the initial $|h_0(x) - h_1(x)|$. When no instance satisfies $|h_0(x) - h_1(x)| > \alpha$, [CL22b] arbitrarily pick an instance to label. For FAL, m is set to 64 as it is reported to give the best performance in the original paper. In each round, all methods label one instance.

Model accuracy is evaluated by F1 score, and model unfairness is evaluated by $\text{Bias}(h_0, h_1) = |\Pr\{h_0(x) > 0.5 \mid S = 0\} - \Pr\{h_1(x) > 0.5 \mid S = 1\}|$. Results on the two data sets are reported in Figures 5.2 and 5.3 respectively.

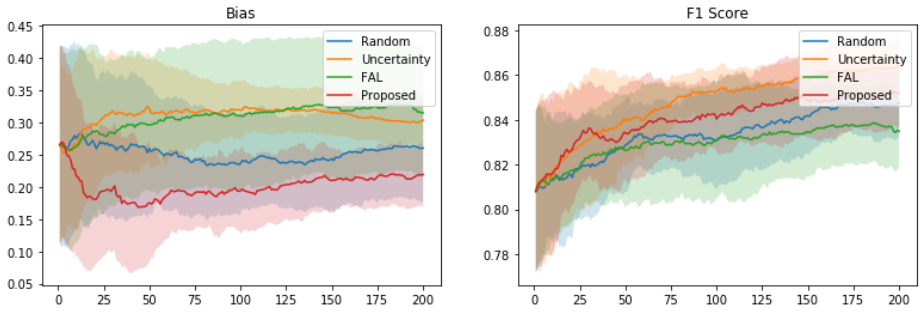
5.3.3 Discussion on the Results

Results on the COMPAS data set are reported in Figure 5.2. One can see D-FA^{2}L is the only method that reduces model bias. In the meantime, D-FA^{2}L improves accuracy as efficiently as other strategies, suggesting it has a small trade-off between fairness and accuracy. These observations are consistent with the theoretical analysis. Results on the Community and



(a) Model Bias vs. Number of Samples (b) F1 Score vs. Number of Samples

Figure 5.2: Performance on the COMPAS Dataset



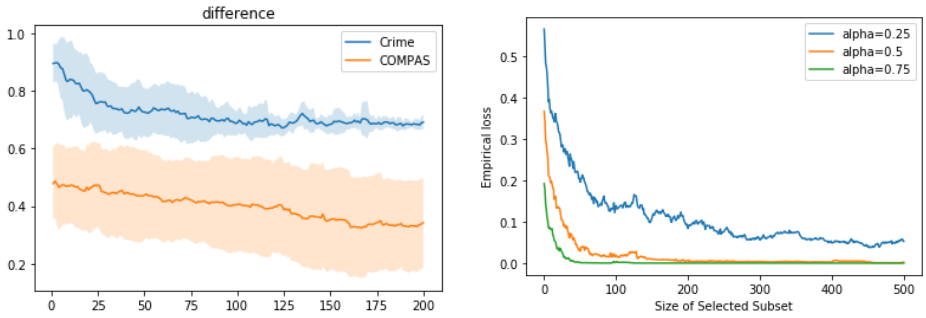
(a) Model Bias vs. Number of Samples (b) F1 Score vs. Number of Samples

Figure 5.3: Performance on the Crime Data Set

Crime data set are shown in Figure 5.3. We see D-FA^{2}L effectively reduce model bias while maintaining a F1 score that is somewhere between that of the random strategy and uncertainty strategy. This further verifies the efficacy of the proposed strategy. FAL does not appear very efficient, which may imply that in the setting of decoupled model, the uncertain instances are rarely useful for improving model fairness.

When applying D-FA^{2}L, one may notice that model bias keeps decreasing on the COMPAS dataset but stops decreasing after about 50 rounds of labeling on the Crime dataset. This is partly related to the different distributions of $|h_0(x) - h_1(x)|$ on the two data sets. In Figure 5.4(a), one can see the averaged $\Delta(x) = |h_0(x) - h_1(x)|$ of the instances labeled by D-FA^{2}L at each round. On Crime, one can see $\Delta(x)$ approaches the threshold $\alpha = 0.7$ at about 50 rounds after which D-FA^{2}L loses the developed theoretical guarantee on bias reduction. On COMPAS, one can see $\Delta(x)$ mostly stays above its threshold $\alpha = 0.2$ and D-FA^{2}L maintains its guarantee on bias reduction.

In Figure 5.5(b), the performance of D-FA^{2}L versus α is shown. One can



(a) $|h_0(x) - h_1(x)|$ of Instance Labeled by D-FA^{2}L

(b) $d_\alpha(h_0, h_1)$ under different α.

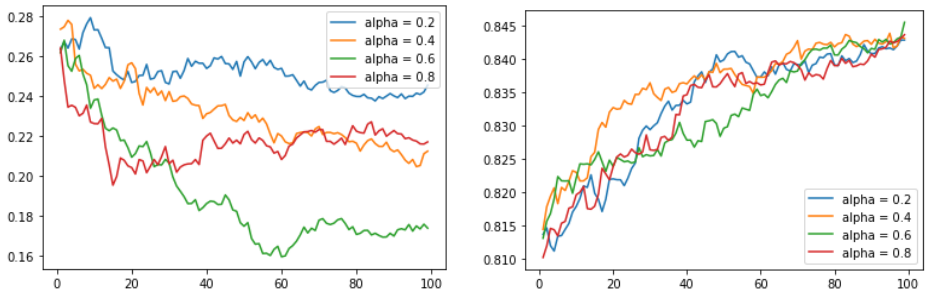
Figure 5.4: Selection of α

see the bias reduction performance is not monotonic w.r.t. α, and among the examined values, $\alpha = 0.6$ achieves the highest efficiency. This makes sense: if α is too small, most instances would fall in the controversial region so selecting one from them is similar to selecting one randomly. On the other hand, if α is too large, few instances would fall in the controversial region – in this case, although D-FA^{2}L can quickly reduce bias, it will also quickly run out of data to select and start performing random selection. This explains why the red curve converges more quickly but only to a limited value in Figure 5.5(b) (a). Surprisingly, α seems to have limited impact on model accuracy, allowing us to achieve a more efficient fairness-accuracy trade-off.

Next $d_\alpha(h_0, h_1)$ is examined. D-FA^{2}L is ran on COMPAS over 20 random trials and the average of $d_\alpha(h_0, h_1)$ under $\alpha = 0.25, 0.5, 0.75$ are documented respectively. Results are in Figure 5.4(b). In general, $d_\alpha(h_0, h_1)$ decreases to a small value after a sufficient number of training data are selected. Larger α implies smaller $d_\alpha(h_0, h_1)$, and when $\alpha = 0.7$ one have $d_\alpha(h_0, h_1) \approx 0$. This suggests D-FA^{2}L is effective in reducing $d_\alpha(h_0, h_1)$, even if it applies only a standard learner!

5.4 Conclusion

In this chapter, [CL22b] study a fairly new research problem called fairness-aware active learning. [CL22b] propose the first disagreement-based fairness-aware active labeling algorithm (D-FA^{2}L) for decoupled model. [CL22b] first theoretically analyze D-FA^{2}L, explaining how it can reduce model unfairness effectively and efficiently and how it impacts accuracy. [CL22b] then empirically verify the efficacy of D-FA^{2}L on two data sets, showing it reduces bias more efficiently than the adapted classic and state-of-the-art methods, while maintaining comparable accuracy.



(a) Model Bias (b) F1 Score

Figure 5.5: Performance on the Crime Data Set. Figure 5.5(a) shows model bias decreases in different speed with different α values. To achieve best performance, α should be neither too small or too large. Figure 5.5(b) shows increases in different speed with different α values. α seems to have limited impact on accuracy, which allows a more efficient fairness-accuracy trade-off.

5.5 Proof

In this section, [CL22b] proves the following bound in (5.18).

$$d_\alpha(h_0, h_1) \leq \hat{d}_{\frac{\alpha}{2}}(h_0, h_1; J) + \frac{K\left(32 + \sqrt{2\log(4/t')}\right)}{\alpha\sqrt{|L|}}. \tag{5.30}$$

The proof is done in 6 steps. For a function class F, let $\hat{R}_m(F)$ be its empirical Rademacher complexity over a sample J of size m, and $R_m(F) = \mathbb{E}_J[\hat{R}_m(F)]$ be its Rademacher complexity. Let σ_i be a Rademarcher noise.

Step 1. Define a set of functions

$$H' : [X] \to \{h_0(x) - h_1(x)\}_{(h_0,h_1)\in H\times H}. \tag{5.31}$$

First, prove $R_m(H') \leq 2R_m(H)$. This is true because

$$\begin{aligned}
R_m(H') &= \mathbb{E}_J[\hat{R}_m(H')] \\
&= \mathbb{E}_{J,\sigma}\left[\sup_{(h_0,h_1)\in H^2}\left(\frac{1}{m}\sum_{i=1}^{m}\sigma_i(h_0(x_i)-h_1(x_i))\right)\right] \\
&\leq \mathbb{E}_{J,\sigma}\left[\sup_{h_0\in H}\left(\frac{1}{m}\sum_{i=1}^{m}\sigma_i h_0(x_i)\right)\right. \\
&\quad \left. + \sup_{h_1\in H}\left(\frac{1}{m}\sum_{i=1}^{m}-\sigma_i h_1(x_i)\right)\right] \qquad (5.32) \\
&= \mathbb{E}_{J,\sigma}\left[\sup_{h_0\in H}\left(\frac{1}{m}\sum_{i=1}^{m}\sigma_i h_0(x_i)\right)\right] \\
&\quad + \mathbb{E}_{J,\sigma}\left[\sup_{h_1\in H}\left(\frac{1}{m}\sum_{i=1}^{m}-\sigma_i h_1(x_i)\right)\right] \\
&= 2R_m(H).
\end{aligned}$$

Step 2. Let $abs(\cdot)$ be the absolute value function. Define a set of composed functions:

$$G := abs \circ H' : [X] \to \{|h_0(x) - h_1(x)|\}_{(h_0,h_1)\in H\times H}. \qquad (5.33)$$

One can prove $R_m(G) \leq 4R_m(H)$. This is true because

$$R_m(G) = R_m(abs \circ H') \leq 2R_m(H') \leq 4R_m(H), \qquad (5.34)$$

where the first inequality is based on [BM02, Theorem 12, Fact 4] and the fact that absolute value function has a Lipschitz constant 1, and the second inequality is based on (5.32).

Step 3. Define a composed function

$$\tilde{F}_\alpha := \tau_\alpha^t \circ G, \qquad (5.35)$$

where

$$\tau_\alpha^t(x) = \begin{cases} 0, & x \leq \alpha \\ t(x-\alpha), & \alpha < x < \alpha + \frac{1}{t} \\ 1, & x \geq \alpha + \frac{1}{t}, \end{cases} \qquad (5.36)$$

One can prove $R_m(\tilde{F}) \leq 8\Delta R_m(H)$. This is true because

$$R_m(\tilde{F}) \leq 2tR_m(G) \leq 8tR_m(H), \qquad (5.37)$$

where the first inequality is based on [BM02, Theorem 12, Fact 4] and the fact that τ_α^t has a Lipschitz constant t; the second inequality is based on (5.34).

Step 4. Let $\tau_\alpha(x)$ be an indicator function outputting 1 if $x > \alpha$ and 0 otherwise. There is

$$\tau_{\alpha+\frac{1}{t}}(z) \leq \tau_\alpha^t(z) \leq \tau_\alpha(z), \tag{5.38}$$

for any input z. Define loss functions for a given (h_0, h_1) as

$$\mathcal{L}_\alpha^t(h_0, h_1) = \mathbb{E}_x \tau_\alpha^t(|h_0(x) - h_1(x)|), \tag{5.39}$$

and, similarly,

$$\mathcal{L}_\alpha(h_0, h_1) = \mathbb{E}_x \tau_\alpha(|h_0(x) - h_1(x)|). \tag{5.40}$$

Let $\hat{\mathcal{L}}_\alpha^t$ and $\hat{\mathcal{L}}_\alpha$ be the empirical estimates of $\mathcal{L}_\alpha^t$ and $\mathcal{L}_\alpha$ over sample S, respectively.

Combining (5.38, 5.39,5.40), one can easily show that

$$\mathcal{L}_{\alpha+\frac{1}{t}}(h_0, h_1) \leq \mathcal{L}_\alpha^t(h_0, h_1) \leq \mathcal{L}_\alpha(h_0, h_1), \tag{5.41}$$

and

$$\hat{\mathcal{L}}_{\alpha+\frac{1}{t}}(h_0, h_1) \leq \hat{\mathcal{L}}_\alpha^t(h_0, h_1) \leq \hat{\mathcal{L}}_\alpha(h_0, h_1). \tag{5.42}$$

Further, by a classic error bound e.g., [MRT18, Theorem 3.1], for any (h_0, h_1), with probability at least $1 - \delta$, there is

$$\begin{aligned} \left|\mathcal{L}_\alpha^t(h_0, h_1) - \hat{\mathcal{L}}_\alpha^t(h_0, h_1)\right| &\leq 2R_m(\tilde{F}) + \sqrt{\frac{\ln(4/\delta)}{2m}} \\ &\leq 8tR_m(\mathcal{H}) + \sqrt{\frac{\ln(4/\delta)}{2m}}. \end{aligned} \tag{5.43}$$

where the second inequality is based on (5.37).

Step 5. Assuming $t \geq 1$, it follows

$$\begin{aligned} &\hat{\mathcal{L}}_\alpha(h_0, h_1) \\ &\geq \hat{\mathcal{L}}_\alpha^t(h_0, h_1) \\ &\geq \mathcal{L}_\alpha^t(h_0, h_1) - \left(8tR_m(\mathcal{H}) + \sqrt{\frac{\ln(4/\delta)}{2m}}\right) \\ &\geq \mathcal{L}_\alpha^t(h_0, h_1) - t\left(8R_m(\mathcal{H}) + \sqrt{\frac{\ln(4/\delta)}{2m}}\right) \\ &\geq \mathcal{L}_{\alpha+\frac{1}{t}}(h_0, h_1) - t\left(8R_m(\mathcal{H}) + \sqrt{\frac{\ln(4/\delta)}{2m}}\right), \end{aligned} \tag{5.44}$$

where the first and last inequalities are by (5.42), the second is based on (5.43), and the third is based on $t \geq 1$.

Step 6. By the assumption that $R_m(H) \in O(1/\sqrt{m})$, there exists a constant $C > 0$ such that, for any $(h_0, h_1) \in H \times H$,

$$\begin{aligned}
&\mathbb{P}\{|h_0(x) - h_1(x)| > \alpha + 1/t\} \\
&= \mathcal{L}_{\alpha+\frac{1}{t}}(h_0, h_1) \\
&\leq \hat{\mathcal{L}}_\alpha(h_0, h_1) + t\left(8R_m(H) + \sqrt{\frac{\ln(4/\delta)}{2m}}\right) \qquad (5.45) \\
&\leq \hat{\mathcal{L}}_\alpha(h_0, h_1) + t\left(\frac{8C}{\sqrt{m}} + \sqrt{\frac{\ln(4/\delta)}{2m}}\right),
\end{aligned}$$

where the first inequality is based on (5.44).

Recall $\alpha \in (0, 1)$. Setting $\frac{1}{t} = \alpha$, one have

$$\begin{aligned}
&\mathbb{P}\{|h_0(x) - h_1(x)| > 2\alpha\} \\
&\qquad \leq \hat{\mathcal{L}}_\alpha(h_0, h_1) + \frac{1}{\alpha}\left(\frac{8C}{\sqrt{m}} + \sqrt{\frac{\ln(4/\delta)}{2m}}\right). \qquad (5.46)
\end{aligned}$$

Setting $\alpha' = 2\alpha$ proves (5.18).

Part II

Randomized Machine Learning Methods

Chapter 6

A Model-Agnostic Randomized Learning Framework based on Random Hypothesis Subspace Sampling

6.1 Introduction

Randomized machine learning is a research topic that studies how to randomize the learning process, often with an aim of improving learning efficiency. Representative techniques range from random projection [Vem05] for efficient dimensionality reduction to extremely randomized decision tree [GEW06], and from random Fourier feature [RR08] for efficient kernel methods to random vector functional link [NNSS20] for efficient network training. These techniques have received adequate research interests over the past decades.

When inspecting the literature, one may notice that most randomized learning techniques are model-specific. For example, in [GEW06], tree generation is randomized by using random features to split tree nodes; in [RR08], a kernel machine is randomized by using random Fourier features to approximate kernel functions; in [NNSS20], neural network training is randomized by fixing all but the output weights to random values. While these techniques have achieved promising results on their designated models respecitvely, it remains unclear how they could be applied on other models or guide the design of randomized learners for them.

Random projection [Vem05] is a randomized dimensionality reduction technique, which projects data into a lower dimensional feature space through random linear projections. It can be applied to speed up downstream learning and considered as model-agnostic to the learner. However, the speedup is often limited because the learner is not randomized and

could remain inefficient especially if its time complexity does not depend heavily on the original feature dimension such as kernel methods.

The above observations reveal the lack of a model-agnostic randomized learning framework that not only ties the existing techniques for certain models but also provides guidance on designing randomized learners for other models. [CL22c] believes such framework will help to significantly advance the research and application of randomized machine learning. This motivates the present study.

A major contribution of this paper is the design of a model-agnostic randomized learning framework based on Random Hypothesis Subspace Sampling (RHSS). Given any hypothesis class, it randomly samples k hypotheses and learns a model in their span that best approximates the target model on a set of n training instances. Importantly, this learning process can always be cast as a simple linear least square problem and solvable in $O(nk^2)$ time. In practice, small k often suffices for good performance, which makes RHSS-based learning extremely efficient no matter how complex the given hypothesis class is.

On the theory side, [CL22c] derives the performance guarantee of RHSS from a generic subspace approximation perspective, leveraging properties of metric entropy and random matrices. Under proper conditions, it is shown that the best model learned from the span of k randomly sampled hypotheses can approximate any target model on a fixed data set by up to an $O(k^{-c})$ error with high probability, where c is a constant in $(0, 1)$. Although this bound is not as tight as those developed for model-specific randomized learners such as in [RR17, AKM$^+$17], in experiments RHSS is observed to have similar or even better performance than their counterparts. Nonetheless, it remains an open question on how to bridge the gap theoretically.

On the practical side, [CL22c] demonstrates the applications of RHSS on kernel, neural network and tree based models, and discusses their connections to the existing randomized learners that are specifically designed for these models, including random Fourier features, random vector functional link and extra tree. In experiments, [CL22c] compares the proposed RHSS-based learners with standard learners and model-specific randomized learners. One can see they approach standard learners efficiently as k increases, and often outperform their model-specific counterparts on real-world data sets.

The rest of this chapter is organized as follows: Section 2 reviews related work; Section 3 presents the proposed RHSS framework; Section 4 presents its theoretical analysis and Section 5 demonstrates its applications; experimental results are shown in Section 6 and conclusion in Section 7.

6.2 Related Work

6.2.1 Random Fourier Feature

Random Fourier Features (RFF) is designed to speed up kernel methods [RR08]. It approximates kernel functions using inner products of explicit feature vectors generated through random Fourier functions, and thus bypasses the need of working with the Gram matrix. With n training instances and k random features, RFF reduces the typical time complexity of learning a kernel machine from $O(n^3)$ to $O(nk^2)$, where k is often smaller than n. Because of its outstanding efficiency, RFF has been intensively studied in the past e.g. [YLM+12, Sza15, RR17, AKM+17, Li17, LTOS19].

It remains unclear, however, that how RFF can be applied on non-kernel machines such as network or tree that do not necessarily work with Gram matrices. One may use it as a feature preprocessing technique such as generating a tree based on Fourier features, but there is little guarantee on the accuracy or efficiency of downstream learning. (See Section 2.4 for more discussion on the limitations of randomized feature preprocessing.) Comparatively, the proposed RHSS framework applies to both kernel and non-kernel machines.

6.2.2 Random Vector Functional Link

Random Vector Functional Link (RVFL) is designed to speed up multi-layer perceptron learning [PPS94, IP95]. It only optimizes the weights between the last hidden layer and the output layer, and randomly sets the other weights for the final network. With n training instances and m neurons in the last hidden layer, RVFL can efficiently learn the network in $O(nm^2)$ time. Although proposed in the last century, RVFL is re-gaining research interests in recent years [ZS16, NNSS20, GS20].

Apparently, it is unclear how RVFL can be applied on non-network models such as kernel machine or tree that are not constructed by ordered layers of weights. Comparatively, the proposed RHSS framework applies to both network and non-network models. Interestingly, when applied on multi-layer perceptron, RHSS can be viewed as applying the RVFL principle on a specially constructed network. See Figure 6.1 and related discussions in Section 5.2.

6.2.3 Extra Tree

Extra tree is designed to speed up tree learning [GEW06]. It randomly selects features to split nodes when generating each tree, and outputs the average of multiple trees as the final model. By avoiding the search of optimal

features for node splitting, extra tree is more efficient than standard tree learning and has received successful applications [DPH+12, MWvdG+15].

Similar to RFF and RVFL, however, it is unclear how extra tree can be applied on non-tree models that do not have nodes to split. Comparatively, the proposed RHSS framework applies to both tree and non-tree models. When applied on tree, RHSS uses the same method as extra tree to generate multiple trees, but then outputs an optimally weighted average of them as the final model.

From the tree ensemble perspective, RHSS and extra tree are both connected to random forest. The latter is a powerful non-randomized tree learning method, which also averages multiple trees but each tree finds optimal features (from a sub-pool) to split nodes. [CL22c] does not expect randomized tree learners to beat random forest in accuracy, yet our experimental results suggest they provide good approximations while being significantly more efficient to learn.

In light of the above discussion, one may also see some connection between RHSS and boosting, since the latter also finds an optimal weighted average of models. Yet, they have a fundamental difference that boosting optimizes each model (thus not a randomized learner) whereas RHSS randomly picks each model. Besides, models in boosting are often dependent whereas models in RHSS are i.i.d. sampled.

6.2.4 Random Projection

Random Projection (RP) is design to speed up dimensionality reduction. It maps a set of data into a lower dimensional feature space through randomly generated linear projections [Vem05]. Compared to other reduction methods, RP is more efficient in that it avoids the search of optimal projections which often has a high time complexity such as $O(p^3)$ for p input features in PCA. Besides, it is proved that data distortion in the randomly projected feature space is likely bounded [AV06] and thus RP will not significantly deteriorate the performance of downstream learning [FM03, MM12, DK13].

RP is often applied to speed up downstream learning and considered as model-agnostic to the learner. However, the speedup is often limited because the learner could remain inefficient. For example, after using RP to reduce feature dimension, learning a kernel machine still takes $O(n^3)$ time with n training instances and RVFL still takes $O(nm^2)$ time with m hidden neurons. Comparatively, the proposed RHSS directly speeds up the learner (through approximation). When applied on linear hypothesis class, RHSS is equivalent to RP followed by the learning of a linear model.

From a broader randomized feature projection perspective, another related work is randomized kernel locality sensitive hashing (R-KLSH) [GGVS+19, GGVSC19]. It designs randomized hash functions (with param-

eters) to generate binary features, then optimizes them for label prediction, and finally use them to train multiple tree models that are at the end assembled into a random forest.

Both R-KLSH and RP generate features with randomness for downstream learning, but they have two fundamental difference. First, features of RP are random while features of R-KLSH are semi-random since they are optimized from data for the prediction task. Second, RP speeds up learning by generating few features, but R-KLSH often needs to generate redundant features and achieves speedup based on the binary property of these features that can be efficiently exploited by tree models – from this perspective, R-KLSH is model-specific. These, plus the difference between RP and RHSS, are the difference between R-KLSH and RHSS.

6.3 The RHSS-based Learning Framework

In this section, [CL22c] presents the RHSS-based randomized learning framework. Its basic idea is to randomly sample some hypotheses and then learn an optimal model in their span. Specifically, given a hypothesis class and a set of n training instances, RHSS operates in four steps:

(1) Randomly sample k hypotheses from the class.

(2) Apply each sampled hypothesis on all training instances and obtain an n-dimensional vector of its predicted labels.

(3) Learn an optimal linear combination of the prediction vectors that best approximates the vector of true labels.

(4) Output the combined hypothesis.

Detailed learning process is elaborated in Framework 3. In the framework, x_i is the i_{th} instance in the training set and y_i is its label. The number of sampled hypotheses k is a hyper-parameter. The design of specific hypothesis sampling approach is dependent on the hypothesis class, and three examples are shown in Section 6.5.

Once an output model f is obtained, one can apply it on a testing point z by first applying the sampled hypotheses to obtain $h_1(z), \ldots, h_k(z)$ and then calculating the prediction as $f(z) = \alpha_1 h_1(z) + \ldots + \alpha_k h_k(z)$.

As one can see, the RHSS framework is fairly simple and easy to apply as learning is always formulated as a linear least square problem solvable in $O(nk^2)$ time, no matter how complex the hypothesis class is. In experiments, one can observe that small k often suffices for good performance, which makes RHSS-based learning very efficient.

Next, the theoretical guarantees of RHSS are derived and its applications are demonstrated on three hypothesis classes.

Framework 3 The RHSS-based Learning Framework

Input: hypothesis class H, sampling distribution D, a labeled set $(x_1, y_1), \ldots, (x_n, y_n)$, hyper-parameter k

1: Independently sample $h_1, \ldots, h_k \in H$ based on D.

2: Calculate $\tilde{h}_i = [h_i(x_1), \ldots, h_i(x_n)]^T$ for each i.

3: Optimize coefficients $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$ by solving

$$\min_{\alpha_1,\ldots,\alpha_k} \left\| \sum_{i=1}^{k} \alpha_i \tilde{h}_i - \tilde{y} \right\|^2, \tag{6.1}$$

where $\tilde{y} = [y_1, \ldots, y_n]^T$.

Output: Model $f = \sum_{i=1}^{k} \alpha_i h_i$.

6.4 Theoretical Analysis of RHSS

In this section, the theoretical guarantee of RHSS is derived from a generic subspace approximation perspective, i.e., it is equivalent to approximating a target set using a random subspace spanned by the columns of a random matrix.

First, a reader may refer to the set of analytic tools introduced in Chapter 2, some borrowed from the literature of metric entropy and random matrices and some developed from there (with proofs given in the appendix). Then, the main theoretical results are presented and their implications discussed.

6.4.1 Preliminaries

The analysis of subspace approximation will be performed on the *Grassmannian*. Let $G_{n,\ell}$ be the Grassmannian consisting of all the ℓ-subspaces of $\mathbb{R}^n$, and the distance between any two $U, V \in G_{n,\ell}$ is measured by

$$d_{G,r}(U, V) = \Delta(U \cap S_r, V \cap S_r), \tag{6.2}$$

where Δ is the Hausdorff distance defined as

$$\Delta(X, Y) = \max\{\sup_{x\in X} \inf_{y\in Y} \|x - y\|, \sup_{y\in Y} \inf_{x\in X} \|x - y\|\} \tag{6.3}$$

with $\|\cdot\|$ being the ℓ_2 norm, and S_r is an $n-1$ dimensional sphere with radius r. The Remark 5 in [Sza82] suggests $d_{G,r}$ is also a metric.

To analyze the approximation error, one can use the covering number of $G_{n,\ell}$. Let N_ε be the ε-covering number of $G_{n,\ell}$ w.r.t. $d_{G,r}$, which is defined as the smallest number of ε-balls whose union contains $G_{n,\ell}$, that is,

$$N_\varepsilon = \arg\min_m \cup_{i=1}^{m} B_\varepsilon(U_i) \supseteq G_{n,\ell}, \tag{6.4}$$

where $B_\varepsilon(U_i) = \{V \in G_{n,\ell} \mid d_{G,r}(U_i, V) \leq \varepsilon\}$ is an ε-ball centered at $U_i \in G_{n,\ell}$ and with radius ε; moreover, $B_\varepsilon(U_1), \ldots, B_\varepsilon(U_{N_\varepsilon})$ is called an ε-covering of $G_{n,\ell}$. The following lemma is a scaled version of the Proposition 8 in [Sza82], originally proposed in [Sza], which bounds the covering number.

Lemma 6.4.1. *There exist universal constants c, C such that*

$$\left(\frac{cr}{\varepsilon}\right)^{\ell(n-\ell)} \leq N_\varepsilon \leq \left(\frac{Cr}{\varepsilon}\right)^{\ell(n-\ell)}, \tag{6.5}$$

for any $\varepsilon \in (0, \sqrt{2}]$.

Our analysis also involves the use of a subspace to approximate a finite set. Inspired by the Kolmogorov n-width theory [Pin12], [CL22c] define the distance from a subspace $U \in G_{n,\ell}$ to a finite set $A \subseteq \mathbb{R}^n$ as

$$d_S(A, U) = \sup_{a \in A} \inf_{u \in U} ||a - u||. \tag{6.6}$$

Note that $\inf_{U \in G_{n,\ell}} d_S(A, U)$ is the Kolmogorov ℓ-width of A in $\mathbb{R}^n$, and its value is zero whenever the cardinality of A is no greater than ℓ (because one can always use elements of A as part of a basis to construct U). In addition, [CL22c] develop the following pseudo-triangular inequality based on this distance. Its proof is in Appendix 6.8.1.

Lemma 6.4.2. *For any $U, V \in G_{\ell,n}$ and finite $A \subseteq S_r$,*

$$d_S(A, U) \leq d_S(A, V) + d_{G,r}(V, U). \tag{6.7}$$

The subspace being analyzed will be random, and is actually the row space of a random matrix $\tilde{H}$ constructed as follows: let $H_1, \ldots, H_k$ be the random hypotheses from which the k hypotheses in Framework 3 are sampled respectively, and $x_1, \ldots, x_n$ be a given data set. Construct

$$\tilde{H} = \begin{bmatrix} H_1(x_1) & \ldots & H_1(x_n) \\ \vdots & \ddots & \vdots \\ H_k(x_1) & \ldots & H_k(x_n) \end{bmatrix} = \begin{bmatrix} \tilde{H}_{1:} \\ \vdots \\ \tilde{H}_{k:} \end{bmatrix}, \tag{6.8}$$

where $\tilde{H}_{i:} = [H_i(x_1), \ldots, H_i(x_n)]$ is the i_{th} row and k is typically way smaller than n.

Our analysis will rely on a fixed dimension of the random subspace, and this only occurs with certain probability that can be characterized by the following property of random matrix, which is from [Ver10] Theorem 5.39.

Theorem 6.4.3. *Let M be a k-by-n matrix whose rows are independent sub-gaussian isotropic random vectors in $\mathbb{R}^n$. Let $\sigma_{\min}$ be the smallest singular value of M. Then*

$$\Pr\{\sigma_{\min} < \sqrt{k} - c\sqrt{n} - t\} \leq 2\exp(-Ct^2), \tag{6.9}$$

for any $t \geq 0$, where $c, C > 0$ are constants depending only on the maximum subgaussian norm of the rows.

Using the above theorem, one may develop the following property of $\tilde{H}$. Its proof is in Appendix 6.8.2.

Lemma 6.4.4. *For random matrix $\tilde{H}$ in (6.8), if $H_1, \ldots, H_k$ are i.i.d. and each $\tilde{H}_{i:}$ follows a sub-Gaussian distribution and has an invertible expected outer product, then*

(i) $\tilde{H}_{1:}, \ldots, \tilde{H}_{k:}$ are i.i.d..

(ii) There exist constants a, b depending on the largest sub-Gaussian norm and expected outer product of $\tilde{H}_{i:}$, such that a sample of $\tilde{H}$ has linearly independent rows with probability at least $1 - 2\exp(-b(\sqrt{k} - a\sqrt{n})^2)$.

Our analysis also relies on the following assumption.

Assumption 6.4.5. There exists an ε-covering of $G_{n,\ell}$, denoted by

$$B_\varepsilon(U_1), \ldots, B_\varepsilon(U_{N_\varepsilon}) \tag{6.10}$$

for some $U_1, \ldots, U_{N_\varepsilon} \in G_{n,\ell}$, such that any ℓ-dimensional row span of $\tilde{H}$ is uniformly distributed in $B_\varepsilon(U_1), \ldots, B_\varepsilon(U_{N_\varepsilon})$.

Finally, let f be a model returned by RHSS. Its error will be analyzed on a labeled set $(x_1, y_1), \ldots, (x_n, y_n)$, defined as

$$er_n(f) = \frac{1}{n}\sum_{i=1}^{n}[f(x_i) - y_i]^2. \tag{6.11}$$

Its error is also analyzed on the population, defined as

$$er(f) = \mathbb{E}[f(x) - y]^2, \tag{6.12}$$

where (x, y) denotes a random instance.

6.4.2 Theoretical Analysis of RHSS

The main result is stated as follows.

Theorem 6.4.6. *Suppose $\tilde{H}$ in (6.8) satisfy the conditions in Lemma 6.4.4 and Assumption 6.4.5. Then, there exist constants $a, b, c > 0$ such that any model f returned by Framework 3 satisfies $er_n(f) \leq \varepsilon$ with probability at least $1 - \delta_1 - \delta_2$ (over the random choice of hypotheses), where $\delta_1 = 2\exp(-b(\sqrt{k} - a\sqrt{n})^2)$ and $\delta_2 = \exp(-k(\frac{\sqrt{n\varepsilon}}{2cr})^{(n-1)})$.*

Proof. This proof has four steps: (i) show the vectorized hypotheses are linearly independent with high probability; (ii) show $er_n(f)$ is bounded by some distance d_S; (iii) upper bound the introduced distance; (iv) specify the probability for the upper bound to hold. Details are elaborated below.

Step (i): Probability of Linear Independence

Consider an event that $\tilde{h}_1, \ldots, \tilde{h}_k$ are linearly independent. Let P_1 be the probability this event does not occur. Then, Lemma 6.4.4 suggests there exists constants a, b such that

$$P_1 \leq 2\exp(-b(\sqrt{k} - a\sqrt{n})^2). \tag{6.13}$$

The rest of the analysis will be based on the assumption that the event of linear independence occurs.

Step (ii): Bound $er_n(f)$ by $d_S(\tilde{Y}, \tilde{V}_i)$.

Let ℓ be a proper number (to be picked later) and evenly divide $\tilde{h}_i$'s into $m = k/\ell$ groups. By the assumption in Step (i), each group spans an ℓ-subspace. Let $\tilde{V}_1, \ldots, \tilde{V}_m \in G_{n,l}$ be the ℓ-subspaces spanned by the m groups, respectively.

Then, RHSS can be viewed as using one $\tilde{V}_i$ to approximate the target set $\tilde{Y} = \{\tilde{y}\}$, as it finds a model whose vectorized representation $\tilde{f} = [f(x_1), \ldots, f(x_n)]^T \in \tilde{V}_i$ has the smallest distance to $\tilde{y} \in \tilde{Y}$, i.e.,

$$||\tilde{f} - \tilde{y}|| = \sup_{\tilde{y}\in\tilde{Y}} \min_{\tilde{h}\in\tilde{V}_i} ||\tilde{h} - \tilde{y}|| = d_S(\tilde{Y}, \tilde{V}_i). \tag{6.14}$$

Moreover, it is easy to verify (by definition) that

$$er_n(f) = [d_S(\tilde{Y}, \tilde{V}_i)]^2/n. \tag{6.15}$$

Thus to bound $er_n(f)$, it suffices to bound $d_S(\tilde{Y}, \tilde{V}_i)$.

Step (iii): Bound $d_S(\tilde{Y}, \tilde{V}_i)$.

To bound $d_S(\tilde{Y}, \tilde{V}_i)$, one can first apply the developed pseudo-triangular inequality. Let $V_* = \arg\min_{V\in G_{n,\ell}} d_S(\tilde{Y}, V)$ be a subspace that best approximates $\tilde{Y}$ and $r = ||\tilde{y}||$. Then, Lemma 6.4.2 and the remark of (6.6) suggest that

$$\begin{aligned} d_S(\tilde{Y}, \tilde{V}_i) &\leq d_S(\tilde{Y}, V_*) + d_{G,r}(V_*, \tilde{V}_i) \\ &= d_{G,r}(V_*, \tilde{V}_i). \end{aligned} \tag{6.16}$$

To bound $d_{G,r}(V_*, \tilde{V}_i)$, one may apply results on Grassmannian. Recall N_ε is the covering number of $G_{n,\ell}$, and let $B_\varepsilon(U_1), \ldots, B_\varepsilon(U_{N_\varepsilon})$ be the ε-covering in assumption 6.4.5.

By definition, V_* must fall in one of the balls – without loss of generality, assume $V_* \in B_\varepsilon(U_1)$. Now, if $\tilde{V}_i$ also falls in $B_\varepsilon(U_1)$, by the triangular inequality for metric d_G,

$$d_{G,r}(\tilde{V}_i, V_*) \leq d_{G,r}(\tilde{V}_i, U_1) + d_{G,r}(U_1, V_*) \leq 2\varepsilon. \tag{6.17}$$

Plugging (6.16) (6.17) back to (6.15), one have

$$er_n(f) \leq (4\varepsilon^2)/n. \tag{6.18}$$

The quantity $\varepsilon' = (4\varepsilon^2)/n$ implies

$$\varepsilon = \sqrt{\varepsilon' n}/2. \tag{6.19}$$

Step (iv): Specify the Probability for the Bound

It remains to specify the probability for (6.18). By the uniform assumption, the probability for one $\tilde{V}_i$ to fall outside $B_\varepsilon(U_1)$ is $1 - 1/N_\varepsilon$, and for all $\tilde{V}_1, \ldots, \tilde{V}_m$ to fall outside $B_\varepsilon(U_1)$ is $P_2 = (1 - 1/N_\varepsilon)^m$. By Lemma 6.4.1 and (6.19), one have

$$\begin{aligned} P_2 \leq \exp(-\frac{m}{N_\varepsilon}) &\leq \exp(-\frac{k}{\ell}\left(\frac{\varepsilon}{cr}\right)^{\ell(n-\ell)}) \\ &\leq \exp(-\frac{k}{\ell}(\frac{\sqrt{n\varepsilon'}}{2cr})^{\ell(n-\ell)}), \end{aligned} \tag{6.20}$$

for some constant c. Now one can pick ℓ. Simple analysis shows the right side of (6.20) is minimum when $\ell = 1$. Thus

$$P_2 \leq \exp(-k(\frac{\sqrt{n\varepsilon'}}{2cr})^{(n-1)}). \tag{6.21}$$

Finally, combining all by a union bound and replacing ε' with ε proves the theorem. □

Implications of Theorem 6.4.6

In Theorem 6.4.6, the probability for RHSS to have guaranteed performance is determined by δ_1 and δ_2, where the former determines how likely the sampled hypotheses are linearly independent, and the latter determines how likely the output model performs well. The following discussion will focus on discussing the impact of k on both terms, since it is the major hyper-parameter of RHSS.

For δ_1, the impact of k is not monotonic based on $\delta_1 = 2\exp(-b(\sqrt{k} - a\sqrt{n})^2)$. When $\sqrt{k} < a\sqrt{n}$, increasing k will increase δ_1 and generate a weaker guarantee; otherwise, increasing k will decrease δ_1. This implies if one wants all sampled hypotheses to be linearly independent, one could sample either very few or a lot. (Fortunately, in experiment one can see a few is sufficient for good performance.)

For δ_2, it monotonically decreases as k increases based on $\delta_2 = \exp(-k(\frac{\sqrt{n\varepsilon}}{2c\|\tilde{y}\|})^{n-1})$, giving a higher probabilistic guarantee. When δ_2 is held constant, one can see that increasing k allows one to pick a smaller ε. This implies sampling more hypotheses allows one to have a smaller error guarantee.

Since both δ_1 and δ_2 have the form $\exp(-c_n k)$ for some c_n, together they provide a strong guarantee that $er_n(f) > \varepsilon$ with probability at most $\exp(-ck)$, which drops exponentially fast as k increases.

It is worth mentioning that, δ_2 often dominates δ_1 in practice, especially for large n and small k. Moreover, it is easy to show $\delta_1 = 0$ if more ideal sampling distributions can be assumed, such as those remarked in following corollary.

Corollary 6.4.7. *If $\tilde{H}_1, \ldots, \tilde{H}_k$, defined in (6.8), are independently and uniformly distributed, then there exists constant $c > 0$ as in Lemma 6.4.1 such that any model f output from Framework 3 satisfies $er_n(f) \leq \varepsilon$ with probability at least $1 - \delta$, where $\delta = \exp(-k(\frac{\sqrt{n\varepsilon}}{2cr})^{(n-1)})$.*

Extension to Generalization Error

In this section, the above result is extended from a data set to the population using Rademacher complexity. Recall such complexity[1] of a hypothesis class H w.r.t. random inputs $x_1, \ldots, x_n$ is defined as

$$\mathcal{R}_n(H) = \mathbb{E}_x \mathbb{E}_t \sup_{h \in H} \frac{1}{n} \left| \sum\nolimits_{i=1}^{n} t_i h(x_i) \right|, \tag{6.22}$$

where $t_1, \ldots, t_n$ are independent random variables uniformly picked from $\{-1, +1\}$. Let k, M be two constants and define the following hypothesis class

$$F = \left\{ \sum\nolimits_{i=1}^{k} \alpha_i h_i \mid h_i \in H, \alpha_i \in \mathbb{R}. |\alpha_i| \leq M \right\}, \tag{6.23}$$

The following relation can be developed between $\mathcal{R}_n(F)$ and $\mathcal{R}_n(H)$. Its proof is in Appendix 6.8.3.

Theorem 6.4.8. *For any finite $K, M, n > 0$,*

$$\mathcal{R}_n(F) = Mk \cdot \mathcal{R}_n(H). \tag{6.24}$$

Combining this with standard generalization arguments e.g., Theorem 3.3 in [MRT18] and Theorem 6.4.8, one may obtain the following generalization error bound for RHSS.

Theorem 6.4.9. *In Framework 3, suppose all hypotheses are bounded by a constant $T > 0$ and the n instances are sampled i.i.d.. Then there exists a constant M depending on λ such that, with probability at least $1 - \delta$, any output model f satisfies $er(f) \leq er_n(f) + 8TMk\,\mathcal{R}_n(H) + T^2\sqrt{\frac{8 \log \frac{1}{\delta}}{n}}$.*

[1]Here the version with absolute value is adopted.

To better interpret the bound, consider the scenario in Corollary 6.4.7 which implies $er_n(f) \leq \frac{4c^2r^2}{n}(\frac{1}{k}\log\frac{1}{\delta})^{\frac{1}{n-1}}$. Plugging this into Theorem 6.4.9, the error bound becomes

$$\frac{4c^2r^2}{n}(\frac{1}{k}\log\frac{1}{\delta})^{\frac{1}{n-1}} + 8TMk\,\mathcal{R}_n(H) + O(\frac{1}{\sqrt{n}}). \tag{6.25}$$

One can see k balances the first two terms, as increasing it will decrease the 1st term (approximation error) but increase the 2nd term (complexity of F). This makes perfect sense.

Interestingly, the balance suggests an optimal k that could minimize the error bound. Let $J(k)$ be the sum of the first two terms in (6.25). Solving $\frac{\partial J(k)}{\partial k} = 0$ gives this optimal $k = \left(\frac{c^2r^2(\log\frac{1}{\delta})^{\frac{2-n}{n-1}}}{2TM\mathcal{R}_n(H)n(n-1)}\right)^{\frac{n-1}{n}}$. Plugging this back to (6.25) and assuming n is sufficiently large, it is easy to show the 1st term is in $O(\frac{1}{n})$ and the 2nd term is in $O(\frac{1}{n^2})$.

Note that both terms are much smaller than the 3rd term, which is in $O(\frac{1}{\sqrt{n}})$ and induced solely from generalization. Then, Theorem 6.4.9 suggests the error induced from approximation is negligible compared to the error induced from generalization. This presents a theoretical justification on the effectiveness of RHSS, and is also consistent with our experimental results.

6.5 Applications of RHSS

Applying RHSS is straightforward: sample some hypotheses, get their predictions and learn their best combination. In practice, the design of hypothesis sampling depends on the model. Three examples are given in this section.

6.5.1 Kernel Ridge Regression (KRR)

Let ϕ be an (implicit) feature mapping and H_ϕ be the set of all linear hypotheses in the mapped space. Standard KRR learns a model in H_ϕ in $O(n^3)$ time.

[CL22c] propose RHSS based KRR (RHSS-KRR), which randomly samples hypotheses from

$$H_\phi = \left\{ \sum_{i=1}^{n} \beta_i \phi(x_i) \mid \beta_i \in \mathbb{R} \right\}. \tag{6.26}$$

Specifically, a hypothesis is sampled by independently sampling its β_i's from a proper distribution such as Gaussian.

Suppose the j_{th} hypothesis $h_j = \sum_{i=1}^{n} \beta_i^j \phi(x_i)$ is sampled. Its prediction vector can be evaluated as

$$\tilde{h}_j = [h_j(\phi(x_1)), \ldots, h_j(\phi(x_n))]^T = K \cdot \vec{\beta}_j, \tag{6.27}$$

where $\vec{\beta}_j = [\beta_1^j, \ldots, \beta_n^j]^T$ and K is the Gram matrix.

In terms of time complexity, RHSS-KRR takes $O(nk^2)$ to learn the output model, which is more efficient than the standard KRR that takes $O(n^3)$, and equally efficient as random Fourier Feature that takes $O(nk^2)$ with k random features. The process of sampling and evaluating $\tilde{h}_1, \ldots, \tilde{h}_k$ takes $O(n^2k)$, which is less efficient than RFF which takes $O(nk)$ but still more efficient than standard KRR.

6.5.2 Multi-Layer Perceptron (MLP)

Let H_τ be the set of MLPs with the same architecture τ which specifies the number of hidden layers, number of neurons per layer and activation functions. Standard MLP learning is done through back-propagation.

[CL22c] propose RHSS based MLP (RHSS-MLP), which randomly samples a network in H_τ by independently sampling all its weights from a proper distribution such as Gaussian.

RHSS-MLP takes $O(nk^2)$ to learn the output model, while RVFL takes $O(nm_\tau^2)$ with m_τ being the number of neurons in the last hidden layer. Interestingly, one can view RHSS-MLP as applying the RVFL principle on a network with special architecture, as illustrated in Figure 6.1.

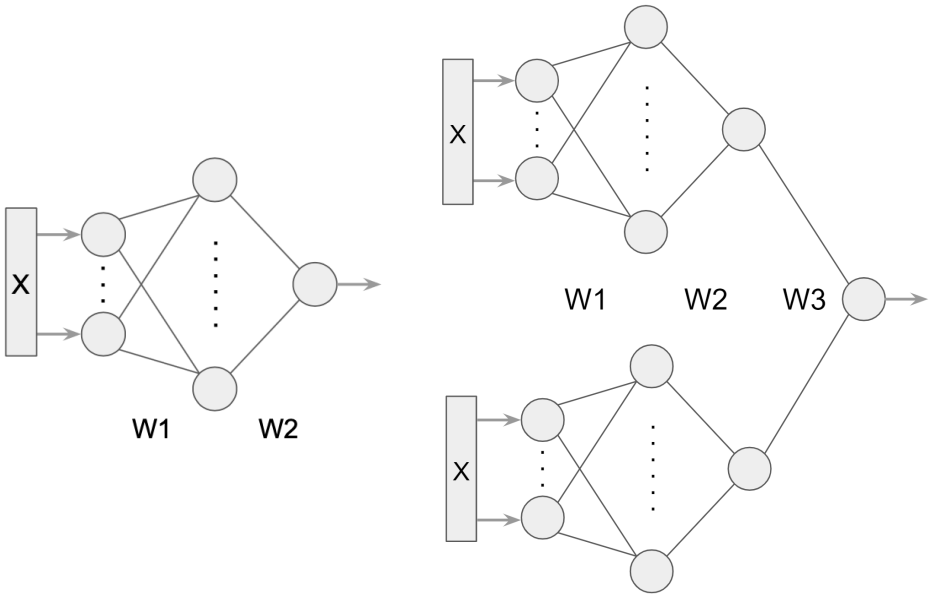
Figure 6.1(a) shows an MLP with a single hidden layer. Let W1 be the set of weights between the input and hidden layers, and W2 be the set of weights between the hidden and output layers. Back-propagation optimizes W1 and W2, while RVFL randomly sets W1 and only optimizes W2.

Figure 6.1(b) shows the corresponding network of RHSS-MLP. It has k blocks of MLP's, as k sampled hypotheses, and combines them at the end. Let W3 be the set of weights between the output and the last hidden layer. RHSS-MLP randomly sets W1 and W2, and only optimizes W3.

6.5.3 Decision Tree

Let H_τ be the set of decision trees that can be generated based on a feature set τ. Standard tree learning algorithms find optimal features to split tree nodes.

We propose RHSS based tree (RHSS-Tree), which randomly samples trees in H_τ by applying the extra tree generation technique [GEW06] on bootstrap samples. More specifically, one can sample a tree by randomly selecting features to split its nodes. Bootstrapping is necessary in this application, since different trees generated by an extra tree will have the



(a) MLP/RVFL (b) RHSS-MLP (k = 2)

Figure 6.1: Architectures of MLP, RVFL and RHSS-RVFL

same predictions on the training set, making the optimization in (6.1) useless (all α_i's are identical).

Standard tree learning and extra tree learning have the same time complexity, although the latter is faster since it avoids the time of finding optimal features for node split. RHSS-Tree has the same time complexity as it applies extra tree.

6.6 Experiments

[CL22c] compares the performance of the proposed RHSS-KRR, RHSS-MLP and RHSS-Tree with their existing randomized counterparts on three public real-world data sets, namely, Crime and Community, Adult and COMPAS. On each data set, we use the first half of the instances for training and the other half for testing. To account for the randomness in randomized learners, [CL22c] runs each learner for 20 times and report its average performance and standard deviation. It focuses on reporting accuracy of the trained models (measured by their rooted mean square errors) versus hyper-parameter k. In all figures, the k values are log-scaled.

The following discussion first presents three sets of experiments, each comparing one RHSS based learner with its existing randomized counterpart. Then, [CL22c] performs a set of sensitivity analysis including the impact

of RP on RHSS. The codes of all experiments are available at `https://github.com/yxc827/RHSS`.

6.6.1 Comparisons with Existing Randomized Learners

[CL22c] designs three sets of comparisons, each based on one proposed application of RHSS in Section 6.5.

The first set compares RHSS-KRR with standard KRR and Random Fourier Feature based KRR (RFF-KRR). It uses RBF kernel with width optimized to 1e-3. For RHSS-KRR, the hypothesis coefficients are sampled independently from $N(0, 1)$. It choses this distribution simply because it is common, but RHSS actually seems fairly robust across different sampling distributions. (See results in the next section.) Both KRR and RFF-KRR apply ridge regression, and the regularization coefficient is optimized to 0.1 on Crime and 0.001 on the other two data sets. In this experiment, k is the number of sampled hypotheses for RHSS-KRR and the number of random features for RFF-KRR. Results are shown in Figure 6.2(a) 6.2(b) 6.2(c). One can see RHSS-KRR converges slightly faster than RFF-KRR, and converges to KRR at around $k = 100$.

The second set compares RHSS-MLP with standard MLP and RVFL. Since RVFL is mainly designed for the single-hidden-layer architecture, [CL22c] apply this architecture with 20 hidden neurons and ReLU activation function. For MLP and RVFL, the regularization coefficient is optimized to 10. For RHSS-MLP and RVFL, all non-optimized parameters are independently sampled from $N(0, 1)$. Results are shown in Figure 6.2(d) 6.2(e) 6.2(f). One can see RHSS-MLP converges to MLP when around $k = 100$ and offers a better approximation of MLP than RVFL at that point on two data sets.

The third set compares RHSS-Tree with decision tree, extra tree and random forest. In this experiment, k is the number of sampled hypotheses for RHSS-Tree and the number of trees for extra tree and random forest. For RHSS-Tree, bootstrap sample size is set to 80% of the original training set. The configurations of all other methods are set as default. Results are shown in Figure 6.2(g) 6.2(h) 6.2(i). One can see RHSS-Tree and extra tree both outperform decision tree as k increases to a small number, and can well approximate the powerful random forest on two data sets. On COMPAS, RHSS-Tree slightly outperforms random forest.

Figures 6.2(j) 6.2(k) 6.2(l) show the training time (in terms of seconds) of all methods on Crime. We see those popular randomized learners RFF-KRR, RVFL and extra tree are indeed extremely efficient, followed by RHSS based learners. They are all a lot faster than standard learners.

Overall, one can see RHSS provides an efficient and effective randomized

learning framework for different models.

6.6.2 Experiment with RP and Sensitivity Analysis

In this section, the performance of random projection (RP) for KRR is evaluated on the Crime and Community data set. [CL22c] experiments two methods: (i) RP-KRR first applies RP and then applies KRR; (ii) RP-RHSS-KRR: first applies RP and then applies RHSS-KRR. In both methods, k is the projected dimension of RP, and we fix the number of sampled hypotheses to 100 for RHSS-KRR. For RP, all projection entries are independently sampled from $N(0, 0.01)$. All other configurations are the same as before. Results are shown in Figure 6.2(m). We see RP-KRR also offers a good approximation to KRR as k increases. However, it does not speed up learning as much as RHSS-KRR, as shown in Figure 6.2(j). One also see RP-RHSS-KRR is not as efficient as other methods, leaving how to effectively combine randomized dimensionality reduction method and randomized learning method an open question.

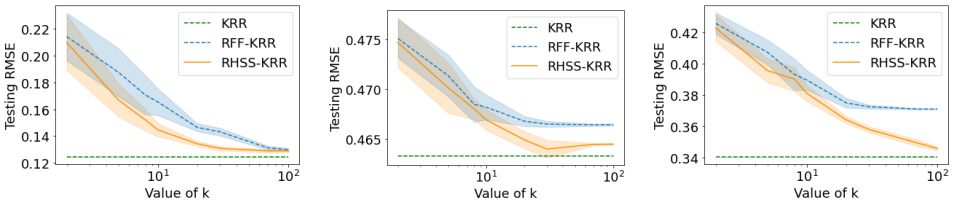
Next, we evaluate the performance of RHSS-KRR on Crime with different sampling distributions. Keeping all other configurations, [CL22c] includes four distributions: (i) Gaussian $N(0, 1)$; (iii) uniform in $[-\sqrt{3}, \sqrt{3}]$, (iii) Laplace with zero mean and unit scale, and (iv) symmetric Bernoulli with $p = 0.5$. Results are shown in Figure 6.2(o). One can see RHSS is fairly robust across the different sampling distributions.

Finally, the performance of RVFL and RHSS-MLP are evaluated when the network architecture varies. Specifically, the number of hidden neurons is increased, with k fixed to 100 for RHSS-MLP, and results are reported in Figure 6.2(n). One can see RVFL improves as more hidden neurons are added, which is a known result. The impact of hidden neurons on RHSS-KRR is limited, however. Our general observation is that RHSS based methods are mainly affected by k.

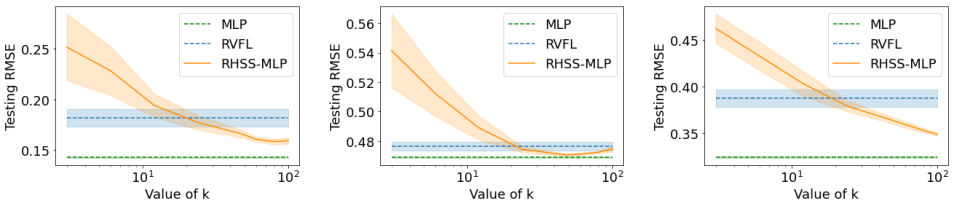
6.7 Conclusion

This chapter presents a model-agnostic randomized learning framework based on Random Hypothesis Subspace Sampling (RHSS), which ties the popular model-specific randomized learners and provides a more unified base for the future developments of randomized machine learning.

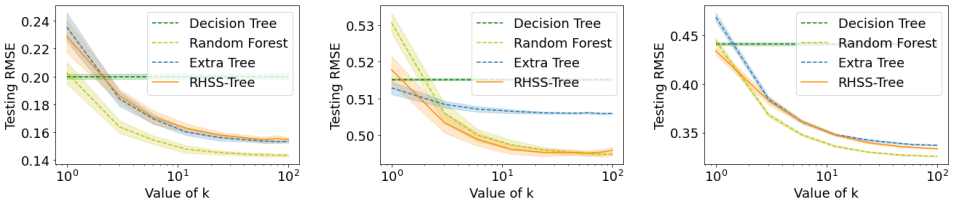
The proposed RHSS framework is simple and easy to apply, and cast learning for any hypothesis class as a linear least square problem solvable in $O(nk^2)$ time with n training instances and k sampled hypotheses. On the theory side, [CL22c] derives error bounds for RHSS and show the approximation error is negligible compared to the generalization error,



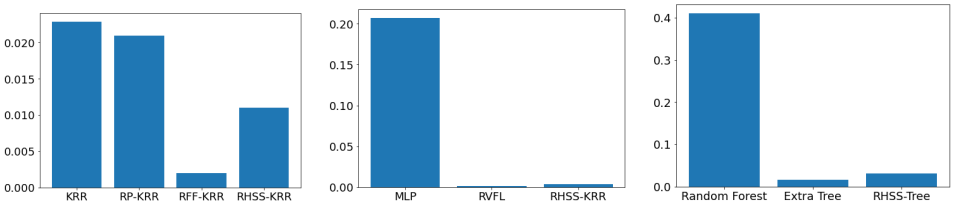
(a) RHSS-KRR on Crime (b) RHSS-KRR on COMPAS (c) RHSS-KRR on Adult



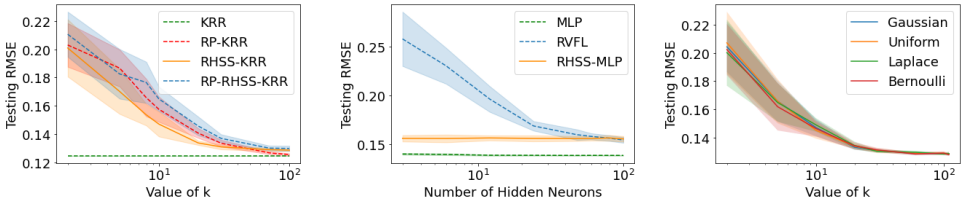
(d) RHSS-MLP on Crime (e) RHSS-MLP on COMPAS (f) RHSS-MLP on Adult



(g) RHSS-Tree on Crime (h) RHSS-Tree on COMPAS (i) RHSS-Tree on Adult



(j) Training Time of Kernel Machines (k) Training Time of Network Models (l) Training Time of Tree Models



(m) Performance of Random Projection (n) Performance of RVFL (o) Sampling Distributions of RHSS-KRR

Figure 6.2: Performance of RHSS based Learning Algorithms on Three Data Sets

which theoretically justifies its effectiveness. On the practical side, [CL22c] demonstrates the applications of RHSS on kernel, neural network and tree based models. In experiments, one can see the proposed RHSS-based learners converge efficiently to standard learners and often outperform their model-specific randomized counterparts, including random Fourier feature, RVFL and extra tree, on real-world data sets. Our results suggest a strong practical value of the proposed unifying framework.

6.8 Appendix

6.8.1 Proof of Lemma 6.4.2

Lemma 6.4.2. For any $U, V \in G_{\ell,n}$ and finite $A \subseteq S_r$,

$$d_S(A, U) \leq d_S(A, V) + d_{G,r}(V, U). \tag{6.28}$$

Proof. Let $a \in A$, $u \in U$ and $v \in V$. There is

$$||a - u|| \leq ||a - v|| + ||v - u||. \tag{6.29}$$

Taking infimum of $u \in U$ on both sides of (6.29) gives

$$\inf_{u \in U} ||a - u|| \leq ||a - v|| + \inf_{u \in U} ||v - u||. \tag{6.30}$$

Let $v_a = \arg\inf_{v \in V} ||a - v||$. Taking infimum of $v \in V$ on both sides of (6.30) gives

$$\begin{aligned} \inf_{u \in U} ||a - u|| &\leq \inf_{v \in V} (||a - v|| + \inf_{u \in U} ||v - u||) \\ &\leq ||a - v_a|| + \inf_{u \in U} ||v_a - u|| \\ &= \inf_{v \in V} ||a - v|| + \inf_{u \in U} ||v_a - u||. \end{aligned} \tag{6.31}$$

Since $A \subseteq S_r$, we have $||a|| = r$ and thus $||v_a|| = r$. Then

$$\inf_{u \in U} ||v_a - u|| \leq \inf_{u \in U \cap S_r} ||v_a - u|| \leq \sup_{v \in V \cap S_r} \inf_{u \in U \cap S_r} ||v - u|| \leq D_{G,r}(V, U). \tag{6.32}$$

Plugging (6.32) back to (6.31) and taking supremum of $a \in A$ on both sides of the inequality proves the lemma. □

6.8.2 Proof of Lemma 6.4.4

Lemma 6.4.4. For random matrix $\tilde{H}$ in (6.8), if $H_1, \ldots, H_k$ are i.i.d. and each $\tilde{H}_{i:}$ follows a sub-Gaussian distribution and has an invertible expected outer product, then

(i) $\tilde{H}_{1:}, \ldots, \tilde{H}_{k:}$ are i.i.d..

(ii) There exist constants a, b depending on the largest sub-Gaussian norm and expected outer product of $\tilde{H}_{i:}$, such that a sample of $\tilde{H}$ has linearly independent rows with probability at least $1 - 2\exp(-b(\sqrt{k} - a\sqrt{n})^2)$.

Proof. We first prove (i). To show any two rows have identical distribution is trivial. To show they are independent, let $\rho(x; E) = \{h \in H; h(x) \in E\}$. Then, for any two H_i, H_j, fixed inputs x, z and sets E_1, E_2, we have

$$\begin{aligned}\Pr\{H_i(x) \in E_1, H_j(z) \in E_2\} &= \Pr\{H_i \in \rho(x; E_1), H_j \in \rho(z; E_2)\} \\ &= \Pr\{H_i \in \rho(x; E_1)\} \cdot \Pr\{H_j \in \rho(z; E_2)\} \\ &= \Pr\{H_i(x) \in E_1\} \cdot \Pr\{H_j(z) \in E_2\},\end{aligned} \tag{6.33}$$

where the third line is by the independence assumption. The argument can be readily generalized to all inputs which implies $[H_i(x_1), \ldots, H_i(x_n)]$ and $[H_j(x_1), \ldots, H_j(x_n)]$ are independent vectors. This proves claim (i).

Now we prove (ii). Let $\Sigma = E[\tilde{H}_{i:}^T \tilde{H}_{i:}]$. Let $\tilde{H}'$ be an k-by-n matrix whose i_{th} row is

$$\tilde{H}'_{i:} = \Sigma^{-1/2} \tilde{H}_{i:}. \tag{6.34}$$

It is easy to show $\tilde{H}'_{i:}$ has i.i.d. sub-Gaussian isotropic rows.

Let $\sigma'_{\min}$ be the least singular value of $\tilde{H}'$. By Theorem 6.4.3,

$$\Pr\{\sigma'_{\min} < \sqrt{k} - a'\sqrt{n} - t\} \leq 2\exp(-bt^2), \tag{6.35}$$

for some constants a', b.

Now, pick an arbitrarily small $\varepsilon > 0$ and set $t = \sqrt{k} - (a' + \varepsilon)\sqrt{n}$, we have $\Pr\{\sigma'_{\min} = 0\} \leq \Pr\{\sigma'_{\min} < \varepsilon\sqrt{n}\} \leq 2\exp(-b[k + (a')^2 n - 2a'\sqrt{nk}])$. Further, let $\sigma_{\min}$ be the least singular value of $\tilde{H}$. Then $\Pr\{\sigma_{\min} = 0\} \leq \Pr\{\sigma'_{\min} = 0\}$, since a sample of $\tilde{H}$ with linearly dependent rows implies the existence of a sample of $\tilde{H}'$ with linearly dependent rows constructed through (6.34). Setting $a = (a')^2$ and putting all together prove claim (ii). □

6.8.3 Proof of Theorem 6.4.8

Theorem 6.4.8. For any finite $K, M, n > 0$,

$$\mathcal{R}_n(F) = MK \cdot \mathcal{R}_n(H). \tag{6.36}$$

Proof. For compact presentation, we will omit subscripts in the expectation in $\mathcal{R}_n$, and use α and h to denote the set of $\alpha_1, \ldots, \alpha_K$ and set of $h_1, \ldots, h_K$ respectively.

We first prove $\mathcal{R}_n(F) \leq MK\mathcal{R}_n(H)$. This is true because

$$\begin{aligned}
\mathcal{R}_n(F) &= \frac{1}{n}\mathbb{E}\sup_{\alpha,h}\left|\sum\nolimits_{i=1}^{n} t_i \cdot \left(\sum\nolimits_{j=1}^{K} \alpha_j h_j(x_i)\right)\right| \\
&= \frac{1}{n}\mathbb{E}\sup_{\alpha,h}\left|\sum\nolimits_{j=1}^{K} \alpha_j \cdot \left(\sum\nolimits_{i=1}^{n} t_i h_j(x_i)\right)\right| \\
&\leq \frac{1}{n}\mathbb{E}\sup_{\alpha,h}\sum\nolimits_{j=1}^{K} |\alpha_j| \cdot \left|\sum\nolimits_{i=1}^{n} t_i h_j(x_i)\right| \\
&= \frac{1}{n}\mathbb{E}\sum\nolimits_{j=1}^{K} M \cdot \sup_{h_j}\left|\sum\nolimits_{i=1}^{n} t_i h_j(x_i)\right| \\
&= \sum\nolimits_{j=1}^{K} M \cdot \frac{1}{n}\mathbb{E}\sup_{h_j}\left|\sum\nolimits_{i=1}^{n} t_i h_j(x_i)\right| \\
&= MK \cdot \mathcal{R}_n(H),
\end{aligned} \tag{6.37}$$

where the third line is by the triangular inequality (for any fixed α, h), the fourth line is by the definition of supremum so that the sum and product of non-negative variables are maximized when these variables are maximized; the fifth line is by the linearity of expectation.

Next we prove $\mathcal{R}_n(F) \geq MK\mathcal{R}_n(H)$. This is true because

$$\begin{aligned}
\mathcal{R}_n(F) &= \frac{1}{n}\mathbb{E}\sup_{\alpha,h}\left|\sum\nolimits_{j=1}^{K} \alpha_j \cdot \left(\sum\nolimits_{i=1}^{n} t_i h_j(x_i)\right)\right| \\
&\geq \frac{1}{n}\mathbb{E}\sup_{h}\left|\sum\nolimits_{j=1}^{K} M \cdot \left(\sum\nolimits_{i=1}^{n} t_i h_j(x_i)\right)\right| \\
&\geq \frac{1}{n}\mathbb{E}\sup_{h_j=h'}\left|\sum\nolimits_{j=1}^{K} M \cdot \left(\sum\nolimits_{i=1}^{n} t_i h'(x_i)\right)\right| \\
&= MK \cdot \frac{1}{n}\mathbb{E}\sup_{h'}\left|\sum\nolimits_{i=1}^{n} t_i h'(x_i)\right| \\
&= MK \cdot \mathcal{R}_n(H),
\end{aligned} \tag{6.38}$$

where the third line is obtained by setting $\alpha_1, \ldots, \alpha_K$ to M; the fourth line is obtained by adding a constraint $h_1 = \ldots = h_K$ when taking the supremum. Combining (6.37) and (6.38) proves the theorem. □

Chapter 7

SpectralSketches: Scaling Up Spectral Norm Estimation through Sketchings

7.1 Introduction

Spectral norm, which is the largest singular value, has long been of interest to the machine learning community. Such interest naturally comes from the birth of kernel-based methods in the early days. It re-emerged under the spotlights due to recent popularity of neural networks and advances in neural tangent kernels, building equivalence between neural networks and kernels.

In this era of big data, the computation of spectral norm is known to be a bottleneck to scalability issues due to its $O(n^3)$ time complexity. Historically, approximation is done through power iterations or Lanczos Algorithm [Lan50] and their variations. Power iterations directly outputs an estimate of the greatest eigenvalue in absolute value (for real symmetric matrices, this coincides with the largest singular value), whereas Lanczos algorithm usually output a tridiagonal square matrix with another matrix with orthonormal columns and one can compute the estimate from the small $k \times k$ squared matrix in $O(k^2)$. However, these methods converge slowly and are only computationally efficient for sparse matrices.

Research on power methods and Lanczos continued throughout decades since spectral norm estimation is important for many machine learning problems. In the last century, this was crucial for principal component analysis (PCA) [Pea01] and for all vibration problems of physics and engineering [Lan50]. Developments in this century continued. Spectral norm estimation is applied in many classical machine learning methods including many clustering methods and Support Vector Machine (SVM) [KK10, TBRS13]. Moreover, a new line of work named spectral learning

begin to thrive. Spectral learning concerns solving an optimization problem which involves regularization with a spectral penalty term [AMP10]. It applies to multi-task learning, collaborative filtering and so on [AMP10, AM05]. In particular, spectral norm regularization is found effective under different senarios in matrix completion, graph embeddings, adversarial learning, and computer vision[MHT10, SCS+15, BMCM19, RKH20, ZHX+21]. Spectral norm is also often used to give upper estimates of approximation errors [DMC05, GM13, BFT17] and serve as a measure of trainability during neural architecture search [XPS20]. Its ability to signal trainability leads to neural architecture search algorithms being developed using spectral norm estimates as an important step[CGW21, DL21].

Existing research on spectral norm estimation focus on iteration-based methods. Our contribution is to show, theoretically and empirically, a more numerically stable and more scalable new paradigm to estimate spectral norm is through sketching-based algorithms (SpectralSketches). We use random fourier features and Nystrom method as classical examples of sketching, representing data-independent and data dependent schemes. While many improved variants are proposed, they still fall in the iteration-based framework and have $O(n^2)$ in terms of sample size n. Comparatively, our proposed method is $O(n)$ in terms of n. Our goal is to demonstrate how sketching may be used in spectral norm approximation, explain what would be the best setting to apply it, and advocate it as a promising alternative to classical iteration-based estimations in the resource-constrained scenario.

The rest of this paper is organized as follows. Section 2 discusses work related to spectral norm approximation and sketching. Sections 3 states the related and proposed algorithms. Section 4 details the corresponding approximation bounds and compares the SpectralNystrom bound to power methods and Lanczos algorithm. Section 5 records the experimental results and discusses its connection to theoretical gaurantees in Section 4. Conclusion is given in Section 6.

7.2 Related Work

7.2.1 Spectral Norm Approximation

Spectral norm is widely used in machine learning, from principal component analysis [Pea01], variations of clustering and support vector machine [KK10, TBRS13], to neural architecture search [XPS20, CGW21]. It is also critical for multi-task learning [APYM07], matrix completion [MHT10], computer vision [ZHX+21], adversarial learning [RKH20, ZHX+21], deep learning [BMCM19] etc. through spectral learning or spectral regularization [AM05, AMP10, SCS+15]. Interests in spectral norm estimation results from all the applications mentioned above. Historically, they mainly come from

popularity of kernel-based methods and recently by and large due to the connection established between kernel machines and neural networks through neural tangent kernels (NTKs) [JGH18].

Many methods have been developed to estimate spectral norm. A basic method to estimate spectral norm is the power iteration. However, power iteration is known to be numerically unstable and converges slowly. Lanczos algorithm is an improvement over Power method introduced by Richard von Mises in 1929. It first got proposed in [RLHK51]. Improvement with complete reorthogonalization suggested in [Wil58], and the Lanczos algorithm for singular value decomposition appeared in [GK65]. Block Lanczos came up in [GU77] for sparse symmetric matrices, which is still the main subject of interest for research today. For additional reference, [Kom03] summarizes the development of Lanczos method well. After decades of research on improvements, [MM15] gives the state-of-the-art version of randomized block Lanczos, which they call randomized block Krylov. They give a faster algorithm with stronger guarantees for sparse matrices. While their contribution is solid, the case for non-sparse matrices is beyond the scope of their paper. Their algorithm would still take $O(n^2k)$ to compute for nonsparse matrix with k being the number of iterations and this time complexity is instrinsic to iteration-based algorithms due to the need for computing matrix-vector product at each iteration.

7.2.2 Sketching

Sketching is a useful numerical linear algebra technique which aims at compressing a big matrix into a small matrix so that the more expensive computations can be done on the small matrix instead [W+14]. Historically, sketching-based methods are used to speed up matrix multiplications [DKM06], finding least square solutions [DMMS11, LWM18], finding low-rank approximation to matrices [LWM+07], finding approximate nearest neighbors [AC06] and so on.

Random Fourier Features (RFF) [RR07] and Nystrom [DMC05] are two famous methods that fall under the umbrella of sketching. The former is data-independent while the latter is data-dependent. They have both seen numerous improvements in various aspects since their debut. Some representative examples include [YSC+16, Bac17, DDSR17, MKBO18, YSSK20] for speeding up RFF convergence, [MM17] for speeding up Nystrom to run in linear time with respect to the number of samples, [RCCR18] further speed up the leverage score sampling part of Nystrom and [DKM20, VSSB22] gives latest improvement on convergence bound ([VSSB22] for sparse matrices). They are also widely applied to other commonly used machine learning structures and problems, including transformer [CZJY21, XZC+21] and k-means clustering [WGM19]. They additionally apply to spectral norm

estimation through SpectralSketches, which we will introduce below.

7.2.3 Estimating Spectral Norm based on Sketched Matrix

The idea of estimating spectral norm of a square matrix by computing spectral norm of a compressed version of it is quite intuitive. Naturally, there are existing analysis that is based on some version of this idea, Eg. [T+15, Git11]. They tend to focus on error ε instead of sample size dimension n, which may be of theoretical interest by default but shows slow convergence rate. Such slow convergence rate is inherited from random features, which conveys the conclusion that this idea is of little practical value. To summarize, those analysis has slightly different settings and fail to capture the strength of this method.

Specifically, result from [T+15] is linear in the matrix dimension while mine has an extra exponentially decaying multiplicative factor that contains the matrix dimension. [T+15] uses $d_1 \times d_2$ for matrix dimension and mine use $n \times n$. I rephrase [T+15]'s result using n in this paragraph. Comparable result is (6.2.6) or (6.5.7) in [T+15]. One can directly compare with (6.2.6), which has the same left hand side as us and the right hand side is $O(n)$ while mine is $O(n \exp(-1/n))$. This means my bound is tighter. One may also apply Markov's inequality to (6.5.7) then compare that with mine, which also gives $O(n)$ on the RHS, similar to (6.2.6). This is likely due to the fact that (6.2.6) is for general not-necessarily-symmetric matrices, which doesn't make use of the symmetry. Furthermore, since (6.5.7) is proven using (6.2.6), such limitation is inherited and Markov's inequality is also loose.

7.3 The SpectralSketches Framework

In this section, I present SpectralSketches (Algorithm 4), a framework for computing spectral norm estimates through sketching. Its basic idea is using square of the largest singular value for sketches to provide estimation for the spectral norm of the underlying symmetric prototype of those sketches. Detailed process is elaborated in Algorithm 4.

In SpectralSketches, any sketching algorithm may be used, although one may need to adjust the output of the chosen sketching algorithm accordingly. For example, while the output $Z \in R^{D \times n}$ from random fourier features (Algorithm 5) may be used directly by SpectralRFF, the output $\tilde{G}_k$ from the Nystrom method (Algorithm 6) need minor adjustment: a best rank-k approximation typically won't return W_k directly [DM18], it usually returns an orthogonal matrix U along with k eigenvalues instead. Forming

Algorithm 4 SpectralSketches

Input: $n \times d$ data matrix X and the kernel map $k : R^d \times R^d \to R$
Compute $k \times n$ matrix Z with a sketching algorithm of choice s.t. $Z^T Z$ approximates K
return: $\sigma^2_{\max}(Z)$ as an approximation to $\lambda_{\max}(K)$

Algorithm 5 Random Fourier Features

Input: A positive definite shift-invariant kernel $k(x, y) = k(x - y)$, samples $x_1, \ldots, x_n \in R^d$
Compute the Fourier transform p of the kernel k: $p(w) = \frac{1}{2\pi} \int e^{iw^T \delta} k(\delta) d\delta$.
Draw D i.i.d. samples $w_1, \ldots, w_D \in R^d$ from p and D i.i.d. samples $b_1, \ldots, b_D \in R$ from unif$[0, 2\pi]$.
return: an $D \times n$ matrix Z formed by
$z(x_i) = \sqrt{\frac{2}{D}}[\cos(w_1^T x_i + b_1) \cos(w_2^T x_i + b_2) \ldots \cos(w_D^T x_i + b_D)]^T$

a diagonal matrix Λ with such k eigenvalues, one can get W_k by computing $W_k = U \Lambda U^T$. This means that instead of returning $\tilde{G}_k$, one should return $Z = (CU\Lambda^{1/2})^T$ when adapting Nystrom method to form SpectralNystrom (a version of Algorithm 4).

In terms of computational complexity, the standard singular value calculation in SpectralSketches takes $O(nk^2)$ (with $k = D$ being the number of random features), while the standard eigenvalue calculation takes $O(n^3)$. One may use Lanczos to do an approximate calculation to speed up the latter, which would still take $O(n^2 k)$ (with k being the number of iterations in Lanczos).

As one can see, SpectralSketches is embarrassingly simple and easy to apply using any existing sketching algorithm. Hence one can take full advantage of the continuous improvements in sketching techniques. The approximation accuracy and convergence rate is consistent with the performance of the chosen sketching algorithm as one will see in the theory and experiment sections. In those later sections, I will exemplify SpectralSketches with SpectralRFF and SpectralNystrom, where the sketching method used in Algorithm 4 are RFF and Nystrom, respectively.

7.4 Theoretical Guarantees of SpectralSketches

7.4.1 Notations

$s_i(A)$'s or $\sigma_i(A)$'s are used to denote singular values of matrix A, and $\lambda_j(K)$'s to denote eigenvalues of a symmetric matrix K. Furthermore, without loss of generality, I assume that the singular values are ordered,

Algorithm 6 the Nystrom Method

Input: $n \times n$ matrix G, $\{p_i\}_{i=1}^n$ such that $\sum_{i=1}^n p_i = 1$, $c \leq n$ and $k \leq c$
- Pick c columns of G in i.i.d. trials with replace and with respect to the probabilities $\{p_i\}_{i=1}^n$; let I be the set of indices of the sampled columns
for each sampled column (whose index is $i \in I$) **do**
Scale by dividing its elements by $\sqrt{cp_i}$
end for
- Let C be the $n \times c$ matrix containing the sampled columns rescaled in this manner and let W be the $c \times c$ submatrix of G whose entries are $G_{ij}/(c\sqrt{p_i p_j})$, $i \in I, j \in I$
- Compute W_k, the best rank-k approximation to W.
return: $\tilde{G}_k = CW_k^+ C^T$

with $\lambda_1(A)$ (resp. $\sigma_1(A)$) referring to the largest eigenvalue (resp. singular value) of A. Let Z be a $D \times n$ sketching matrix (Eg. consisting of random fourier features) throughout this paper. Since $\|\cdot\|_2 = s_1(\cdot) = s_{\max}(\cdot)$, we will use them interchangeably. $\|\cdot\|$ refer to the spectral norm and $\|\cdot\|_F$ refer to the Frobenius norm throughout this paper.

Note that for symmetric matrices, largest singular value is the absolute value of largest eigenvalue. The spectral radius is the maximum of absolute values of eigenvalues. So these terms may also be used interchangeably throughout this chapter.

7.4.2 Main Theorem

From previous analysis on RFF, one see that $Z^T Z$ approximates K by Lemma 2.2.1. Hence, $E = K - Z^T Z$ will be written as the error matrix.

Theorem 7.4.1 (SpectralRFF bound)**.** *Let M be a compact subset of R^d with diameter $diam(M)$. Let $x \in M$ for all row x of X with size $n \times d$. Let $K \in M^{n \times n}$ be the Gaussian kernel matrix of X with $k(x,y) = \exp(-\gamma\|x-y\|^2)$ and $Z \in M^{D \times n}$ be the matrix consisting of the random Fourier features of K. With probability at least*

$$1 - O(n\exp(-1/n)) \tag{7.1}$$

in terms of n, one has

$$\left|\|K\|_2 - s_1^2(Z)\right| \leq \varepsilon. \tag{7.2}$$

Proof. By Lemma 2.1.5 and Lemma 2.1.2,

$$\max_{1\leq k\leq n} |\lambda_k(K) - \lambda_k(Z^T Z)| \tag{7.3}$$

$$= \max_{1\leq k\leq n} |\lambda_k(K) - \lambda_k(K-E)| \tag{7.4}$$

$$\leq |\lambda_1(E)|. \tag{7.5}$$

From [RR07], one can know that $E[z(x)^T z(y)] = k(x,y)$ by design, hence $z(x)^T z(y) - k(x,y)$ is mean zero. Since the rows of Z are independent, then one can write the $n \times n$ matrix E as a sum of D independent mean-zero symmetric matrices: Let E_i $(1 \leq i \leq D)$ has its (j,l)-th entry as $Z_{ij}Z_{il} - \frac{K_{jl}}{D}$, then it is easy to verify that $E = \sum_{i=1}^{D} E_i$.

Apply Lemma 2.1.3 to E, one has

$$P[|\lambda_1(E)| \geq t] = P[|\lambda_1(\sum_{i=1}^{D} E_i)| \geq t] \leq 2n \exp(-\frac{t^2/2}{\sigma^2 + Ct/3}). \tag{7.6}$$

for every $t \geq 0$. By uniform convergence of RFF (Lemma 2.2.2), $E_{ij}^2 \leq \varepsilon^2$ with high probability, $\|E\| \leq \|E\|_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} E_{ij}^2} \leq n\varepsilon$, so $C = n\varepsilon$ with failure probability at most $2^{12}\frac{d^2\gamma^2 diam(M)^2}{\varepsilon^2}\exp(-\frac{D\varepsilon^2}{4(d+2)})$. Observe that $\sigma^2 \leq D\lambda_{\max}^2(E_i)$, because

$$\sigma^2 = \|\mathbb{E}\sum_{i=1}^{D} E_i^2\| = \|\sum_{i=1}^{D}\mathbb{E}(E_i^2)\| = \|D\mathbb{E}(E_1^2)\| \leq D\mathbb{E}\|E_1^2\| = D\lambda_{\max}^2(E_1), \tag{7.7}$$

where the first equality of by linearity of expectation, second by the fact that $\mathbb{E}(E_i^2)$ are equal for all i based on the i.i.d. assumption of random feature mappings, third by property of the norm, the inequality is by Jensen's inequality and that spectral norms are convex, and the fifth equality by the definition of spectral norm.

To summarize,

$$\begin{aligned}
P[|\lambda_1(E)| \geq \varepsilon] &\leq 2n\exp(-\frac{\varepsilon^2/2}{\sigma^2 + C\varepsilon/3}) \\
&\leq 2n\exp(-\frac{\varepsilon^2}{D\lambda_{\max}^2(E_1) + n\varepsilon^2/3}) + 2^{12}\frac{d^2\gamma^2 diam(M)^2}{\varepsilon^2}\exp(-\frac{D\varepsilon^2}{4(d+2)}) \\
&\leq 2n\exp(-\frac{\varepsilon^2/2}{D\varepsilon^2 + n\varepsilon^2/3}) + 2^{12}\frac{d^2\gamma^2 diam(M)^2}{\varepsilon^2}\exp(-\frac{D\varepsilon^2}{4(d+2)}) + 2\exp(-\varepsilon^2/4) \\
&= 2n\exp(-\frac{3}{6D+2n}) + 2^{12}\frac{d^2\gamma^2 diam(M)^2}{\varepsilon^2}\exp(-\frac{D\varepsilon^2}{4(d+2)}) + 2\exp(-\varepsilon^2/4),
\end{aligned} \tag{7.8}$$

where the first comes from setting $t = \varepsilon$ in Equation 7.6, the second from Equation 7.7, and the third holds by Lemma 2.2.1 with $D = 1$. Note the last two lines have no effect on n, so one can in fact stop at line 2 if ε is not of interest. □

One can reach a similar bound for Nystrom. Note the assumption about a in the next theorem holds unless $\varepsilon = \frac{2\sum_{i=1}^{n} K_{ii}^2}{\sqrt{c}}$ which is almost impossible because $\varepsilon > 0$ is small. Hence such assumption is an extremely weak one.

Theorem 7.4.2 (SpectralNystrom bound)**.** *Let M be a compact subset of R^d with diameter diam(M). Let $x \in M$ for all row x of X with size $n \times d$. Let $Z \in M^{c \times n}$ be the matrix consisting of the Nystrom-transformed data of X with respect to the Gaussian kernel $k(x,y) = \exp(-\gamma\|x-y\|^2)$. Assume $|\frac{\sqrt{c}}{2\sum_{i=1}^{n} K_{ii}^2} - \frac{1}{\varepsilon}| \geq a$ for some $a > 0$. With probability at least $1 - O(n\exp(-1/n))$,*

$$\left|\|K\|_2 - s_1^2(Z)\right| \leq \varepsilon, \tag{7.9}$$

where σ^2 is the variance of Nystrom approximation and c is the number of columns sampled.

Proof. From Lemma 2.2.3 with $k = r := rank(K)$ and taking $c = \frac{4(1+\sqrt{8\log(1/\delta)})^2}{\varepsilon^2}$, one can have

$$P[\|K - \tilde{K}\| \geq \varepsilon] \leq \exp(-\frac{(\frac{\varepsilon\sqrt{c}}{2\sum_{i=1}^{n} K_{ii}^2} - 1)^2}{8}). \tag{7.10}$$

From the implications of Lemma 2.1.4, one can have

$$\begin{aligned} P[|g(x)^T g(y) - k(x,y)| \geq \varepsilon] \leq \\ \leq P[\|K - \tilde{K}\| \geq \varepsilon] \\ \leq \exp(-\frac{(\frac{\varepsilon\sqrt{c}}{2\sum_{i=1}^{n} K_{ii}^2} - 1)^2}{8}). \end{aligned} \tag{7.11}$$

By assumption on a, one can have

$$P[|g(x)^T g(y) - k(x,y)| \geq \varepsilon] \leq \exp(-\frac{a\varepsilon^2}{8}). \tag{7.12}$$

Observe that this is similar to what one has in Lemma 2.2.1. Apply argument similar to what is used to obtain Lemma 2.2.2 from Lemma 2.2.1, one can get a similar uniform convergence bound on E. With a similar argument for norm bound using Frobenius norm to bound spectral norm, one can have $\|E\| \leq n\varepsilon$ with failure probability $\frac{d^2}{\varepsilon^2}\exp(-\frac{a\varepsilon^2}{d})$. Now apply Lemma 2.1.3 similarly,

$$\begin{aligned} P[|\lambda_1(E)| \geq \varepsilon] = P[|\lambda_1(\sum_{i=1}^{D} E_i)| \geq \varepsilon] \leq 2n\exp(-\frac{3}{6a+2n}) \\ + 2^{12}\frac{d^2\gamma^2 diam(M)^2}{\varepsilon^2}\exp(-\frac{a\varepsilon^2}{4(d+2)}) + 2\exp(-\varepsilon^2/4). \end{aligned} \tag{7.13}$$

□

Note that the two bounds have the same order with a in SpectralNystrom and D in SpectralRFF. I will use a in Theorem 7.4.4, but note the same is

true for D. Based on our assumption on a, with fixed number of columns sampled c, $a = O(1/\varepsilon)$. With fixed error budget ε, $a = O(\sqrt{c})$.

Observe that with respect to error budget ε, the failure probability is $O(\frac{\exp(-\varepsilon^2)}{\varepsilon^2})$. With respect to sample size n, the failure probability is $O(n\exp(-1/n))$. With respect to the feature dimension d of original data matrix, the failure probability is $O(d^2\exp(-1/d))$.

7.4.3 Comparison to Power Iteration and Lanczos Algorithm

On a high level, the following theorem states that the probablistic relative failure goes to zero roughly as $\sqrt{n}(1-\varepsilon)^k$ for the power algorithm and at most as $\sqrt{n}\exp(-(2k-1)\sqrt{\varepsilon})$ for the Lanczos algorithm [KW92]. It assumes the starting vectors are sampled uniformly from the unit sphere.

Theorem 7.4.3 (Theorem 4.1(a) [KW92])**.** *For any symmetric positive definite matrix A and for any $k \geq 2$ one can have for the power method*

$$P[\frac{\big|\|\tilde{A}\| - \|A\|\big|}{\|A\|} > \varepsilon] \tag{7.14}$$

$$\leq \min(0.824, \frac{0.354}{\sqrt{\varepsilon(k-1)}})\sqrt{n}(1-\varepsilon)^{k-1/2}. \tag{7.15}$$

and for the Lanczos algorithm,

$$P[\frac{\big|\|\tilde{A}\| - \|A\|\big|}{\|A\|} > \varepsilon] \leq 1.648\sqrt{n}\exp(-\sqrt{\varepsilon}(2k-1)). \tag{7.16}$$

Theorem 7.4.4 (probabilistic relative failure for SpectralSketches)**.** *For any symmetric positive definite matrix K,*

$$P[\frac{\big|\|\tilde{K}\| - \|K\|\big|}{\|K\|} \geq \varepsilon] \leq \exp(-\frac{a\varepsilon^2}{8r\|K\|}), \tag{7.17}$$

where $r \leq n$ is the rank of K and a is as defined in Theorem 7.4.2.

Proof. $\sum_{i=1}^n K_{ii}^2 \leq \sum_{i=1}^n \sum_{j=1}^n K_{ij}^2 = \|K\|_F^2$. It is well-known that $\|K\|_2 \leq \|K\|_F \leq \sqrt{r}\|K\|_2$, so from Theorem 7.4.2, one can have

$$P[\frac{\|\tilde{K} - K\|}{\|K\|} \geq \varepsilon r\|K\|] \leq \exp(-\frac{(\frac{\varepsilon\sqrt{c}}{2\|K\|_F^2} - 1)^2}{8}). \tag{7.18}$$

By the triangle inequality and reorganizing,

$$P[\frac{\big|\|\tilde{K}\| - \|K\|\big|}{\|K\|} \geq \varepsilon] \leq \exp(-\frac{(\frac{\varepsilon\sqrt{c}}{2r\|K\|\|K\|_F^2} - 1)^2}{8}). \tag{7.19}$$

Write $x = \frac{\varepsilon\sqrt{c}}{2r\|K\|\|K\|_F^2}$ and observe that $\frac{\varepsilon}{\sqrt{r}\|K\|_F} \leq \frac{3}{4} \leq (\frac{1}{2} - x)^2 + \frac{3}{4}$ because K has all 1's on the diagonal for the Gaussian kernel which makes $\|K\|_F \geq \sqrt{n}$ by definition of Frobenius norm. Then

$$\begin{aligned} \frac{\varepsilon}{r\|K\|} \leq \frac{\varepsilon}{\sqrt{r}\|K\|_F} &\leq (\frac{1}{2} - x)^2 + \frac{3}{4} \\ &= x^2 - x + 1 \\ \frac{\varepsilon}{r\|K\|} - x &\leq x^2 - 2x + 1 = (x-1)^2 \\ \frac{\varepsilon}{r\|K\|} - \frac{\varepsilon\sqrt{c}}{2r\|K\|\|K\|_F^2} &\leq (\frac{\varepsilon\sqrt{c}}{2r\|K\|\|K\|_F^2} - 1)^2. \end{aligned} \tag{7.20}$$

Furthermore, by assumption on a

$$\begin{aligned} a \leq 1 - \frac{\varepsilon\sqrt{c}}{2\sum_{i=1}^n K_{ii}^2} &\leq 1 - \frac{\varepsilon\sqrt{c}}{2\|K\|_F^2}, \\ \frac{a\varepsilon^2}{r\|K\|} \leq \frac{a\varepsilon}{r\|K\|} &\leq \frac{\varepsilon}{r\|K\|} - \frac{\varepsilon\sqrt{c}}{2r\|K\|\|K\|_F^2}. \end{aligned} \tag{7.21}$$

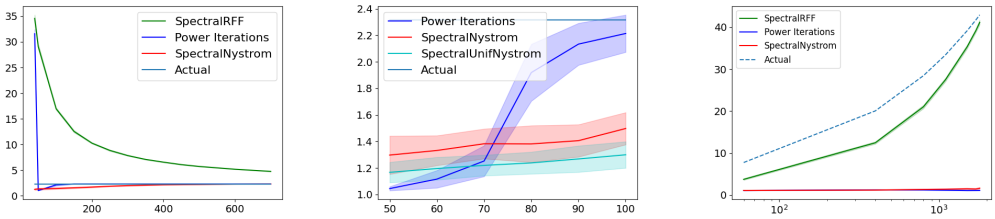
By combining equation 7.20 and equation 7.21, one can easily get

$$\frac{a\varepsilon^2}{r\|K\|} \leq (\frac{\varepsilon\sqrt{c}}{2r\|K\|\|K\|_F^2} - 1)^2.$$

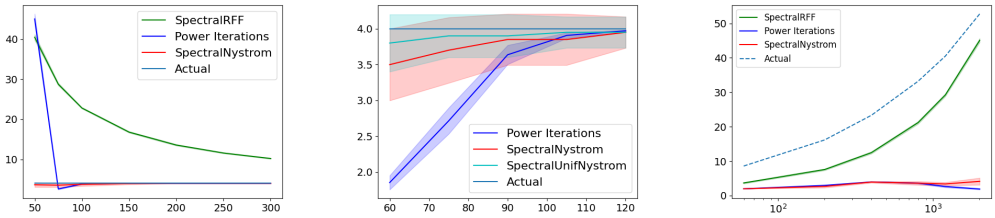
Combining this with equation 7.19 gives the theorem. □

By comparing these bounds, one can see that SpectralSketches scale better (as n increases) than both power method and Lanczos algorithm as n gets large. In particular, when matrix norm is fixed, the failure probability is of order $O(\exp(-1/r))$ with r being the matrix rank. When matrix norm is not fixed, it is $O(\exp(-\frac{1}{r\|K\|}))$. When n is fixed, however, SpectralSketches may have a slower convergence speed than iteration-based methods, since ε is squared in the exponential term of the failure probability bound. This is consistent with my experimental result in Figure 7.1, see thorough discussion in Section 7.5.

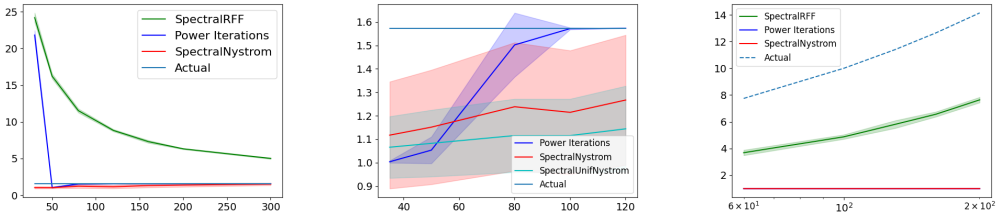
Finally, there exists seemingly faster algorithm and seemingly more superior bound in modern literature, Eg. [MM15]. However, their algorithm is fast for sparse matrix and still takes $O(n^2k)$ for non-sparse matrix while SpectralSketches takes $O(nk^2)$. Their bound is better than what's shown in Theorem 7.4.3 in terms of ε but not n. SpectralSketches may have slower convergence than Lanczos-type algorithms in the fixed-small-n and large-number-of-iterations (large k) case, but it is a better choice for small k and large n (i.e. when scalability is required and a only small computational budget is available). Further empirical evidence of this fact is shown in Section 7.5, consistent with our analysis in the previous paragraph.



(a) Estimates vs. k on Crime (b) Estimates vs. k on Crime (c) Estimates vs. n on Crime



(d) Estimates vs. k on Wine (e) Estimates vs. k on Wine (f) Estimates vs. n on Wine



(g) Estimates vs. k on Boston (h) Estimates vs. k on Boston (i) Estimates vs. n on Boston

Figure 7.1: Approximation Performance of SpectralSketches and Power Method on Three Datasets over 20 Trials. The first two columns vary the number of features or number of iterations, keeping the overall time complexity $O(nk^2)$ across all three methods. The third column vary the number of samples used while keeping $k = 60$ constant across all three datasets. The first column is a zoomed-out version, which gives intuition about the convergence speed for each method while sample/matrix size is kept the same. The second column is a zoomed-in version, which enables finer comparison between the data-dependent methods. The third column gives intuition about scalability - we see that SpectralRFF consistently outperforms the Power Method, which is consistent with our theoretical finding in Section 7.4.3 (also discussed in Section 7.5).

7.5 Experiments

For fair comparison, we compare each class of methods in their primitive form. Namely, we experiment on power method as an iteration-based repre-

sentative, SpectralSketches for random fourier feature (SpectralRFF) as a data-independent representative, and SpectralSketches for Nystrom method (SpectralNystrom) as a data-dependent sampling-based representative. As mentioned towards the end of Section 7.1, all three methods has seen numerous improvements over the years, as discussed in Section 7.2.2, however, those improvements either suffer from the same drawbacks as the original method or trade time or space complexity to address those drawbacks. While it might be worthwhile to research incremental improvements following roughly the same pattern, it is also critical to think outside the box and recognize new class of methods for solving the same problem. Our goal is to point out SpectralSketches as a strong alternative to existing research which all seems to follow the iteration-based pattern throughout decades.

To this end, we show performance of our example SpectralSketches algorithms, SpectralRFF and SpectralNystrom, on three public real-world datasets (Figure 7.1). The three columns in Figure 7.1 give us intuition about overall convergence speed (a)(d)(g), approximation quality under different computation budget (b)(e)(h), and scalability (c)(f)(i). To clarify, we state the power iterations algorithm used for producing Figure 7.1 in Algorithm 7. The exact procedures of SpectralRFF and SpectralNystrom is discussed in Section 7.3, so we won't repeat them here. In Figure 7.1 (b)(e)(h), we also included SpectralUnifNystrom as a benchmark. UnifNystrom is almost the Nystrom algorithm described in Algorithm 6 but without the leverage score sampling. In other words, force $p_i = 1/n$ for all i, meaning sample the columns uniformly. The reason for including SpectralUnifNystrom is not only to demonstrate the flexibility of SpectralSketches and show the performance difference, but also to try and include existing widely used packages whenever possible. Namely, the package we used for SpectralUnifNystrom is `from sklearn.kernel_approximation import Nystroem`.

The datasets we used are the UCI Communities and Crime dataset, the UCI Wine Quality (White) Dataset [DG17], and the Kaggle Boston Housing dataset. We generate symmetric matrices from these datasets by implicitly (for RFF) or explicitly (for all other methods) applying the Gaussian kernel. The spectral norms of such matrices are of tremendous importance for various applications as discussed in Section 7.2. For Figure 7.1 (a)(b), the matrix evaluated is of size 996×996, for Figure 7.1 (d)(e), 1469×1469 and for Figure 7.1 (g)(h), 479×479. The matrix size changes as n changes in Figure 7.1 (c)(f)(i), with its size being $n \times n$ at all times.

Overall, individual SpectralSketches algorithms inherit strength and weaknesses from the chosen Sketching algorithm. Next, we give a finer discussion on all three aspects.

Algorithm 7 Power Iterations

Input: number of iterations t, number of features k, a distribution p for guessed starting vectors, $n \times n$ matrix K
for i = 1 to t **do**
  Sample starting vector v from p
  for j = 1 to k **do**
    Compute Kv and normalize the resulting vector, update v with the vector and update the norm estimate with the max absolute value of entry in Kv
  end for
  Output an estimated norm
end for

7.5.1 Convergence Speed

RFF is known to have slow convergence speed, and a string of research papers have tried to address that over the years [YSC+16, Bac17, DDSR17, MKBO18, YSSK20] as discussed in Section 7.2. Since RFF is known to have slower convergence than Nystrom, it is expected that such slow convergence behavior gets inherited in SpectralRFF as shown in Figure 7.1(a)(d)(g). As observed, SpectralNystrom steadily converges to the actual value as number of features increases, which corroborates our theory in Theorem 7.4.2.

7.5.2 Approximation Quality

Figure 7.1(b)(e)(h) is a zoomed-in version of the previous column. The difference between data-dependent and data-independent methods is again observed - SpectralNystrom give closer estimate than SpectralRFF in the fixed sample size setting due to its ability to incorporate information from data. Also as expected, SpectralNystrom has similar and usually slightly better approximation than SpectralUnifNystrom, since these algorithms are extremely alike in nature with UnifNystrom being more coarse than leverage-scored Nystrom. It appears that Spectral(Unif)Nystrom consistently give better estimate than Power method when the computational budget is low (k is small) despite the slightly larger variance. When k is even smaller than what's shown in Figure 7.1(b)(e)(h), we see that SpectralNystrom is obviously the winner by referring back to Figure 7.1(a)(d)(g).

To summarize, SpectralNystrom has strong advantages over Power method in the low budget regime. It consistently provides more stable and more accurate approximations.

7.5.3 Scalability

As indicated by our theoretical analysis in Section 7.4.3, SpectralSketches has better scalability gaurantee than Power method and Lanczos algorithm. This is real world data so we don't have fixed matrix norm, which means the failure probability should scale as $O(\exp(-\frac{1}{r\|K\|}))$. This is exactly what our experimental results show in Figure 7.1 (c)(f)(i). With a given computational budget and increased data size (i.e. $k = 60$ fixed and n increases for the $n \times n$ matrix), SpectralRFF scales a lot better than all other methods. (The performance gap between SpectralRFF and SpectralNystrom might be due to our assumption on a. Even though their bound have the same order substituting D with a, the D in SpectralRFF is fixed to $k = 60$ while a depends on the matrix norm which is not fixed in this case.) Note that $k = 60$ is a extremely low budget, especially considering n grows into hundreds and thousands for the matrix size n^2 in Figure 7.1 (c)(f)(i). This means instead of dealing with matrices of size $n \times n$ with $n = 996, 1469, 479$ respectively in Figure 7.1 (c)(f)(i), we only need to store matrix of size $n \times 60$, and the time complexity for estimating the spectral norm is reduced from $O(n^3)$ to $O(60^2 n)$. Even though it may seem like the estimates are not as accurate as need be in Figure 7.1 (c)(f)(i), keep in mind that this is simply because we chose a small budget $k = 60$. One can easily get more accurate approximation by extending the budget, as suggested by the convergence behavior in other plots of Figure 7.1. Furthermore, we don't always need spectral norm estimate to be very exact in practice – a lot of times it is enough to know the correct order of magnitude that spectral norm has, for example when spectral norm is used as a regularization term [AMP10, SCS+15, BMCM19, ZTSG19] or as a trainability signal in neural architecture search [XPS20, CGW21].

7.6 Conclusion

This chapter presents SpectralSketches, a framework for approximating spectral norm through any sketching-based algorithms. I give SpectralRFF and SpectralNystrom as example algorithms of SpectralSketches and perform theoretical and empirical analysis on them. I derive rigorous theoretical guarantee for the approximation quality of SpectralSketches and show that it scales better than the current mainstream methods (Power method and Lanczos algorithm, both iteration-based). In particular, for a fixed error budget ε, the probablistic relative failure is reduced from $O(\sqrt{n})$ to $O(\exp(-1/r))$ with $r \leq n$ being the matrix rank. I corroborate such superior scalability through experiments on three real-world datasets. Empirical evidence further suggest that SpectralSketches give a more numerically stable and more accurate estimate in the low (computational) budget

regime. While current research on spectral norm estimation mostly follow the iteration-based paradigm, SpectralSketches provides a fundamentally different alternative to spectral norm approximation with strong theoretical guarantee and empirical performance.

Part III

Limitations and Future Directions

Chapter 8

Limitations and Future Research Directions

8.1 Fair Machine Learning

While the insight from [CL22b] that suggests separate standardization of features between groups is interesting, the ethical implications of doing so is not thoroughly discussed. It is worth taking a closer look at when it would make sense to standardize data separately and when it would not. For example, when such method is applied to social settings that affect decision-making involving people, it should be used with caution, whereas under the setting when fairness is applied to objects instead, for example, new versus old content in a recommendation system, then the ethical concern is minuscule. The relative distribution shift that results from separate standardization is not quantified in [CL22b], but it may be worth exploring in future research.

Additionally, [CL22a, CL22b] both discuss achieving individual fairness when the sensitive attribute is available. It is worth researching the group fairness aspect and explore alternatives when the sensitive attribute may not be available.

8.2 Randomized Algorithms in Machine Learning

[CL22c] gives an interesting alternative to traditional ensemble method and proves the possibility of removing the weakly-learned assumption for base models. However, there is still a gap between the theoretical analysis and empirical observations. Namely, the empirical results seems to converge much faster than the theoretical guarantee, suggesting the current bound is not sharp. Obtaining a sharp bound likely require a fundamentally different

way of modeling, which is still an open problem at the time of this writing. Alternatively, sharper bounds may exist under more granular settings, for example, by enforcing certain assumption on data. Currently, analysis in [CL22c] make no such assumptions.

For further applications of [CL22c], two questions are worth answering. First, how to generate hypothesis in practice to obtain uniform direction when those hypothesis are applied to data. It is an important assumption in the analysis, and currently it is done through making sure the resulting vector coordinates have equal probability of being positive or negative. That means each vector has equal probability of falling into any of the high-dimensional quadrant and hence the directions are almost uniform. For neural networks it might be possible to do something similar, but the exact empirical performance of doing that is unknown. Second, whether the inherent regularization from such randomized framework can be exploited to achieve federated learning, differential privacy and robustness. Randomized algorithms are typically easy to parallelize and so is RHSS. Hence it might have natural applications in federated learning. Modern research on differential privacy and robustness usually involves adding a small random noise. Generating a random vector instead with RHSS might provide a new paradigm and alternative to the existing solutions in these areas.

Bibliography

[AAK+20] Jacob Abernethy, Pranjal Awasthi, Matthäus Kleindessner, Jamie Morgenstern, Chris Russell, and Jie Zhang. Active sampling for min-max fairness. *arXiv preprint arXiv:2006.06879*, 2020.

[AAT20] Hadis Anahideh, Abolfazl Asudeh, and Saravanan Thirumuruganathan. Fair active learning. *arXiv preprint arXiv:2001.01796*, 2020.

[AC06] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 557–563, 2006.

[AIK18] Daniel Alabi, Nicole Immorlica, and Adam Kalai. Unleashing linear optimizers for group-fair learning and optimization. In *Conference On Learning Theory*, pages 2043–2066. PMLR, 2018.

[AKG+14] Charu C Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and S Yu Philip. Active learning: A survey. In *Data Classification*, pages 599–634. Chapman and Hall/CRC, 2014.

[AKM+17] Haim Avron, Michael Kapralov, Cameron Musco, Christopher Musco, Ameya Velingker, and Amir Zandieh. Random fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *International Conference on Machine Learning*, pages 253–262. PMLR, 2017.

[AM05] Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *International Conference on Computational Learning Theory*, pages 458–469. Springer, 2005.

[AMP10] Andreas Argyriou, Charles A Micchelli, and Massimiliano Pontil. On spectral learning. *Journal of Machine Learning Research*, 11(2), 2010.

[APYM07] Andreas Argyriou, Massimiliano Pontil, Yiming Ying, and Charles Micchelli. A spectral regularization framework for multi-task structure learning. *Advances in neural information processing systems*, 20, 2007.

[AV06] Rosa I Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine learning*, 63(2):161–182, 2006.

[AZL06] Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 504–509, 2006.

[Bac17] Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017.

[BBL09] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[BFT17] Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

[BHV10] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine learning*, 80(2):111–139, 2010.

[BJW20] Yahav Bechavod, Christopher Jung, and Steven Z Wu. Metric-free individual fairness in online learning. *Advances in neural information processing systems*, 33:11214–11225, 2020.

[BL19] Ananth Balashankar and Alyssa Lees. Fairness sample complexity and the case for human intervention. *arXiv preprint arXiv:1910.11452*, 2019.

[BM02] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[BMCM19] Alberto Bietti, Grégoire Mialon, Dexiong Chen, and Julien Mairal. A kernel perspective for regularizing deep neural

networks. In *International Conference on Machine Learning*, pages 664–674. PMLR, 2019.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[BRK07] Robert Burbidge, Jem J Rowland, and Ross D King. Active learning for regression based on query by committee. In *International conference on intelligent data engineering and automated learning*, pages 209–218. Springer, 2007.

[BS16] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 2016.

[CGW21] Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. *arXiv preprint arXiv:2102.11535*, 2021.

[CL22a] Yiting Cao and Chao Lan. Active approximately metric-fair learning. In *Uncertainty in Artificial Intelligence*, pages 275–285. PMLR, 2022.

[CL22b] Yiting Cao and Chao Lan. Fairness-aware active learning for decoupled model. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–9. IEEE, 2022.

[CL22c] Yiting Cao and Chao Lan. A model-agnostic randomized learning framework based on random hypothesis subspace sampling. In *International Conference on Machine Learning*, pages 2597–2608. PMLR, 2022.

[CR18] Alexandra Chouldechova and Aaron Roth. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*, 2018.

[CW18] Jason Chan and Jing Wang. Hiring preferences in online labor markets: Evidence of a female hiring bias. *Management Science*, 64(7):2973–2994, 2018.

[CZJY21] Yifan Chen, Qi Zeng, Heng Ji, and Yun Yang. Skyformer: Remodel self-attention with gaussian kernel and nystr\" om method. *Advances in Neural Information Processing Systems*, 34:2122–2135, 2021.

[CZZ13] Wenbin Cai, Ya Zhang, and Jun Zhou. Maximizing expected model change for active learning in regression. In *2013 IEEE 13th international conference on data mining*, pages 51–60. IEEE, 2013.

[Das05] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. *Advances in neural information processing systems*, 18, 2005.

[data]

[datb]

[DDSR17] Tri Dao, Christopher M De Sa, and Christopher Ré. Gaussian quadrature for kernel features. *Advances in neural information processing systems*, 30, 2017.

[DG17] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[DHM07] Sanjoy Dasgupta, Daniel J Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. *Advances in neural information processing systems*, 20, 2007.

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[DIKL18] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on fairness, accountability and transparency*, pages 119–133. PMLR, 2018.

[DK13] Robert Durrant and Ata Kabán. Sharp generalization error bounds for randomly-projected classifiers. In *International Conference on Machine Learning*, pages 693–701, 2013.

[DKM06] Petros Drineas, Ravi Kannan, and Michael W Mahoney. Fast monte carlo algorithms for matrices i: Approximating matrix multiplication. *SIAM Journal on Computing*, 36(1):132–157, 2006.

[DKM20] Michal Derezinski, Rajiv Khanna, and Michael W Mahoney. Improved guarantees and a multiple-descent curve for column subset selection and the nystrom method. *Advances in Neural Information Processing Systems*, 33:4953–4964, 2020.

[DL21] Tu Do and Ngoc Hoang Luong. Training-free multi-objective evolutionary neural architecture search via neural tangent kernel and number of linear regions. In *International Conference on Neural Information Processing*, pages 335–347. Springer, 2021.

[DM16] Petros Drineas and Michael W Mahoney. Randnla: randomized numerical linear algebra. *Communications of the ACM*, 59(6):80–90, 2016.

[DM18] Petros Drineas and Michael W Mahoney. Lectures on randomized numerical linear algebra. *The Mathematics of Data*, 25(1), 2018.

[DMC05] Petros Drineas, Michael W Mahoney, and Nello Cristianini. On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(12), 2005.

[DMMS11] Petros Drineas, Michael W Mahoney, Shan Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische mathematik*, 117(2):219–249, 2011.

[DMR18] Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional gaussians. *arXiv preprint arXiv:1810.08693*, 2018.

[DPH+12] Chesner Desir, Caroline Petitjean, Laurent Heutte, Mathieu Salaun, and Luc Thiberville. Classification of endomicroscopic images of the lung based on random subwindows and extra-trees. *IEEE Transactions on Biomedical Engineering*, 59(9):2677–2683, 2012.

[FM03] Dmitriy Fradkin and David Madigan. Experiments with random projections for machine learning. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–522, 2003.

[FPCDG16] Avi Feller, Emma Pierson, Sam Corbett-Davies, and Sharad Goel. A computer program used for bail and sentencing decisions was labeled biased against blacks. it's actually not that clear. *The Washington Post*, 17, 2016.

[GEW06] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[GGVS+19] Sahil Garg, Aram Galstyan, Greg Ver Steeg, Irina Rish, Guillermo Cecchi, and Shuyang Gao. Kernelized hashcode representations for relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6431–6440, 2019.

[GGVSC19] Sahil Garg, Aram Galstyan, Greg Ver Steeg, and Guillermo A Cecchi. Nearly-unsupervised hashcode representations for biomedical relation extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4026–4036, 2019.

[GHZGW16] Nina Grgic-Hlaca, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. The case for process fairness in learning: Feature selection for fair decision making. In *NIPS symposium on machine learning and the law*, volume 1, page 2, 2016.

[GHZGW18] Nina Grgić-Hlača, Muhammad Bilal Zafar, Krishna P Gummadi, and Adrian Weller. Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Git11] A Gittens. The spectral norm error of the naıve nyström extension. Technical report, Technical report, California Institute of Technology, 2012. Preprint, 2011.

[GK65] Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

[GM13] Alex Gittens and Michael Mahoney. Revisiting the nystrom method for improved large-scale machine learning. In *International Conference on Machine Learning*, pages 567–575. PMLR, 2013.

[GS20] Claudio Gallicchio and Simone Scardapane. Deep randomized neural networks. *Recent Trends in Learning From Data*, pages 43–68, 2020.

[GU77] Gene H Golub and Richard Underwood. The block lanczos method for computing eigenvalues. In *Mathematical software*, pages 361–377. Elsevier, 1977.

[H+14] Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

[Han07] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pages 353–360, 2007.

[HJZL06] Steven CH Hoi, Rong Jin, Jianke Zhu, and Michael R Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424, 2006.

[Ilv20] Christina Ilvento. Metric learning for individual fairness. In *1st Symposium on Foundations of Responsible Computing*, 2020.

[IP95] Boris Igelnik and Yoh-Han Pao. Stochastic choice of basis functions in adaptive function approximation and the functional-link net. *IEEE transactions on Neural Networks*, 6(6):1320–1329, 1995.

[JGH18] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[KK10] Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 299–308. IEEE, 2010.

[Kom03] Louis Komzsik. *The Lanczos method: evolution and application*. SIAM, 2003.

[KRR18] Michael Kim, Omer Reingold, and Guy Rothblum. Fairness through computationally-bounded awareness. *Advances in Neural Information Processing Systems*, 31, 2018.

[KW92] Jacek Kuczyński and Henryk Woźniakowski. Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. *SIAM journal on matrix analysis and applications*, 13(4):1094–1122, 1992.

[Lan50] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.

[Li17] Ping Li. Linearized gmm kernels and normalized random fourier features. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 315–324, 2017.

[Liu04] Ying Liu. Active learning with support vector machine applied to gene expression data for cancer classification. *Journal of chemical information and computer sciences*, 44(6):1936–1941, 2004.

[LQRI+22] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, Wenbin Zhang, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1452, 2022.

[LTOS19] Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards a unified analysis of random fourier features. In *International Conference on Machine Learning*, pages 3905–3914. PMLR, 2019.

[LWM+07] Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51):20167–20172, 2007.

[LWM18] Miles Lopes, Shusen Wang, and Michael Mahoney. Error estimation for randomized least-squares algorithms via the bootstrap. In *International Conference on Machine Learning*, pages 3217–3226. PMLR, 2018.

[MHT10] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

[MKBO18] Marina Munkhoeva, Yermek Kapushev, Evgeny Burnaev, and Ivan Oseledets. Quadrature-based features for kernel approximation. *Advances in neural information processing systems*, 31, 2018.

[MM12] Odalric Maillard and Rémi Munos. Linear regression with random projections. *Journal of Machine Learning Research*, 13:2735–2772, 2012.

[MM15] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. *Advances in neural information processing systems*, 28, 2015.

[MM17] Cameron Musco and Christopher Musco. Recursive sampling for the nystrom method. *Advances in neural information processing systems*, 30, 2017.

[MMS+21] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

[MOS20] Hussein Mozannar, Mesrob Ohannessian, and Nathan Srebro. Fair learning with private demographic data. In *International Conference on Machine Learning*, pages 7066–7075. PMLR, 2020.

[MR08] Mehryar Mohri and Afshin Rostamizadeh. Rademacher complexity bounds for non-iid processes. *Advances in Neural Information Processing Systems*, 21, 2008.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[MWvdG+15] Oskar Maier, Matthias Wilms, Janina von der Gablentz, Ulrike M Krämer, Thomas F Münte, and Heinz Handels. Extra tree forests for sub-acute ischemic stroke lesion segmentation in mr sequences. *Journal of neuroscience methods*, 240:89–100, 2015.

[MYBS20] Debarghya Mukherjee, Mikhail Yurochkin, Moulinath Banerjee, and Yuekai Sun. Two simple ways to learn individual fairness metrics from data. In *International Conference on Machine Learning*, pages 7097–7107. PMLR, 2020.

[NNSS20] Deanna Needell, Aaron A Nelson, Rayan Saab, and Palina Salanevich. Random vector functional link networks for function approximation on manifolds. *arXiv preprint arXiv:2007.15776*, 2020.

[Pea01] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[Pin12] Allan Pinkus. *N-widths in Approximation Theory*, volume 7. Springer Science & Business Media, 2012.

[PPS94] Yoh-Han Pao, Gwang-Hoon Park, and Dejan J Sobajic. Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing*, 6(2):163–180, 1994.

[PS22] Dana Pessach and Erez Shmueli. A review on fairness in machine learning. *ACM Computing Surveys (CSUR)*, 55(3):1–44, 2022.

[RCCR18] Alessandro Rudi, Daniele Calandriello, Luigi Carratino, and Lorenzo Rosasco. On fast leverage score sampling and optimal learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[RKH20] Kevin Roth, Yannic Kilcher, and Thomas Hofmann. Adversarial training is a form of data-dependent operator norm regularization. *Advances in Neural Information Processing Systems*, 33:14973–14985, 2020.

[RLHK51] JB Rosser, C Lanczos, MR Hestenes, and W Karush. Separation of close eigenvalues af a real symmetric matrix. *JOURNAL OF RESEARCH OF THE NATIONAL BUREAU OF STANDARDS*, 47(4):291–297, 1951.

[RR07] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[RR08] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[RR17] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. *Advances in neural information processing systems*, 30:3215–3225, 2017.

[RY18] Guy N Rothblum and Gal Yona. Probably approximately metric-fair learning. *arXiv preprint arXiv:1803.03242*, 5(2), 2018.

[SCM20] Eliran Shabat, Lee Cohen, and Yishay Mansour. Sample complexity of uniform convergence for multicalibration. *Advances in Neural Information Processing Systems*, 33:13331–13340, 2020.

[SCS+15] Rakesh Shivanna, Bibaswan K Chatterjee, Raman Sankaran, Chiranjib Bhattacharyya, and Francis Bach. Spectral norm regularization of orthonormal representations for graph transduction. *Advances in Neural Information Processing Systems*, 28, 2015.

[SDI20] Amr Sharaf and Hal Daumé III. Promoting fairness in learned models by learning to active learn under parity constraints. In *Workshop on Real World Experiment Design and Active Learning. International Conference on Machine Learning*, 2020.

[Set09] Burr Settles. Active learning literature survey. 2009.

[SFGJ21] Shubhanshu Shekhar, Greg Fields, Mohammad Ghavamzadeh, and Tara Javidi. Adaptive sampling for minimax fair classification. *Advances in Neural Information Processing Systems*, 34, 2021.

[SMKR19] Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. Average individual fairness: Algorithms, generalization and experiments. *Advances in Neural Information Processing Systems*, 32, 2019.

[SN09] Masashi Sugiyama and Shinichi Nakajima. Pool-based active learning in approximate linear regression. *Machine Learning*, 75(3):249–274, 2009.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Sza] S Szarek. Metric entropy of homogeneous spaces, quantum probability,(gdensk 1977).

[Sza82] Stanislaw J Szarek. Nets of grassmann manifold and orthogonal group. In *Proceedings of research workshop on Banach space theory (Iowa City, Iowa, 1981)*, volume 169, page 185, 1982.

[Sza15] Zoltán Szabó. Optimal rates for random fourier features. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 1144–1152. 2015.

[T+15] Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

[TBRS13] Martin Takác, Avleen Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for svms. In *International Conference on Machine Learning*, pages 1022–1030. PMLR, 2013.

[TCM99] Cynthia A Thompson, Mary Elaine Califf, and Raymond J Mooney. Active learning for natural language parsing and information extraction. In *ICML*, pages 406–414. Citeseer, 1999.

[Van91] R Vanderbei. Uniform continuity is almost lipschitz continuity. Technical report, Technical Report SOR-91-11, Statistics and Operations Research Series . . . , 1991.

[Vem05] Santosh S Vempala. *The random projection method*, volume 65. American Mathematical Soc., 2005.

[Ver10] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[Ver18] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[VSSB22] Sattar Vakili, Jonathan Scarlett, Da-shan Shiu, and Alberto Bernacchia. Improved convergence rates for sparse approximation methods in kernel-based learning. In *International Conference on Machine Learning*, pages 21960–21983. PMLR, 2022.

[W+14] David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

[WGM19] Shusen Wang, Alex Gittens, and Michael W Mahoney. Scalable kernel k-means clustering with nyström approximation: relative-error bounds. *The Journal of Machine Learning Research*, 20(1):431–479, 2019.

[Wil58] James Hardy Wilkinson. The calculation of eigenvectors by the method of lanczos. *The Computer Journal*, 1(3):148–152, 1958.

[WLH19] Dongrui Wu, Chin-Teng Lin, and Jian Huang. Active learning for regression using greedy sampling. *Information Sciences*, 474:90–105, 2019.

[WLR+03] Manfred K Warmuth, Jun Liao, Gunnar Rätsch, Michael Mathieson, Santosh Putta, and Christian Lemmen. Active learning with support vector machines in the drug discovery process. *Journal of chemical information and computer sciences*, 43(2):667–673, 2003.

[XPS20] Lechao Xiao, Jeffrey Pennington, and Samuel Schoenholz. Disentangling trainability and generalization in deep neural networks. In *International Conference on Machine Learning*, pages 10462–10472. PMLR, 2020.

[XZC+21] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14138–14148, 2021.

[YK10] Hwanjo Yu and Sungchul Kim. Passive sampling for regression. In *2010 IEEE International Conference on Data Mining*, pages 1151–1156. IEEE, 2010.

[YLM+12] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in neural information processing systems*, 25:476–484, 2012.

[YR18] Gal Yona and Guy Rothblum. Probably approximately metric-fair learning. In *International Conference on Machine Learning*, pages 5680–5688. PMLR, 2018.

[YSC+16] Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. *Advances in neural information processing systems*, 29, 2016.

[YSSK20] Hayata Yamasaki, Sathyawageeswar Subramanian, Sho Sonoda, and Masato Koashi. Learning with optimized random features: Exponential speedup by quantum machine learning without sparsity and low-rank assumptions. *Advances in Neural Information Processing Systems*, 33:13674–13687, 2020.

[ZH13] Peilin Zhao and Steven CH Hoi. Cost-sensitive online active learning with application to malicious url detection. In *Proceedings of the 19th ACM SIGKDD international conference*

on Knowledge discovery and data mining, pages 919–927, 2013.

[ZHX+21] Jiaru Zhang, Yang Hua, Zhengui Xue, Tao Song, Chengyu Zheng, Ruhui Ma, and Haibing Guan. Robust bayesian neural networks by spectral expectation bound regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3815–3824, 2021.

[ZS16] Le Zhang and Ponnuthurai N Suganthan. A comprehensive evaluation of random vector functional link networks. *Information sciences*, 367:1094–1105, 2016.

[ZTSG19] Han Zhao, Yao-Hung Hubert Tsai, Russ R Salakhutdinov, and Geoffrey J Gordon. Learning neural networks with adaptive regularization. *Advances in Neural Information Processing Systems*, 32, 2019.

[ZWS+13] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013.