Sharp Analysis of Random Fourier Features in Classification

Zhu Li

Gatsby Computational Neuroscience Unit, University College London
zhu.li@ucl.ac.uk

Abstract

We study the theoretical properties of random Fourier features classification with Lipschitz continuous loss functions such as support vector machine and logistic regression. Utilizing the regularity condition, we show for the first time that random Fourier features classification can achieve $O(1/\sqrt{n})$ learning rate with only $\Omega(\sqrt{n}\log n)$ features, as opposed to $\Omega(n)$ features suggested by previous results. Our study covers the standard feature sampling method for which we reduce the number of features required, as well as a problem-dependent sampling method which further reduces the number of features while still keeping the optimal generalization property. Moreover, we prove that the random Fourier features classification can obtain a fast $O(1/n)$ learning rate for both sampling schemes under Massart's low noise assumption. Our results demonstrate the potential effectiveness of random Fourier features approximation in reducing the computational complexity (roughly from $O(n^3)$ in time and $O(n^2)$ in space to $O(n^2)$ and $O(n\sqrt{n})$ respectively) without having to trade-off the statistical prediction accuracy. In addition, the achieved trade-off in our analysis is at least the same as the optimal results in the literature under the worst case scenario and significantly improves the optimal results under benign regularity conditions.

Introduction

Kernel methods have been widely used in many machine learning tasks such as regression and classification (Schölkopf and Smola 2001; Schölkopf, Tsuda, and Vert 2004), as they provide a simple framework to model highly complicated functional relationships and well-established theoretical guarantees (Caponnetto and De Vito 2007; Steinwart and Christmann 2008). The power of kernel methods comes from the so-called "kernel trick", where it utilizes a feature function $\phi(\cdot)$ to implicitly map the data into a high or possibly infinite dimensional feature space and thus allows non-linear functional learning. However, kernel methods are notorious for being time-consuming, since a typical kernel learning algorithm requires $O(n^3)$ computation and $O(n^2)$ memory, where n is the number of training samples. Due to the prohibitive computational requirements, a flurry of research has been devoted to developing algorithms that efficiently approximate kernel functions (Smola and Schökopf 2000; Williams and Seeger 2001a; Rahimi and Recht 2007; Mahoney and Drineas 2009; Alaoui and Mahoney 2015; Rudi, Carratino, and Rosasco 2017; Zhang, Duchi, and Wainwright 2015).

Among many approximation frameworks, the random Fourier features (RFFs) method proposed by (Rahimi and Recht 2007) has received great attention recently (see Liu et al. 2020, for a comprehensive review). The key idea of RFFs is to approximate the infinite dimensional feature map $\phi(\cdot)$ with an explicit s-dimensional random feature map $\phi_s(\cdot)$ through Bochner's theorem (Bochner 1932; Rudin 2017), which states that $\phi_s(\cdot)$ can be constructed through sampling from some spectral measure. Kernel methods are now reduced to linear learning in the feature space, which can be computed via fast linear solver (Shalev-Shwartz et al. 2011). The computational cost decreases from roughly $O(n^3)$ in time and $O(n^2)$ in space to $O(ns^2)$ and $O(ns)$ respectively. As a result, significant computational savings can be achieved as long as $s \ll n$.

Despite their empirical success (Rahimi and Recht 2007; Huang et al. 2014; Dai et al. 2014), theoretical understanding of the RFFs is incomplete. In particular, the question of how to choose s in order to obtain the RFFs estimators with performance provably comparable to original kernel methods remains unclear. To this end, several authors study the properties of the RFFs to approximate the kernel function and the kernel Gram-matrix (see e.g., Rahimi and Recht 2007; Sriperumbudur and Szabó 2015; Sutherland and Schneider 2015, and references therein). However, all of these works require $s = \Omega(n)$ features to guarantee no loss of prediction accuracy, which translates to no computational savings at all. A highly refined analysis in the context of ridge regression is proposed recently (see e.g., Rudi and Rosasco 2017; Avron et al. 2017; Li et al. 2021). When the spectral measure is used for sampling, they first show that $O(\sqrt{n}\log n)$ features are adequate to guarantee the minimax optimal learning rate $O(1/\sqrt{n})$, the same learning rate obtained with full kernel ridge regression. Furthermore, they prove that the RFFs regression can obtain a fast learning rate at the expense of increasing the number of features. Finally, they demonstrate that using a problem-dependent sampling distribution can significantly reduce the number of features to $s = \Omega(1)$.

A question motivating our study is whether similar theoretical results hold in the classification setting where a key difference is the loss functions employed (Lipschitz continuous loss such as support vector machine and logistic regression). (Bach 2017), (Sun, Gilbert, and Tewari 2018), and (Li et al. 2021) study the generalization properties of RFFs approximations in the classification setting. They show that RFFs estimators can provide computational gains while still preserving the statistical properties of the original kernel method. Nevertheless, a key requirement in these analyses is to employ a certain problem-dependent sampling distribution. Computing such a distribution often requires $O(n^3)$ in time and $O(n^2)$ in space already and hence is itself intractable. Therefore, whether RFFs classification can provide computational savings without using the problem-dependent sampling distribution remains unclear, and a detailed trade-off between the number of features required and the statistical prediction accuracy is still missing.

A key step in obtaining a better trade-off for RFFs regression is to employ the regularity condition (see Assumption A.3). However, this property is not used while analyzing the RFFs classification. In this paper, by incorporating the regularity condition, we improve the optimal results in the literature and provide a definitive answer to questions mentioned above by making the following contributions

- Under suitable regularity condition (Assumption A.3), Theorem 1 shows that RFFs classification only requires $\Omega(\sqrt{n}\log n)$ features to guarantee the minimax optimal $O(1/\sqrt{n})$ learning rate, the same prediction accuracy as the original kernel classification methods. Our analysis allows the computational cost to reduce from $O(n^3)$ in time and $O(n^2)$ in space to $O(n^2)$ and $O(n\sqrt{n})$ respectively, and suggest that for a wide range of classification problems, RFFs approximations provide dramatic computational cost savings without loss of prediction accuracy. To the best of our knowledge, this is the first result confirming that such a computational gain is possible in the classification setting when the standard sampling method is used.
- Using Massart's low noise assumption (Assumption A.4), Theorem 2 further provides a more refined analysis on the generalization properties of the RFF classification estimators. We obtain a sharp $O(1/n)$ learning rate for classification at the expense of more random features required.
- We also discuss how problem-dependent sampling distribution further reduces the computational cost in the $O(1/\sqrt{n})$ rate setting and the $O(1/n)$ rate setting. Our analysis expresses the trade-off between the number of features required and the statistical prediction accuracy in terms of the regularization parameter (λ) and the *effective degree of freedom* ($d(\lambda)$) and points out how utilizing the optimized feature can lead to a significant reduction in the computational cost.
- Finally, in Table 1 and 2, we provide a comprehensive comparison between achieved results in this paper and the optimal bound in the literature. The analysis demonstrates that under benign conditions, our study obtains the sharpest bound on the number of features required in literature, while under worst case scenario, we match the optimal results in the literature.

Background

Supervised Learning with Kernels

Let $P(x, y) = P_x P(y \mid x)$ be a joint probability density function defined on $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is an instance space and $\mathcal{Y}$ a label space. While in regression tasks $\mathcal{Y} \subset \mathbb{R}$, in classification tasks it is typically the case that $\mathcal{Y} = \{-1, 1\}$. Let $\{(x_i, y_i)\}_{i=1}^n$ be a training set sampled independently from $P(x, y)$. The goal of a supervised learning defined with a kernel function k (and the associated reproducing kernel Hilbert space $\mathcal{H}$) is to find a hypothesis $f\colon \mathcal{X} \to \mathcal{Y}$ such that $f \in \mathcal{H}$ and $f(x)$ is a good estimate of the label $y \in \mathcal{Y}$ corresponding to a previously unseen instance $x \in \mathcal{X}$. In particular, the learning can be formulated as the following optimization problem

$$\hat{f}^\lambda := \operatorname*{arg\,min}_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n l(y_i, f(x_i)) + \lambda \|f\|_{\mathcal{H}}^2 .$$

where $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ is a loss function and λ is the regularization parameter to prevent overfitting. As a result of the representer theorem (Schölkopf and Smola 2001), an empirical risk minimization estimator in this setting can be expressed as $\hat{f}^\lambda = \sum_{i=1}^n \alpha_i k(x_i, \cdot)$ with $\alpha \in \mathbb{R}^n$ and the optimization problem can be reformulated as

$$\hat{\alpha}_k^\lambda := \operatorname*{arg\,min}_{\alpha \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^n l(y_i, (\mathbf{K}\alpha)_i) + \lambda \alpha^T \mathbf{K} \alpha , \quad (1)$$

where $\mathbf{K}$ is the kernel Gram-matrix with $\mathbf{K}_{i,j} = k(x_i, x_j)$.

Learning Risk The hypothesis $\hat{f}^\lambda$ is an empirical estimator and we use the learning risk to assess its ability to capture the relationship between instances and labels given by P (Caponnetto and De Vito 2007)

$$\mathbb{E}_P[l_{\hat{f}^\lambda}] = \int_{\mathcal{X} \times \mathcal{Y}} l(y, \hat{f}^\lambda(x)) dP(x, y) ,$$

where we use l_f to denote $l(y, f(x))$. When the context is clear, we will omit P from the expectation and write $\mathbb{E}[l_{\hat{f}^\lambda}]$.

The empirical distribution $P_n(x, y)$ is given by a sample of n examples drawn independently from $P(x, y)$. The empirical risk is used to estimate the learning risk $\mathbb{E}[l_{\hat{f}^\lambda}]$ and it is given by

$$\mathbb{E}_n[l_{\hat{f}^\lambda}] = \frac{1}{n} \sum_{i=1}^n l(y_i, \hat{f}^\lambda(x_i)) .$$

Similar to Rudi and Rosasco (2017) and Caponnetto and De Vito (2007), we will assume [1] the existence of $f_{\mathcal{H}} \in \mathcal{H}$ such that $f_{\mathcal{H}} = \arg\min_{f \in \mathcal{H}} \mathbb{E}[l_f]$. Note that $\mathbb{E}[l_{f_{\mathcal{H}}}]$ is the lowest learning risk one can achieve in the reproducing kernel Hilbert space $\mathcal{H}$. Hence, theoretical studies of the estimator $\hat{f}^\lambda$ often concern how fast its learning risk $\mathbb{E}[l_{\hat{f}^\lambda}]$ converges to $\mathbb{E}[l_{f_{\mathcal{H}}}]$, that is, how fast the excess risk $\mathbb{E}[l_{\hat{f}^\lambda}] - \mathbb{E}[l_{f_{\mathcal{H}}}]$ converges to zero. In the remainder of the manuscript, we will refer to the rate at which the excess risk converges to zero as the *learning rate*.

[1] The existence of $f_{\mathcal{H}}$ depends on the complexity of $\mathcal{H}$ which is related to the data distribution $P(y|x)$. For more details, please see Caponnetto and De Vito (2007) and Rudi and Rosasco (2017).

Random Fourier Features

Despite providing a flexible non-linear approximation framework, kernel methods suffer from the scalability issue. In particular, kernel supervised learning often requires the store or the inverse of the kernel Gram matrix $\mathbf{K}$ ($O(n^2)$ and $O(n^3)$ computations respectively), which is prohibitive. As a result, many low-rank approximation algorithms have been designed to resolve this issue (see, e.g., Smola and Schökopf 2000; Williams and Seeger 2001b; Rahimi and Recht 2007, 2009; Mahoney and Drineas 2009, and references therein).

Among them, RFFs method is a widely used, simple, and effective technique for scaling up kernel methods. The idea is due to Bochner's theorem (Bochner 1932), which states that any bounded, continuous, and shift-invariant kernel is the Fourier transform of a bounded positive measure. Assuming the spectral measure $d\tau$ has a density function $p(\cdot)$, we can write the corresponding kernel as

$$
\begin{aligned}
k(x,y) &= \int_{\mathcal{V}} e^{-2\pi i v^T(x-y)} d\tau(v) \\
&= \int_{\mathcal{V}} \left(e^{-2\pi i v^T x}\right)\left(e^{-2\pi i v^T y}\right)^* p(v) dv \ ,
\end{aligned}
$$

where c^* denotes the complex conjugate of $c \in \mathbb{C}$. Typically, the kernel is real valued and we can ignore the imaginary part (see e.g., Rahimi and Recht 2007). Bach (2017) and Rudi and Rosasco (2017) further generalize the idea by considering the following decomposition of kernel functions

$$
k(x,y) = \int_{\mathcal{V}} \psi(v,x)\psi(v,y)p(v)dv \ , \tag{2}
$$

where $\psi \colon \mathcal{V} \times \mathcal{X} \to \mathbb{R}$ is a continuous and bounded function with respect to v and x. Hence, we can approximate the kernel function using its Monte-Carlo estimate

$$
\begin{aligned}
\tilde{k}(x,y) &= \frac{1}{s}\sum_{i=1}^{s} \psi(v_i,x)\psi(v_i,y) \ , \\
&= \phi_s(x)^T \phi_s(y) \ . \qquad (3)
\end{aligned}
$$

where $\{v_i\}_{i=1}^s$ are sampled independently from the spectral measure $p(v)$ and

$$
\phi_s(x) = \frac{1}{\sqrt{s}}[\psi(v_1,x), \ldots, \psi(v_s,x)]^\top .
$$

We denote the reproducing kernel Hilbert space spanned by $\tilde{k}$ as $\tilde{\mathcal{H}}$ (note that in general $\tilde{\mathcal{H}} \nsubseteq \mathcal{H}$). Let $\tilde{\mathbf{K}}$ be Gram-matrices with entries $\tilde{\mathbf{K}}_{ij} = \tilde{k}(x_i, x_j)$. Then the following equalities can be derived easily from Eq. (3)

$$
k(x,y) = \mathbb{E}_{v\sim p}\big[\tilde{k}(x,y)\big] \quad \wedge \quad \mathbf{K} = \mathbb{E}_{v\sim p}[\tilde{\mathbf{K}}] \ .
$$

In addition to the kernel Gram-matrix approximation, Bach (2017) establishes that any $f \in \mathcal{H}$ can be expressed as [2]

$$
f(x) = \int_{\mathcal{V}} g(v)\psi(v,x)p(v)dv \qquad (\forall x \in \mathcal{X}) \tag{4}
$$

where $g \in L_2(d\tau)$ is a real-valued function such that $\|g\|^2_{L_2(d\tau)} < \infty$ and $\|f\|_{\mathcal{H}} = \min_g \|g\|_{L_2(d\tau)}$, with the minimum taken over all possible decompositions of f. Thus, one can take an independent sample $\{v_i\}_{i=1}^s \sim p(v)$ (we refer to this sampling scheme as *plain RFF*) and approximate a function $f \in \mathcal{H}$ by an element from $\tilde{\mathcal{H}}$ as

$$
\tilde{f}(\cdot) = \sum_{i=1}^{s} \alpha_i \psi(v_i, \cdot) = \phi_s(\cdot)^\top \alpha \quad \text{with} \quad \alpha \in \mathbb{R}^s \ .
$$

As the latter approximation is simply a Monte Carlo estimate, one could also select an importance weighted probability density function $q(\cdot)$ and sample features $\{v_i\}_{i=1}^s$ from q (we refer to this sampling scheme as *weighted RFF*). The function f can then be approximated by

$$
\tilde{f}_q(\cdot) = \sum_{i=1}^{s} \alpha_i \psi_q(v_i, \cdot) = \phi_{q,s}(\cdot)^\top \alpha \ ,
$$

with $\psi_q(v_i,\cdot) = \sqrt{p(v_i)/q(v_i)}\psi(v_i,\cdot)$ and $\phi_{q,s}(\cdot) = (1/\sqrt{s})[\psi_q(v_1,\cdot), \cdots, \psi_q(v_s,\cdot)]^\top$.

For both plain RFF and weighted RFF, the goal is to find $\tilde{f}$ with minimal norm such that the computation error between $\tilde{f}$ and f is minimized. Similar to Bach (2017), the RFFs sampling can be formulated as the following optimization problem

$$
\|\tilde{f} - f\|^2_{L_2(P_x)} + \lambda \|\tilde{f}\|^2_{\tilde{\mathcal{H}}} \ . \tag{5}
$$

Note that since $\tilde{f} \notin \mathcal{H}$ in general, we use the $L_2(P_x)$ norm to measure the computation error.

Integral Operator & Leverage Score Sampling

Kernel methods and RFFs are often studied through the integral operator $L : L_2(P_x) \to L_2(P_x)$, which we define below

$$
(Lf)(\cdot) = \int_{\mathcal{X}} k(x,\cdot)f(x)dP_x(x) \ .
$$

Given the kernel decomposition as Eq. (2), the integral operator can be expressed as an expectation (Bach 2017)

$$
\begin{aligned}
Lf &= \int_{\mathcal{X}} k(x,\cdot)f(x)dP_x(x) \ , \\
&= \left(\int_{\mathcal{V}} \psi(v,\cdot) \otimes \psi(v,\cdot)p(v)dv\right) f \ , \qquad (6)
\end{aligned}
$$

where $f \otimes g$ is the $L_2(P_x)$ outer product operator such that $(f \otimes g)\, h = \langle g, h\rangle_{L_2(P_x)} f$. Finally, if k and ψ are both bounded and continuous, then L is positive definite, self-adjoint and trace-class. In particular, if $\|\psi\| \leq \kappa$, we have $\|L\| \leq \kappa^2$.

Similarly, for kernel $\tilde{k}$, we define the integral operator $L_s : L_2(P_x) \to L_2(P_x)$:

$$
\begin{aligned}
L_s f &= \int_{\mathcal{X}} \tilde{k}(x,\cdot)f(x)dP_x(x) \ , \\
&= \int_{\mathcal{X}} \frac{1}{s}\sum_{i=1}^{s} \psi(v_i,\cdot)\psi(v_i,x)f(x)dP_x \ , \\
&= \left(\frac{1}{s}\sum_{i=1}^{s} \psi(v_i,\cdot) \otimes \psi(v_i,\cdot)\right) f \ . \qquad (7)
\end{aligned}
$$

[2] It is not necessarily true that for any $g \in L_2(d\tau)$, there exists a corresponding $f \in \mathcal{H}$.

Hence, L_s can be seen as an empirical estimator of L.

The study of the integral operator is important because it provides information on how to select the optimal sampling distribution $q(v)$. A large body of literature shows that finding an optimal sampling distribution $q(v)$ often significantly reduces the number of features required (Bach 2017; Alaoui and Mahoney 2015; Avron et al. 2017; Rudi and Rosasco 2017). The reason is that random features sampled according to $p(v)$ often focus on approximating the leading eigenvalues of the integral operator L. In contrast, a reweighted sampling distribution $q(v)$ allows the random features to span the whole eigenspectrum of L.

In light of this, a leverage score based weighted distribution function is first introduced in Alaoui and Mahoney (2015) in the context of the Nyström approximation (Nyström 1930; Smola and Schölkopf 2000; Williams and Seeger 2001b). Utilizing the importance reweighted nature, Alaoui and Mahoney (2015) establish a sharp convergence rate of the low-rank estimator based on the Nyström method.

The success of the leverage score distribution further motivates the pursuit of a similar notion for RFFs. In particular, Bach (2017) first proposes the leverage score sampling based on a leverage score function defined below

$$\tau_\lambda(v) = p(v)\langle \psi(v,\cdot), (L+\lambda I)^{-1}\psi(v,\cdot)\rangle_{L_2(P_x)} \, . \quad (8)$$

From our assumption, it follows that there exists a constant κ such that $|\psi(v,x)| \le \kappa$ (for all v and x). We now have

$$\tau_\lambda(v) \le p(v)\frac{\kappa^2}{\lambda} \, .$$

An important property of function $\tau_\lambda(v)$ is its relation to the effective number of parameters:

$$\int_{\mathcal{V}} \tau_\lambda(v) dv = \mathrm{Tr}\big[L(L+\lambda I)^{-1}\big] := d(\lambda) \, ,$$

where $d(\lambda)$ implicitly determines the number of parameters in a supervised learning problem and is thus called the *number of effective degrees of freedom* (Bach 2013; Hastie 2017).

We can now sample features according to $q^*(v) = \tau_\lambda(v)/d(\lambda)$, since $q^*(v)$ is a probability density function. Bach (2017) studies the property of $q^*(v)$ and demonstrates that sampling according to $q^*(v)$ requires fewer Fourier features compared to the standard spectral measure sampling. From now on, we refer to $q^*(v)$ as the *ridge leverage score distribution* and refer to this sampling strategy as *leverage weighted RFF*.

Main Results

In this section, we provide our theoretical analysis on the trade-off between the number of random features and the statistical prediction accuracy. We first discuss the worst case scenario where the estimator achieves the $O(1/\sqrt{n})$ learning rate, followed by demonstrating the trade-off in the fast convergence rate setting.

$O(1/\sqrt{n})$ Learning Rate

We study the scenario where the RFFs estimator obtains the minimax learning rate $O(1/\sqrt{n})$. As discussed before, kernel supervised learning can be formulated as Eq. (1). Since we are investigating the classification setting, we mainly consider the loss function l to be uniformly Lipschitz continuous functions such as support vector machine and logistic regression. A fatal problem for kernel supervised learning is the computational cost since kernel learning problem such as Eq. (1) often requires the store of the kernel Gram matrix $\mathbf{K}$ or even the inversion of $\mathbf{K}$, which are $O(n^2)$ and $O(n^3)$ computations respectively.

In order to overcome the computation issue, the RFFs provide an efficient way to approximate the kernel function. Specifically, we sample $v_1, \ldots, v_s$ according to some importance sampling distribution $q(v)$ to form the random feature vector $\phi_s(\cdot)$. For a given data (x,y), we then approximate the label y with the random feature hypothesis $\tilde{f}_q(x) = \phi_{q,s}(x)^\top \beta$. The RFFs learning can be cast as the following optimization problem

$$\tilde{f}^\lambda := \underset{\tilde{f}_q \in \tilde{\mathcal{H}}}{\arg\min} \; \frac{1}{n}\sum_{i=1}^{n} l(y_i, \tilde{f}_q(x_i)) + \lambda \|\tilde{f}\|^2_{\tilde{\mathcal{H}}} \, .$$

According to Bach (2017) and Li et al. (2019), we have $\|\tilde{f}\|^2_{\tilde{\mathcal{H}}} \le \|\beta\|_2^2$, as a result, the above optimization can be reformulated as

$$\tilde{\beta}^\lambda := \underset{\beta \in \mathbb{R}^s}{\arg\min} \; \frac{1}{n}\sum_{i=1}^{n} l(y_i, \phi_{q,s}^\top \beta) + \lambda \|\beta\|_2^2 \, . \quad (9)$$

The RFFs hypothesis with loss function l can be represented as $\tilde{f}^\lambda = \phi_{q,s}^\top \tilde{\beta}^\lambda$. Through the RFFs approximation, we now only need to store the feature matrix $\Phi_q = [\phi_{q,s}(x_1), \ldots, \phi_{q,s}(x_n)]^\top \in \mathbb{R}^{n\times s}$. The inversion of $\mathbf{K}$ can be approximated as inverting $\Phi_q^\top \Phi_q \in \mathbb{R}^{s\times s}$. Hence the computation cost is now $O(ns)$ and $O(ns^2+s^3)$ respectively. We can see that if $s \ll n$, RFFs method enjoys a huge computational savings. However, a key question is how the choice of s affects the prediction accuracy of $\tilde{f}^\lambda$.

In this section, we try to address the above issue. We first list our assumptions below

A.1 We assume that the kernel has integral expansion as Eq. (2) such that $\psi(v,x)$ is continuous in both v and x and $|\psi(v,x)| \le \kappa$ for all $x \in \mathcal{X}$ and $v \in \mathcal{V}$;

A.2 Assume that the loss function l in Eq. (9) is uniformly Lipschitz continuous with constant M, i.e.,

$$|l(y,x_1) - l(y,x_2)| \le M\|x_1 - x_2\|_2.$$

A.3 Recall $f_{\mathcal{H}} = \arg\min_{f\in\mathcal{H}} \mathbb{E}[l_f]$, we assume that

$$f_{\mathcal{H}} = L^r g, \;\; \text{for some } r \in [1/2, 1] \;\&\; g \in L_2(P_x);$$

Assumptions A.1 and A.2 are standard assumptions made in classification problems. A.3 is a regularity condition that is commonly used in approximation theory (Smale and Zhou 2003). It describes the decay rate of the coefficients of $f_{\mathcal{H}}$ along the basis given by the integral operator L, which further allows controlling the bias of the estimator. While being overlooked in the classification setting, A.3 is a key property used in RFFs regression to obtain a better computation and accuracy trade-off. Utlizing A.3 enables us to prove the following refined analysis.

Theorem 1. *Assume A.1, A.2 and A.3 hold. Suppose we have a measurable function $\tilde{\tau} : \mathcal{V} \to \mathbb{R}$ such that $\tilde{\tau}(v) \geq \tau_\lambda(v)$ almost surely. Denote $d_{\tilde{\tau}} = \int_{\mathcal{V}} \tilde{\tau}(v) dv$, and let $q(v) = \frac{\tilde{\tau}(v)}{d_{\tilde{\tau}}}$. We sample $v_1, \ldots, v_s \sim q(v)$ and compute the hypothesis $\tilde{f}^\lambda$ by solving the optimization problem in Eq. (9). Let $\delta \in (0, 1)$, if we have*

$$s \geq 12 d_{\tilde{\tau}} \log \frac{d(\lambda)}{\delta},$$

with probability over $1 - \delta$,

$$\mathbb{E}(l_{\tilde{f}^\lambda}) - \mathbb{E}(f_{\mathcal{H}}) \leq 2MR\lambda^r + O\left(1/\sqrt{n}\right) . \quad (10)$$

Theorem 1 expresses the trade-off between the computational cost and statistical efficiency through the regularization parameter λ, the effective dimension of the problem $d(\lambda)$, and the normalization constant $d_{\tilde{\tau}}$ of the sampling distribution. The regularization parameter λ is used as a key quantity in the analysis of supervised learning setting (Caponnetto and De Vito 2007; Rudi and Rosasco 2017; Li et al. 2019). In particular, if we set $\lambda \propto 1/n^{2r}$, we observe that the estimator $\tilde{f}^\lambda$ attains the $O(1/\sqrt{n})$ learning rate (Bach 2017). As a consequence of Theorem 1, we have the following bounds on the number of required features for the two strategies: *plain* RFF (Corollary 1) and *leverage weighted* RFF (Corollary 2).

Corollary 1. *If the probability density function from Theorem 1 is the spectral measure $p(v)$, then the upper bound on the learning risk from Eq. (10) holds for all $s \geq 5\kappa^2/\lambda \log \frac{16d(\lambda)}{\delta}$.*

Proof. We set $\tilde{\tau}(v) = p(v)\kappa^2/\lambda$ and obtain $d_{\tilde{\tau}} = \int_{\mathcal{V}} p(v)\kappa^2/\lambda dv = \kappa^2/\lambda$. □

Theorem 1 and Corollary 1 have several implications on the choice of λ and s in the classification setting with plain RFF. In particular, the usual generalization bound for kernel estimator $\hat{f}^\lambda$ (i.e., minimizer of Eq. (1)) is $O(1/\sqrt{n})$ (see e.g., Rahimi and Recht 2009; Shalev-Shwartz and Ben-David 2014; Bach 2017). As such, if we set $\lambda = O(n^{-1/2r})$, we can see that the RFFs estimator $\tilde{f}^\lambda$ incurs no loss of prediction accuracy while offering computational gains.

Specifically, in the benign case where $r = 1$, $O(\sqrt{n} \log n)^3$ features is able to achieve the $O(1/\sqrt{n})$ learning rate. Comparing with the existing analysis where $O(n \log n)$ features are required (Rahimi and Recht 2009; Li et al. 2019), our result is a significant improvement. We also achieve remarkable computational savings: from roughly $O(n^3)$ and $O(n^2)$ in time and space for original kernel methods to $O(n^2)$ and $O(n\sqrt{n})$ for the RFFs approximation. Moreover, when $r > 1/2$, we also obtain computational gain as the number of features required now is $\Omega(n^{1/2r})$ with $2r > 1$. In the worst scenario where $r = 1/2$ (equivalent to assuming $f_{\mathcal{H}}$ exists), we recover the results from existing analysis (Rahimi and Recht 2007; Li et al. 2019).

To our knowledge, this is the first result showing that for a large class of classification problems ($r > 1/2$), RFFs classification can dramatically reduce the computational cost while preserving the optimal generalization properties.

Corollary 2. *If the probability density function from Theorem 1 is the ridge leverage score distribution $q^*(v)$, the upper bound on the risk from Eq. (10) holds for all $s \geq 5d(\lambda) \log \frac{16d(\lambda)}{\delta}$.*

Proof. For this corollary, we set $\tilde{\tau}(v) = \tau_\lambda(v)$ and deduce $d_{\tilde{l}} = \int_{\mathcal{V}} \tau_\lambda(v) dv = d(\lambda)$. □

Corollary 2 details the number of features required in the leverage weighted RFF setting. Similar to the plain RFF setting, the RFFs estimator obtains $O(1/\sqrt{n})$ rate once we set $\lambda = O(n^{1/2r})$. However, the choice of s now is determined by two factors: the regularity condition r and the decay rate of the eigenspectrum of L.

We first consider the benign scenario where $r = 1$. Depending on the eigenspectrum decay rate, we have several different cases. Denote $\{\mu_1, \mu_2, \ldots, \}$ to be the eigenvalue of L, in the best case where L has finite rank, $d(\lambda)$ remains constant as n grows. We therefore conclude that even $\Omega(1)$ features can guarantee the $O(1/\sqrt{n})$ learning rate. Next, if the eigenspectrum displays exponential decay, i.e., $\mu_i \propto C_0 r^i$, we have $d(\lambda) \leq \log(C_0^0/\lambda)$. We can see that $s \geq \log n \log \log n$ is enough to achieve the $O(1/\sqrt{n})$ learning rate. As such, significant computational savings is obtained: from $O(n^3)$ and $O(n^2)$ to $O(n \log^4 n)$ and $O(n \log n)$ respectively. In the case of a slower decay with $\mu_i \propto C_0 i^{-2\gamma}$, we have $d(\lambda) \leq (R_0/\lambda)^{1/(2\gamma)}$ and $s \geq n^{1/4\gamma} \log n$. Hence, substantial computational savings can be achieved even in this case. Furthermore, in the worst case with μ_i close to $C_0 i^{-1}$, our bound implies that $s \geq n^{1/2} \log n$ features are sufficient.

The analysis for the worst case scenario where $r = 1/2$ is similar. The required numbers of features are $\Omega(1), \Omega(\log n \log \log n), \Omega(n^{1/2\gamma})$ and $\Omega(n \log n)$ for the cases where the eigenspectrum has finite rank, decays exponentially, proportional to $C_0 i^{-2\gamma}$ and close to $C_0 i^{-1}$, respectively. Our results demonstrate that huge computational savings are possible as long as the eigenspectrum of L displays fast decay (faster than i^{-1}).

Comparison with Existing Sharpest Results Under $O(1/\sqrt{n})$ learning rate setting, Rahimi and Recht (2009), Bach (2017), and Li et al. (2021) analyze the trade-off between the number of features and the statistical prediction accuracy. Table 1 provides a detailed comparison between this work and that from Li et al. (2021). Our results show that we obtain at least the same rate as the previous best rate in the literature when $r = 1/2$, while significantly improving the trade-off under benign conditions ($r > 1/2$).

The first block of rows in Table 1 illustrates the difference between our work and that from Li et al. (2021) when plain sampling is used. A key feature in our results is that when $r > 1/2$, our results state that the RFFs approximation provides computational gain without trading off for the prediction accuracy. In comparison, results from Li et al. (2021) state that there is no computational gain ($s = \Omega(n)$) if we were to achieve the $O(1/\sqrt{n})$ learning rate. In addition, we recover the results from Li et al. (2021) when $r = 1/2$.

We observe a similar pattern when leverage weighted RFF is used. In particular, our results match those from Li et al.

[3] We use the fact that $d(\lambda) \ll n$

SAMPLING SCHEME	SPECTRUM	THIS WORK	LI ET AL. (2021)	LEARNING RATE
PLAIN RFF	FINITE RANK	$s \in \Omega(n^{1/2r})$	$s \in \Omega(n)$	$O(1/\sqrt{n})$
	EXPONENTIAL DECAY	$s \in \Omega(n^{1/2r} \cdot \log\log n)$	$s \in \Omega(n \cdot \log\log n)$	
	POLYNOMIAL DECAY	$s \in \Omega(n^{1/2r} \cdot \log n)$	$s \in \Omega(n \cdot \log n)$	
WEIGHTED RFF	FINITE RANK	$s \in \Omega(1)$	$s \in \Omega(1)$	$O(1/\sqrt{n})$
	EXPONENTIAL DECAY	$s \in \Omega(\log n \cdot \log\log n)$	$s \in \Omega(\log n \cdot \log\log n)$	
	$\mu_i \propto i^{-2\gamma}, \gamma \geq 1$	$s \in \Omega(n^{1/4\gamma r} \cdot \log n)$	$s \in \Omega(n^{1/2\gamma} \cdot \log n)$	
	$\mu_i \propto i^{-1}$	$s \in \Omega(n^{1/2r} \cdot \log n)$	$s \in \Omega(n \cdot \log n)$	

Table 1: The comparison of our results to the sharpest learning rates from prior work (Li et al. 2021), where $r \in [1/2, 1]$.

(2021), when the eigenspectrum has finite rank or displays exponential decay. However, as soon as the eigenspectrum has polynomial decay, our result is sharper. Specifically, when the eigenvalue decays polynomially with $\mu_i \propto i^{-2\gamma}$ and $r = 1$, our results show that $\Omega(n^{1/4\gamma \log n})$ features are enough to achieve $O(1/\sqrt{n})$ learning rate, comparing with $\Omega(n^{1/2\gamma})$ features required from Li et al. (2021). When the eigenvalue decays close to i^{-1}, our results require $\Omega(n^{1/2} \log n)$ features while Li et al. (2021) require $\Omega(n \log n)$ features.

Refined Learning Rate

In the previous section, we study the trade-off between the number of features and the statistical prediction accuracy in the $O(1/\sqrt{n})$ minimax learning rate setting. In general, it is hard for classification problems to obtain learning rates sharper than $O\left(1/\sqrt{n}\right)$. However, under some benign conditions, it is possible to obtain $O(1/n)$ convergence rate as demonstrated by Bartlett, Jordan, and McAuliffe (2006) and Steinwart and Christmann (2008). As such, with the help of the following assumption, we derive a sharp learning rate for RFFs classification problems in this section.

A.4 Recall that $f_{\mathcal{H}}$ is the optimal estimator in A.3. We assume that there exists a constant G such that for all $f \in \mathcal{H}$

$$\mathbb{E}[(f - f_{\mathcal{H}})^2] \leq G\mathbb{E}[l_f - l_{f_{\mathcal{H}}}] \,.$$

Assumption A.4 is a widely used condition for classification problems to obtain faster learning rates. It typically requires that the loss function l is uniformly convex and the function space $\mathcal{H}$ is convex and uniformly bounded. It can be shown that many loss functions satisfy this assumption, including squared loss (Bartlett et al. 2005) and hinge loss (Steinwart and Christmann 2008, Chapter 8.5). Additional examples of these loss functions are discussed in Bartlett, Jordan, and McAuliffe (2006) and Mendelson (2002). In addition, since l is Lipschitz continuous, we can rewrite A.4 as

$$\mathbb{E}[(l_f - l_{f_{\mathcal{H}}})^2] \leq L^2\mathbb{E}[(f - f_{\mathcal{H}})^2] \leq GL^2\mathbb{E}[l_f - l_{f_{\mathcal{H}}}] \,.$$

This is the variance condition described in Steinwart and Christmann (2008, Chapter 7.3), which is also linked to the Massart's low noise condition or more generally to the Tsybakov condition (Sun, Gilbert, and Tewari 2018). Intuitively speaking, the condition requires that the bayes classifier $P(Y = 1 \mid X = x)$ is not close to $1/2$ (see e.g., Tsybakov et al. 2004; Koltchinskii 2011, for more details).

Theorem 2. *Assume A.1-A.4 hold. In addition, we assume the condition for $\tilde{\tau}$, $d_{\tilde{\tau}}$ and $q(v)$ hold as that in Theorem 1. Let $\{\tilde{\mu}_1, \tilde{\mu}_2, \dots\}$ be the eigenvalues of the normalized Gram-matrix $(1/n)\tilde{\mathbf{K}}$, c_1, c_2, c_3 be some universal constant, and $\delta \in (0, 1)$, if we have*

$$s \geq 12 d_{\tilde{\tau}} \log \frac{d(\lambda)}{\delta},$$

with probability over $1 - \delta$,

$$\mathbb{E}(l_{\tilde{f}^{\lambda}}) - \mathbb{E}\left(f_{\mathcal{H}}\right) \leq 2MR\lambda^r + c_1 \hat{r}^* + \frac{c_2}{n} \log \frac{1}{\delta} \,,$$

where

$$\hat{r}^* \leq c_3 \min_{0 \leq h \leq n} \left(\frac{h}{n} + \sqrt{\frac{1}{n} \sum_{i>h} \tilde{\mu}_i} \right) .$$

Theorem 2 covers a wide range of cases and can provide sharp risk convergence rates. In particular, $\hat{r}^*$ has an upper bound of $O(1/\sqrt{n})$ in all cases, which happens when $\hat{\mu}_i$ decays polynomially as $O(n^{-\gamma})$ with $\gamma > 1$ and we let $h = 0$. On the other hand, if $\hat{\mu}_i$ decays exponentially, then setting $h = \lceil \log n \rceil$ implies that $\hat{r}^* \leq O(\log n/n)$. In the best case, when $(1/n)\mathbf{K}$ has only finite rank d_K, then $\hat{r}^* \leq O(1/n)$ by letting $h = d_K + 1$. These different upper bounds provide insights into various trade-offs between computational complexity and statistical efficiency. We now split the discussion into two cases: plain RFF and leverage weighted RFF.

Under plain RFF strategy, similar to Corollary 1, we have $d_{\tilde{\tau}} \leq \kappa^2/\lambda$. If the eigenvalues decay polynomially, i.e., $\mu_i \propto i^{-\gamma}$ with $\gamma > 1$, then the learning rate is upper bounded by $O(1/\sqrt{n})$. In this case, we need $s = \Omega(n^{1/2r} \log n)$. On the other hand, if μ_i decays exponentially, $\Omega(n^{1/r})$ features are able to guarantee $O(\log n/n)$ learning rate. Finally, if the eigenspectrum has finite rank, $\Omega(n^{1/r})$ features yield $O(1/n)$ fast learning rate. Under the leverage weighted RFF, the

SAMPLING SCHEME	RESULTS	SPECTRUM	NUMBER OF FEATURES	LEARNING RATE
PLAIN RFF	THIS WORK	FINITE RANK	$s \in \Omega(n^{1/r})$	$O(1/n)$
		EXPONENTIAL DECAY	$s \in \Omega(n^{1/r})$	$O(\log n/n)$
		$\mu_i \propto i^{-\gamma}$	$s \in \Omega(n^{1/2r} \cdot \log n)$	$O(1/\sqrt{n})$
	LI ET AL. (2021)	FINITE RANK	$s \in \Omega(n^2)$	$O(1/n)$
		EXPONENTIAL DECAY	$s \in \Omega(n^2)$	$O(\log n/n)$
		$\mu_i \propto i^{-\gamma}$	$s \in \Omega(n \cdot \log n)$	$O(1/\sqrt{n})$
WEIGHTED RFF	THIS WORK	FINITE RANK	$s \in \Omega(1)$	$O(1/n)$
		EXPONENTIAL DECAY	$s \in \Omega(\log n \cdot \log\log n)$	$O(\log n/n)$
		$\mu_i \propto i^{-\gamma}$	$s \in \Omega(n^{1/4\gamma r} \cdot \log n)$	$O(1/\sqrt{n})$
	SUN (2018)	FINITE RANK	$s \in \Omega(1)$	$O(1/n)$
		EXPONENTIAL DECAY	$s \in \Omega(\log^d n \cdot \log\log^d n)$	$O(\log^{d+2} n/n)$
		$\mu_i \propto i^{-\gamma}$	$s \in \Omega(n^{\frac{2}{2+\gamma}} \cdot \log n)$	$O(1/n^{\frac{\gamma}{2+\gamma}})$

Table 2: The comparison of our results to the sharpest results in the literature under fast learning rate setting, where $r \in [1/2, 1]$.

required numbers of features and the corresponding learning rates for the three above cases are: *i)* $s = \Omega(n^{1/4\gamma r} \log n)$ and $O(1/\sqrt{n})$ (polynomial decay $\mu_i \propto i^{-\gamma}$ with $\gamma > 1$); *ii)* $s = \Omega(\log n \log\log n)$ and $O(\log n/n)$ (exponential decay); and *iii)* $s = \Omega(1)$ and $O(1/n)$ (finite rank).

Comparison with Existing Sharpest Results For fast learning rate scenario, Li et al. (2021) and Sun, Gilbert, and Tewari (2018) both study the trade-off between the number of features and the prediction accuracy and obtain a similar bound on the number of features required. Table 2 compare our results with that in Li et al. (2021) under plain RFF sampling, and that in Sun, Gilbert, and Tewari (2018) under leverage weighted RFF sampling. Similar to the analysis in the $O(1/\sqrt{n})$ scenario, our results strictly dominate previous optimal results in Li et al. (2021) when $r > 1/2$ and match the obtained bound in Li et al. (2021) when $r = 1/2$, because of Assumption A.3.

Under weighted RFF, our results match that in Sun, Gilbert, and Tewari (2018) when the eigenspectrum has finite rank. However, when the eigenspectrum displays exponential decay, results from Sun, Gilbert, and Tewari (2018) suffer from the curse of dimension, since both the number of features required and the learning rate obtained depend on the data dimension d. In contrast, our analysis does not have this dependency. When the eigenspectrum exhibits a polynomial decay, our results achieve the $O(1/\sqrt{n})$ learning rate while Sun, Gilbert, and Tewari (2018) obtain a more flexible rate that depends on γ. If $\gamma \leq 2$, our results have a better trade-off as both the number of features and the learning rate are sharper than those from Sun, Gilbert, and Tewari (2018). For example, if $\gamma = 2$, analysis in Sun, Gilbert, and Tewari (2018) shows that $\Omega(n^{1/2} \log n)$ features can obtain $O(n^{-1/2})$ learning rate, whereas our results state that $\Omega(n^{1/8r} \log n)$ features yield the same learning rate. When $\gamma > 2$, the learning rate obtained by Sun, Gilbert, and Tewari (2018) is faster than ours at the cost of increasing the number of features, i.e., $\Omega(n^{\frac{2}{2+t}} \cdot \log n)$ versus $\Omega(n^{\frac{1}{4\gamma r}} \cdot \log n)$. In particular, setting $\gamma = 4$, we can see that Sun, Gilbert, and Tewari (2018) obtain a fast $O(n^{2/3})$ learning rate, but at the cost of requiring $\Omega(n^{1/3} \cdot \log n)$ random features. For the same setting, on the other hand, we obtain the minimax optimal $O(1/\sqrt{n})$ learning rate with only $\Omega(n^{1/16r} \cdot \log n)$ random features.

Conclusion

In this paper, we thoroughly study the generalization properties of RFFs classification with Lipschitz continuous loss such as support vector machine and logistic regression. Our main results for the first time demonstrate that RFFs classification can indeed provide computational gains without hurting the prediction accuracy when plain RFF is used. This is in contrast with all previous results that suggest that computational savings come at the expense of prediction accuracy. Furthermore, our analysis shows that a fast $O(1/n)$ learning rate is possible at the cost of increasing the number of features unless the leverage weighted RFF is used. However, a limitation in our work is that in the worst case where $r = 1/2$, we can see that the current analysis on RFFs classification cannot guarantee a computational gain. Therefore, how to obtain a sharper result in the worst case is an interesting future direction.

References

Alaoui, A.; and Mahoney, M. W. 2015. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems*, 775–783.

Avron, H.; Kapralov, M.; Musco, C.; Musco, C.; Velingker, A.; and Zandieh, A. 2017. Random Fourier features for kernel ridge regression: Approximation bounds and statistical guarantees. In *International Conference on Machine Learning*, 253–262.

Bach, F. 2013. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, 185–209.

Bach, F. 2017. On the equivalence between kernel quadrature rules and random feature expansions. *Journal of Machine Learning Research*, 18(21): 1–38.

Bartlett, P. L.; Bousquet, O.; Mendelson, S.; et al. 2005. Local Rademacher complexities. *The Annals of Statistics*, 33(4): 1497–1537.

Bartlett, P. L.; Jordan, M. I.; and McAuliffe, J. D. 2006. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473): 138–156.

Bochner, S. 1932. Vorlesungen über Fouriersche Integrale. In *Akademische Verlagsgesellschaft*.

Caponnetto, A.; and De Vito, E. 2007. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3): 331–368.

Dai, B.; Xie, B.; He, N.; Liang, Y.; Raj, A.; Balcan, M.-F. F.; and Song, L. 2014. Scalable Kernel Methods via Doubly Stochastic Gradients. *Advances in Neural Information Processing Systems*, 27: 3041–3049.

Hastie, T. J. 2017. Generalized additive models. In *Statistical models in S*, 249–307. Routledge.

Huang, P.-S.; Avron, H.; Sainath, T. N.; Sindhwani, V.; and Ramabhadran, B. 2014. Kernel methods match deep neural networks on timit. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 205–209. IEEE.

Koltchinskii, V. 2011. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media.

Li, Z.; Ton, J.-F.; Oglic, D.; and Sejdinovic, D. 2019. Towards a unified analysis of random Fourier features. In *International Conference on Machine Learning*, 3905–3914. PMLR.

Li, Z.; Ton, J.-F.; Oglic, D.; and Sejdinovic, D. 2021. Towards a Unified Analysis of Random Fourier Features. *Journal of Machine Learning Research*, 22(108): 1–51.

Liu, F.; Huang, X.; Chen, Y.; and Suykens, J. A. 2020. Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond. *arXiv preprint arXiv:2004.11154*.

Mahoney, M. W.; and Drineas, P. 2009. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3): 697–702.

Mendelson, S. 2002. Improving the sample complexity using global data. *IEEE transactions on Information Theory*, 48(7): 1977–1991.

Nyström, E. J. 1930. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*.

Rahimi, A.; and Recht, B. 2007. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, 1177–1184.

Rahimi, A.; and Recht, B. 2009. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, 1313–1320.

Rudi, A.; Carratino, L.; and Rosasco, L. 2017. FALKON: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, 3891–3901.

Rudi, A.; and Rosasco, L. 2017. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, 3218–3228.

Rudin, W. 2017. *Fourier analysis on groups*. Courier Dover Publications.

Schölkopf, B.; and Smola, A. J. 2001. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.

Schölkopf, B.; Tsuda, K.; and Vert, J.-P. 2004. *Kernel methods in computational biology*. MIT press.

Shalev-Shwartz, S.; and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; and Cotter, A. 2011. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1): 3–30.

Smale, S.; and Zhou, D.-X. 2003. Estimating the approximation error in learning theory. *Analysis and Applications*, 1(01): 17–41.

Smola, A. J.; and Schökopf, B. 2000. Sparse Greedy Matrix Approximation for Machine Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 911–918.

Smola, A. J.; and Schölkopf, B. 2000. Sparse Greedy Matrix Approximation for Machine Learning. In *Proceedings of the 17th International Conference on Machine Learning*.

Sriperumbudur, B.; and Szabó, Z. 2015. Optimal rates for random Fourier features. In *Advances in Neural Information Processing Systems*, 1144–1152.

Steinwart, I.; and Christmann, A. 2008. *Support vector machines*. Springer Science & Business Media.

Sun, Y.; Gilbert, A.; and Tewari, A. 2018. But How Does It Work in Theory? Linear SVM with Random Features. In *Advances in Neural Information Processing Systems*, 3379–3388.

Sutherland, D. J.; and Schneider, J. 2015. On the error of random Fourier features. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 862–871. AUAI Press.

Tsybakov, A. B.; et al. 2004. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1): 135–166.

Williams, C. K.; and Seeger, M. 2001a. Using the Nyström method to speed up kernel machines. In *Advances in neural information processing systems*, 682–688.

Williams, C. K. I.; and Seeger, M. 2001b. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems 13*.

Zhang, Y.; Duchi, J.; and Wainwright, M. 2015. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1): 3299–3340.