Optimal Distributed Learning with Multi-pass Stochastic Gradient Methods

Junhong Lin [1] **Volkan Cevher** [1]

Abstract

We study generalization properties of distributed algorithms in the setting of nonparametric regression over a reproducing kernel Hilbert space (RKHS). We investigate distributed stochastic gradient methods (SGM), with mini-batches and multi-passes over the data. We show that optimal generalization error bounds can be retained for distributed SGM provided that the partition level is not too large. Our results are superior to the state-of-the-art theory, covering the cases that the regression function may not be in the hypothesis spaces. Particularly, our results show that distributed SGM has a smaller theoretical computational complexity, compared with distributed kernel ridge regression (KRR) and classic SGM.

1. Introduction

In statistical learning theory, a set of N input-output pairs from an unknown distribution is observed. The aim is to learn a function which can be used to predict future outputs given the corresponding inputs. The quality of a predictor is often measured in terms of the mean-squared error. In this case, the conditional mean, which is called as the regression function, is optimal among all the measurable functions (Cucker & Zhou, 2007; Steinwart & Christmann, 2008).

In nonparametric regression problems, the properties of the function to be estimated are not known a priori. Nonparametric approaches, which can adapt their complexity to the problem at hand, are key to good results. Kernel methods is one of the most common nonparametric approaches to learning (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004). It is based on choosing a RKHS as the hypothesis space in the design of learning algorithms. With an appropriate reproducing kernel, RKHS can be used to approximate any smooth function.

The classical algorithms to perform learning task are regularized algorithms, such as KRR, kernel principal component regression (KPCR), and more generally, spectral regularization algorithms (SRA). From the point of view of inverse problems, such approaches amount to solving an empirical, linear operator equation with the empirical covariance operator replaced by a regularized one (Engl et al., 1996; Bauer et al., 2007; Gerfo et al., 2008). Here, the regularization term is used for controlling the complexity of the solution to against over-fitting and for ensuring best generalization ability. Statistical results on generalization error had been developed in (Smale & Zhou, 2007; Caponnetto & De Vito, 2007) for KRR and in (Caponnetto, 2006; Bauer et al., 2007) for SRA.

Another type of algorithms to perform learning tasks is based on iterative procedure (Engl et al., 1996). In this kind of algorithms, an empirical objective function is optimized in an iterative way with no explicit constraint or penalization, and the regularization against overfitting is realized by early-stopping the empirical procedure. Statistical results on generalization error and the regularization roles of the number of iterations/passes have been investigated in (Zhang & Yu, 2005; Yao et al., 2007) for gradient methods (GM, also known as Landweber algorithm in inverse problems), in (Caponnetto, 2006; Bauer et al., 2007) for accelerated gradient methods (AGM, known as ν-methods in inverse problems) in (Blanchard & Krämer, 2010) for conjugate gradient methods (CGM), in (Rosasco & Villa, 2015) for incremental gradient methods (IGM), and in (Lin & Rosasco, 2017b) for (multi-pass) SGM.

Statistical results have been well studied for these algorithms; however, these algorithms suffer from computational burdens at least of order $O(N^2)$ due to the nonlinearity of kernel methods, where N is the sample size. Indeed, a standard execution of KRR requires $O(N^2)$ in space and $O(N^3)$ in time, while SGM after T-iterations requires $O(N)$ in space and $O(NT)$ (or T^2) in time. Such approaches would be prohibitive when dealing with large-scale learning problems, especially in the case where data cannot be stored on a single machine. These thus motivate one to study distributed learning algorithms (Mcdonald et al., 2009; Zhang et al., 2012). The basic idea of distributed learning is very simple: randomly divide a dataset of size N into m subsets of equal size, compute an independent estimator using a fixed algorithm on each subset, and then average the local

[1] Laboratory for Information and Inference Systems, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland. Correspondence to: Junhong Lin <junhong.lin@epfl.ch>, Volkan Cevher <volkan.cevher@epfl.ch>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

solutions into a global predictor. Interestingly, distributed learning technique has been successfully combined with KRR (Zhang et al., 2015; Lin et al., 2017) and more generally, SRA (Guo et al., 2017; Blanchard & Mucke, 2016b), and it has been shown that statistical results on generalization error can be retained provided that the number of partitioned subsets is not too large. Moreover, it was highlighted (Zhang et al., 2015) that distributed KRR not only allows one to handle large datasets that restored on multiple machines, but also leads to a substantial reduction in computational complexity versus the standard approach of performing KRR on all N samples.

In this paper, we study distributed SGM, with multi-passes over the data and mini-batches. The algorithm is a combination of distributed learning technique and (multi-pass) SGM (Lin & Rosasco, 2017b): it randomly partitions a dataset of size N into m subsets of equal size, computes an independent estimator by SGM for each subset, and then averages the local solutions into a global predictor. It has several free parameters: step-size, mini-batch size, total number of iterations and partition level m.

We show that with appropriate choices of algorithmic parameters, optimal generalization error bounds can be achieved provided that the partition level m is not too large. The proposed configuration has certain advantages on computational complexity. For example, without considering any benign properties of the studied problem such as the regularity of the regression function (Smale & Zhou, 2007; Caponnetto & De Vito, 2007) and a capacity assumption on the RKHS (Zhang, 2005; Caponnetto & De Vito, 2007), even implementing on a single machine, distributed SGM has an optimal convergence rate of order $O(N^{-1/2})$, with a computational complexity $O(N)$ in space and $O(N^{3/2})$ in time, compared with $O(N)$ in space and $O(N^2)$ in time of classic SGM performing on all N samples, or $O(N^{3/2})$ in space and $O(N^2)$ in time of distributed KRR. Moreover, the approach dovetails naturally with parallel and distributed computation: we are guaranteed a superlinear speedup with m parallel processors (though we must still communicate the function estimates from each processor). The proof of the main results is based on a similar error decomposition from (Lin & Rosasco, 2017b), which decomposes the excess risk into three terms: bias, sample and computational variance. The error decomposition allows one to study distributed GM and distributed SGM simultaneously. Different to those in (Lin & Rosasco, 2017b) which rely heavily on the intrinsic relationship of GM with the square loss, in this paper, an integral operator approach (Smale & Zhou, 2007; Caponnetto & De Vito, 2007) is used, combining with some novel and refined analysis. As a byproduct, we derive optimal statistical results on generalization error for non-distributed SGM, which improve on the results in (Lin & Rosasco, 2017b). Note also that we can extend our analysis to distributed SRA, and get better statistical results than those from (Zhang et al., 2015; Guo et al., 2017). We will report these results in a longer version of this paper.

The remainder of the paper is organized as follows. Section 2 introduces the supervised learning setting. Section 3 describes distributed SGM and its numerical realization, and then presents theoretical results on generalization error for distributed SGM, following with simple comments and discussions. Section 4 discusses and compares our results with related work. Proofs for distributed SGM and auxiliary lemmas are provided in the appendix.

2. Supervised Learning Problems

We consider a supervised learning problem. Let ρ be a probability measure on a measure space $Z = X \times Y$, where X is the input space and $Y \subseteq \mathbb{R}$ is the output space. Here, ρ is fixed but unknown. Its information can be only known through a set of samples $\bar{\mathbf{z}} = \{z_i = (x_i, y_i)\}_{i=1}^N$ of $N \in \mathbb{N}$ points, which we assume to be i.i.d..

The quality of a predictor $f : X \to Y$ can be measured in terms of the expected risk with a square loss defined as

$$\mathcal{E}(f) = \int_Z (f(x) - y)^2 d\rho(z). \tag{1}$$

In this case, the function minimizing the expected risk over all measurable functions is the regression function given by

$$f_\rho(x) = \int_Y y d\rho(y|x), \qquad x \in X. \tag{2}$$

The performance of an estimator $f \in L^2_{\rho_X}$ can be measured in terms of generalization error (excess risk), i.e., $\mathcal{E}(f) - \mathcal{E}(f_\rho)$. It is easy to prove that

$$\mathcal{E}(f) - \mathcal{E}(f_\rho) = \|f - f_\rho\|_\rho^2. \tag{3}$$

Here, $L^2_{\rho_X}$ is the Hilbert space of square integral functions with respect to ρ_X, with its induced norm given by $\|f\|_\rho = \|f\|_{L^2_{\rho_X}} = \left(\int_X |f(x)|^2 d\rho_X\right)^{1/2}$. For any $t \in \mathbb{N}_+$, the set $\{1, \cdots, t\}$ is denoted by $[t]$.

Kernel methods are based on choosing the hypothesis space as a RKHS. Recall that a reproducing kernel K is a symmetric function $K : X \times X \to \mathbb{R}$ such that $(K(u_i, u_j))_{i,j=1}^\ell$ is positive semidefinite for any finite set of points $\{u_i\}_{i=1}^\ell$ in X. The reproducing kernel K defines a RKHS $(H, \|\cdot\|_H)$ as the completion of the linear span of the set $\{K_x(\cdot) := K(x, \cdot) : x \in X\}$ with respect to the inner product $\langle K_x, K_u \rangle_H := K(x, u)$.

Given only the samples $\bar{\mathbf{z}}$, the goal is to learn the regression function f_ρ through efficient learning algorithms.

3. Distributed Learning with Stochastic Gradient Methods

In this section, we first state the distributed SGM we study and discuss its numerical realization. We then present theoretical results on generalization properties for distributed SGM and non-distributed SGM, following with simple discussions.

3.1. Distributed SGM and Numerical Realization

Throughout this paper, as that in (Zhang et al., 2015), we assume that[1] the sample size $N = mn$ for some positive integers n, m, and we randomly decompose $\bar{\mathbf{z}}$ as $\mathbf{z}_1 \cup \mathbf{z}_2 \cup \cdots \cup \mathbf{z}_m$ with $|\mathbf{z}_1| = |\mathbf{z}_2| = \cdots = |\mathbf{z}_m| = n$. For any $s \in [m]$, we write $\mathbf{z}_s = \{(x_{s,i}, y_{s,i})\}_{i=1}^n$. We study the following distributed SGM, with mini-batches and multi-pass over the data. For any $t \in \mathbb{R}$, the set $\{1, ..., t\}$ of the first t positive integers is denoted by $[t]$.

Algorithm 1. *Let $b \in [n]$. The b-minibatch stochastic gradient methods over the sample $\mathbf{z}_s$ is defined by $f_{s,1} = 0$ and for all $t \in [T]$,*

$$f_{s,t+1} = f_{s,t} - \eta_t \frac{1}{b} \sum_{i=b(t-1)+1}^{bt} (f_{s,t}(x_{s,j_{s,i}}) - y_{s,j_{s,i}}) K_{x_{s,j_{s,i}}}, \tag{4}$$

where $\{\eta_t > 0\}$ is a step-size sequence. Here, $j_{s,1}, j_{s,2}, \cdots, j_{s,bT}$ are i.i.d. random variables from the uniform distribution on $[n]$.[2] The global predictor averaging over these local estimators is given by

$$\bar{f}_t = \frac{1}{m} \sum_{s=1}^m f_{s,t}.$$

In the above algorithm, at each iteration t, for each $s \in [m]$, the local estimator updates its current solution by subtracting a scaled gradient estimate. It is easy to see that the gradient estimate at each iteration for the s-th local estimator is an unbiased estimate of the full gradient of the empirical risk over $\mathbf{z}_s$. The global predictor is the average over these local solutions. In the special case $m = 1$, the algorithm reduces to the classic multi-pass SGM studied in (Lin & Rosasco, 2017b).

There are several free parameters in the algorithm, the step-size η_t, the mini-batch size b, the total number of iterations/passes, and the number of partition/subsets m. All these parameters will affect the algorithm's generalization properties and computational complexity. In the coming subsection, we will show how these parameters can be chosen so that the algorithm can generalize optimally, as long as the number of subsets m is not too large. Different choices on η_t, b, and T correspond to different regularization strategies. In this paper, we are particularly interested in the cases that both η_t and b are fixed as some universal constants that may depend on the local sample size n, while T is tuned.

The total number of iterations T for each local estimator can be bigger than the local sample size n, which means that the algorithm can use the data more than once, or in another words, we can run the algorithm with multiple passes over the data. Here and in what follows, the number of (effective) 'passes' over the data is referred to $\frac{bt}{n}$ after t iterations of the algorithm.

For any finite subsets $\mathbf{x}$ and $\mathbf{x}'$ in X, denote the $|\mathbf{x}| \times |\mathbf{x}'|$ kernel matrix $[K(x, x')]_{x \in \mathbf{x}, x' \in \mathbf{x}'}$ by $\mathbf{K}_{\mathbf{x}\mathbf{x}'}$. Obviously, using an inductive argument, one can prove that Algorithm 1 is equivalent to

$$\bar{f}_t = \frac{1}{m} \sum_{s=1}^m \sum_{i=1}^n \mathbf{b}_{s,t}(i) K_{x_{s,i}},$$

where for all $s \in [m]$, $\mathbf{b}_{s,t} = [\mathbf{b}_{s,t}(1), \cdots, \mathbf{b}_{s,t}(n)]^\top \in \mathbb{R}^n$ and it is generated by, with $\mathbf{b}_{s,1} = \mathbf{0} \in \mathbb{R}^n$, for all $t \in [T]$,

$$\mathbf{b}_{s,t+1} = \mathbf{b}_{s,t} - \frac{\eta_t}{b} \sum_{i=b(t-1)+1}^{bt} (\mathbf{b}_{s,t}^\top \mathbf{K}_{\mathbf{x}_s x_{s,j_{s,i}}} - y_{s,j_{s,i}}) \mathbf{e}_{j_{s,i}}. \tag{5}$$

Here, $\mathbf{e}_1, \cdots, \mathbf{e}_n$ are standard basis of $\mathbb{R}^n$. The space and time complexities for each local estimator are

$$O(n) \quad \text{and} \quad O(bnT), \tag{6}$$

respectively. The total space and time complexities of the algorithm are

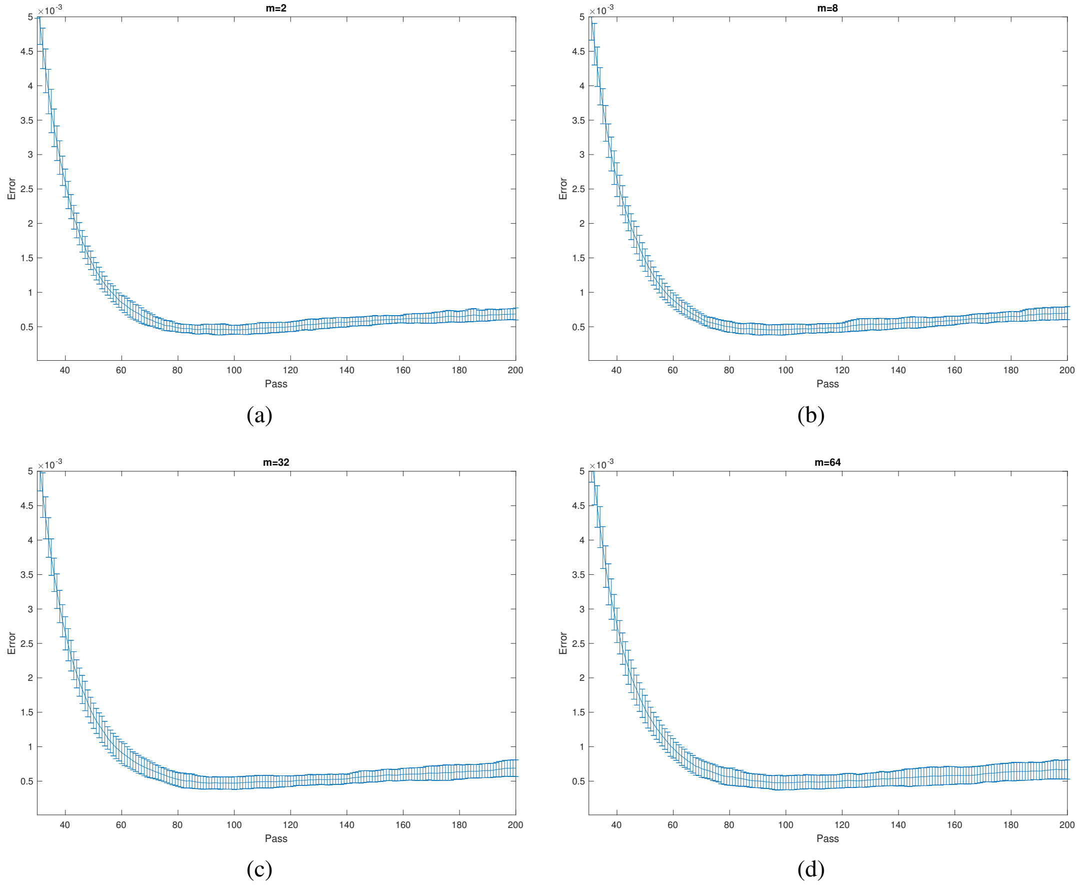
$$O(N) \quad \text{and} \quad O(bNT), \quad \text{respectively.} \tag{7}$$

In order to see the empirical performance of the studied algorithm, we carried out some numerical simulations on a non-parametric regression problem with simulated datasets. We constructed a training data $\{(x_i, y_i)\}_{i=1}^N \subseteq \mathbb{R} \times \mathbb{R}$ with $N = 2^{12}$ from the regression model $y = f_\rho(x) + \xi$, where the regression function $f_\rho(x) = |x - 1/2| - 1/2$, the input x is uniformly drawn from $[0, 1]$, and ξ is a Gaussian noise with zero mean and standard deviation 1. In all the simulations, the RKHS is associated with a Gaussian kernel $K(x, x') = \exp(-\frac{|x - x'|^2}{2\sigma^2})$ where $\sigma = 0.2$, and the mini-batch size $b = 1$. For each number of partitions $m \in \{2, 8, 32, 64\}$, we set the step-size as $\eta_t = \frac{1}{8n}$ as that suggested by Part 1) of Corollary 2 in the coming subsection[3], and executed simulation 50 times. In each trial, an approximated generalization error is computed over an empirical measure with 1000 points. The mean and the standard deviation of these computed generalization errors

[1] For the general case, one can consider the weighted averaging scheme, as that in (Lin et al., 2017), and our analysis still applies with a simple modification.

[2] Note that the random variables $j_{s,1}, \cdots, j_{s,bT}$ are conditionally independent given the sample $\mathbf{z}_\mathbf{s}$.

[3] It would be interesting to run the algorithm with other step-sizes suggested by Corollary 2.



(a)

(b)

(c)

(d)

Figure 1. Approximated Generalization Errors for Distributed SGM with Different Partition Levels $m = \{2, 8, 32, 64\}$.

over 50 trials with respect to the number of passes are depicted in the above figures. As we can see from the figures, distributed SGM performs well, and after some number of passes, it achieves the minimal (approximated) generalization error. As the number of subsets m increases, the error and the number of passes to reach minimal error will also slightly increase. Note that the computational cost for n iteration (one pass) of the global estimator is $O(N^2/m)$. Thus the total computational cost for the algorithm to reach minimal error would be reduced if one enlarges the number of partition m. Finally, the accuracy is comparable with 0.809×10^{-3} of KRR with cross validation.

3.2. Generalization Properties for Distributed Stochastic Gradient Methods

In this section, we state our theoretical results on generalization error for distributed SGM, following with simple discussions. To do so, we need to introduce some basic assumptions. Throughout this paper, we make the following two basic assumptions.

Assumption 1. *H is separable, K is measurable and furthermore, there exists a constant $\kappa \in [1, \infty[$, such that for all $x \in X$,*

$$K(x, x) \leq \kappa^2. \tag{8}$$

Assumption 2. *For some $M, \sigma \geq 0$,*

$$\int_Y y^2 d\rho(y|x) \leq M$$

and

$$\int_Y (f_\rho(x) - y)^2 d\rho(y|x) \leq \sigma^2, \tag{9}$$

ρ_X-almost surely.

The above two assumptions are quite common in statistical learning theory, see, e.g., (Steinwart & Christmann, 2008; Cucker & Zhou, 2007). The constant σ from Equation (9) measures the noise level of the studied problem. The condition $\int_Y y^2 d\rho(y|x) \leq M$ implies that the regression function is bounded almost surely,

$$|f_\rho(x)| \leq M. \tag{10}$$

It is trivially satisfied when the output domain Y is bounded, for example, $Y = \{-1, 1\}$ in the classification problem.

Corollary 1. *Assume that $f_\rho \in H$ and*

$$m \leq N^{\beta}, \quad 0 \leq \beta < \frac{1}{2}.$$

Consider Algorithm 1 with any of the following choices on η_t, b and T.
1) $\eta_t \simeq m/\sqrt{N}$ for all $t \in [T_*]$, $b = 1$, and $T_* \simeq N/m$.
2) $\eta_t \simeq \frac{1}{\log N}$ for all $t \in [T_*]$, $b \simeq \sqrt{N}/m$, and $T_* \simeq \sqrt{N}\log N$.
Then,

$$\mathbb{E}\mathcal{E}(\bar{f}_{t+1}) - \mathcal{E}(f_\rho) \lesssim N^{-1/2}\log N.$$

Here, we use the notations $a_1 \lesssim a_2$ to mean $a_1 \leq Ca_2$ for some positive constant C which is depending only on (a polynomial function) $\kappa, M, \sigma, \|\mathcal{T}\|, \|f_\rho\|_H$, and $a_1 \simeq a_2$ to mean $a_2 \lesssim a_1 \lesssim a_2$.

The above result provides generalization error bounds for distributed SGM with two different choices on step-size η_t, mini-batch size b and total number of iterations/passes. The convergence rate is optimal up to a logarithmic factor, in the sense that it matches the minimax rate in (Caponnetto & De Vito, 2007) and the convergence rate for KRR (Smale & Zhou, 2007; Caponnetto & De Vito, 2007). The number of passes to achieve optimal error bounds in both cases is roughly one. The above result asserts that distributed SGM generalizes optimally after one pass over the data for two different choices on step-size and mini-batch size, provided that the partition level m is not too large. In the case that $m \simeq \sqrt{N}$, according to (7), the computational complexities are $O(N)$ in space and $O(N^{1.5})$ in time, comparing with $O(N)$ in space and $O(N^2)$ in time of classic SGM.
Corollary 1 provides statistical results on generalization error bounds with a convergence rate of order $O(N^{-1/2}\log N)$ for distributed SGM. It does not consider any benign assumptions about the learning problem, such as the regularity of the regression function and the capacity of the RKHS. In what follows, we will show how the convergence rate can be further improved, if we make two benign assumptions of the learning problem.
The first benign assumption relates to the regularity of the regression function. We introduce the integer operator $\mathcal{L} : L^2_{\rho_X} \to L^2_{\rho_X}$, defined by $\mathcal{L}f = \int_X f(x)K(x,\cdot)d\rho_X$. Under Assumption (8), $\mathcal{L}$ is positive trace class operators (Cucker & Zhou, 2007), and hence $\mathcal{L}^\zeta$ is well defined using the spectral theory.

Assumption 3. *There exist $\zeta > 0$ and $R > 0$, such that $\|\mathcal{L}^{-\zeta} f_\rho\|_\rho \leq R$.*

This assumption characterizes how large the subspace that the regression function lies in. The bigger the ζ is, the smaller the subspace is, the stronger the assumption is, and the easier the learning problem is, as $\mathcal{L}^{\zeta_1}(L^2_{\rho_X}) \subseteq \mathcal{L}^{\zeta_2}(L^2_{\rho_X})$ if $\zeta_1 \geq \zeta_2$. Moreover, if $\zeta = 0$, we are making no assumption, and if $\zeta = \frac{1}{2}$, we are requiring that there exists some $f_* \in H$ such that $f_H = f_\rho$ almost surely (Steinwart & Christmann, 2008).
The next assumption relates to the capacity of the hypothesis space.

Assumption 4. *For some $\gamma \in [0,1]$ and $c_\gamma > 0$, $\mathcal{L}$ satisfies*

$$\operatorname{tr}(\mathcal{L}(\mathcal{L}+\lambda I)^{-1}) \leq c_\gamma \lambda^{-\gamma}, \quad \text{for all } \lambda > 0. \tag{11}$$

The left hand-side of (11) is called effective dimension (Zhang, 2005) or degrees of freedom (Caponnetto & De Vito, 2007). It is related to covering/entropy number conditions, see (Steinwart & Christmann, 2008). The condition (11) is naturally satisfied with $\gamma = 1$, since $\mathcal{L}$ is a trace class operator which implies that its eigenvalues $\{\sigma_i\}_i$ satisfy $\sigma_i \lesssim i^{-1}$. Moreover, if the eigenvalues of $\mathcal{L}$ satisfy a polynomial decaying condition $\sigma_i \sim i^{-c}$ for some $c > 1$, or if $\mathcal{L}$ is of finite rank, then the condition (11) holds with $\gamma = 1/c$, or with $\gamma = 0$. The case $\gamma = 1$ is refereed as the capacity independent case. A smaller γ allows deriving faster convergence rates for the studied algorithms, as will be shown in the following results.
Making these two assumptions, we have the following general results on generalization error for the studied algorithms.

Theorem 1. *Under Assumptions 3 and 4, let $\zeta \leq 1$ and $\eta_t = \eta$ for all $t \in [T]$ with η satisfying*

$$0 < \eta \leq \frac{1}{4\kappa^2 \log T}. \tag{12}$$

Then for all $t \in [T]$ and any $\tilde{\lambda} = n^{\theta-1}$ with $\theta \in [0,1]$,

$$\mathbb{E}\mathcal{E}(\bar{f}_{t+1}) - \mathcal{E}(f_\rho) \lesssim [\frac{1}{(\eta t)^{2\zeta}} + \frac{1}{N\tilde{\lambda}^{\gamma}} + \frac{\eta}{mb}] \\ \times ((\tilde{\lambda}\eta t)^2 \vee [\gamma(\theta^{-1} \wedge \log n)]^{2\zeta \vee 1} \vee 1 \vee \log t). \tag{13}$$

Here and throughout the rest of this paper, we use the notation $a_1 \lesssim a_2$ to mean $a_1 \leq Ca_2$ for some positive constant C which is depending only on $\kappa, M, \zeta, R, \gamma, C_\gamma, \sigma$ and $\|\mathcal{T}\|$.

In the above result, we only consider the setting of a fixed step-size. Results with a decaying step-size can be directly derived following our proofs in the coming sections, combining with some basic estimates from (Lin & Rosasco, 2017b). The derived error bound from (13) depends on the number of iteration t, the step-size η, the mini-batch size, the number of sample points N and the partition level m. It holds for any pseudo regularization parameter $\tilde{\lambda}$ where $\tilde{\lambda} \in [n^{-1}, 1]$. When $t \leq n/\eta$, we can choose $\tilde{\lambda} = (\eta t)^{-1}$, and ignoring the logarithmic factor, (13) reads as

$$\mathbb{E}\mathcal{E}(\bar{f}_{t+1}) - \mathcal{E}(f_\rho) \lesssim \frac{1}{(\eta t)^{2\zeta}} + \frac{(\eta t)^{\gamma}}{N} + \frac{\eta}{mb}. \tag{14}$$

The right-hand side of the above inequality is composed of three terms. The first term is related to the regularity parameter ζ of the regression function f_ρ, and it results from estimating bias. The second term depends on the sample size N, and it results from estimating sample variance. The last term results from estimating computational variance due to random choices of the sample points. In comparing with the error bounds derived for classic SGM performed on a local machine, one can see that averaging over the local solutions can reduce sample and computational variances, but keeps bias unchanged. As the number of iteration t increases, the bias term decreases, and the sample variance term increases. This is a so-called trade-off problem in statistical learning theory. Solving this trade-off problem leads to the best choice on number of iterations. Notice that the computational variance term is independent of the number of iterations t and it depends on the step-size, the mini-batch size, and the partition level. To derive optimal rates, it is necessary to choose a small step-size, and/or a large mini-batch size, and a suitable partition level. In what follows, we provide different choices of these algorithmic parameters, corresponding to different regularization strategies, while leading to the same optimal convergence rates.

Corollary 2. *Under Assumptions 3 and 4, let $\zeta \le 1$, $2\zeta + \gamma > 1$ and*

$$m \le N^\beta, \quad \text{with } 0 \le \beta < \frac{2\zeta+\gamma-1}{2\zeta+\gamma}. \tag{15}$$

Consider Algorithm 1 with any of the following choices on η_t, b and T.
1) $\eta_t \simeq n^{-1}$ for all $t \in [T_*]$, $b = 1$, and $T_* \simeq N^{\frac{1}{2\zeta+\gamma}} n$.
2) $\eta_t \simeq n^{-1/2}$ for all $t \in [T_*]$, $b \simeq \sqrt{n}$, and $T_* \simeq N^{\frac{1}{2\zeta+\gamma}} \sqrt{n}$.
3) $\eta_t \simeq N^{-\frac{2\zeta}{2\zeta+\gamma}} m$ for all $t \in [T_*]$, $b = 1$, and $T_* \simeq N^{\frac{2\zeta+1}{2\zeta+\gamma}}/m$.
4) $\eta_t \simeq \frac{1}{\log N}$ for all $t \in [T_*]$, $b \simeq N^{\frac{2\zeta}{2\zeta+\gamma}}/m$, and $T_* \simeq N^{\frac{1}{2\zeta+\gamma}} \log N$.
Then,

$$\mathbb{E}\mathcal{E}(\bar{f}_{T_*+1}) - \mathcal{E}(f_\rho) \lesssim N^{-\frac{2\zeta}{2\zeta+\gamma}} \log N.$$

We add some comments on the above theorem. First, the convergence rate is optimal, as it is the same as that for KRR from (Caponnetto & De Vito, 2007; Smale & Zhou, 2007) and also it matches the minimax rate in (Caponnetto & De Vito, 2007), up to a logarithmic factor. Second, distributed SGM saturates when $\zeta > 1$. The reason for this is that averaging over local solutions can only reduce sample and computational variances, not bias. Similar saturation phenomenon is also observed when analyzing distributed KRR in (Zhang et al., 2015; Lin et al., 2017). Third, the condition $2\zeta + \gamma > 1$ is equivalent to assuming that the learning problem can not be too difficult. We believe that such a condition is necessary for applying distributed learning technique to reduce computational costs, as there are no means to reduce computational costs if the learning problem itself is not easy. Fourth, as the learning problem becomes easier (corresponds to a bigger ζ), the faster the convergence rate is, and moreover the larger the number of partition m can be. Finally, different parameter choices leads to different regularization strategies. In the first two regimes, the step-size and the mini-batch size are fixed as some prior constants (which only depends on n), while the number of iterations depends on some unknown distribution parameters. In this case, the regularization parameter is the number of iterations, which in practice can be tuned by using cross-validation methods. Besides, the step-size and the number of iterations in the third regime, or the mini-batch size and the number of iterations in the last regime, depend on the unknown distribution parameters, and they have some regularization effects. The above theorem asserts that distributed SGM with differently suitable choices of parameters can generalize optimally, provided the partition level m is not too large.

3.3. Optimal Convergence for Multi-pass SGM on a Single Dataset

As a byproduct of our new analysis in the coming sections, we derive the following results for classic multi-pass SGM.

Theorem 2. *Under Assumptions 3 and 4, consider Algorithm 1 with $m = 1$ and any of the following choices on η_t, b and T.*
1) $\eta_t \simeq N^{-1}$ for all $t \in [T_*]$, $b = 1$, and $T_* \simeq N^{\alpha+1}$.
2) $\eta_t \simeq N^{-1/2}$ for all $t \in [T_*]$, $b \simeq \sqrt{N}$, and $T_* \simeq N^{\alpha+1/2}$.
3) $\eta_t \simeq N^{-2\zeta\alpha}$ for all $t \in [T_*]$, $b = 1$, and $T_* \simeq N^{\alpha(2\zeta+1)}$.
4) $\eta_t \simeq \frac{1}{\log N}$ for all $t \in [T_*]$, $b \simeq N^{2\zeta\alpha}$, and $T_* \simeq N^\alpha \log T$.
Here, $\alpha = \frac{1}{(2\zeta+\gamma)\vee 1}$. Then,

$$\mathbb{E}\mathcal{E}(\bar{f}_{t+1}) - \mathcal{E}(f_\rho) \lesssim \begin{cases} N^{-\frac{2\zeta}{2\zeta+\gamma}} \log N, & \text{if } 2\zeta+\gamma > 1; \\ N^{-2\zeta} \log N, & \text{otherwise}. \end{cases} \tag{16}$$

The above results provide generalization error bounds for multi-pass SGM trained on a single dataset. The derived convergence rate is optimal in the minimax sense (Caponnetto & De Vito, 2007; Blanchard & Mucke, 2016a). Note that SGM does not have a saturation effect, and optimal convergence rates can be derived for any $\zeta \in]0, \infty]$. Theorem 2 improves the result in (Lin & Rosasco, 2017b) in two aspects. First, the convergence rates are better than those (i.e., $O(N^{-\frac{2\zeta}{2\zeta+\gamma}} \log N)$ if $2\zeta + \gamma \ge 1$ or $O(N^{-2\zeta} \log^4 N)$ otherwise) from (Lin & Rosasco, 2017b). Second, the above theorem does not require the extra condition $m \ge m_\delta$ made in (Lin & Rosasco, 2017b).

3.4. Error Decomposition

The key to our proof is an error decomposition. To introduce the error decomposition, we need to introduce two auxiliary sequences.
The first auxiliary sequence is generated by distributed GM.

Algorithm 2. *For any $s \in [m]$, the GM over the sample set $\mathbf{z}_s$ is defined by $g_{s,1} = 0$ and for $t = 1, \cdots, T$,*

$$g_{s,t+1} = g_{s,t} - \eta_t \frac{1}{n} \sum_{i=1}^{n} (g_{s,t}(x_{s,i}) - y_{s,i}) K_{x_{s,i}}, \tag{17}$$

where $\{\eta_t > 0\}$ is a step-size sequence given by Algorithm 1. The average estimator over these local estimators is given by

$$\bar{g}_t = \frac{1}{m} \sum_{s=1}^{m} g_{s,t}. \tag{18}$$

The second auxiliary sequence is generated by distributed pseudo GM as follows.

Algorithm 3. *For any $s \in [m]$, the pseudo GM over the input set $\mathbf{x}_s$ is defined by $h_{s,1} = 0$ and for $t = 1, \cdots, T$,*

$$h_{s,t+1} = h_{s,t} - \eta_t \frac{1}{n} \sum_{i=1}^{n} (h_{s,t}(x_{s,i}) - f_\rho(x_{s,i})) K_{x_{s,i}}, \tag{19}$$

where $\{\eta_t > 0\}$ is a step-size sequence given by Algorithm 1. The average estimator over these local estimators is given by

$$\bar{h}_t = \frac{1}{m} \sum_{s=1}^{m} h_{s,t}. \tag{20}$$

Note that Algorithm (19) can not be implemented in practice, as $f_\rho(x)$ is unknown in general.

For any $s \in [m]$, using an inductive argument, one can prove that (Lin & Rosasco, 2017b)

$$\mathbb{E}_{\mathbf{J}_s|\mathbf{z}_s}[f_{s,t}] = g_{s,t}. \tag{21}$$

Here $\mathbb{E}_{\mathbf{J}_s|\mathbf{z}_s}$ (or abbreviated as $\mathbb{E}_{\mathbf{J}_s}$) denotes the conditional expectation with respect to $\mathbf{J}_s$ given $\mathbf{z}_s$. Similarly, using the definition of the regression function (2) and an inductive argument, one can also prove that

$$\mathbb{E}_{\mathbf{y}_s}[g_{s,t}] = h_{s,t}. \tag{22}$$

With the above two equalities, we can prove and the following error decomposition. We introduce the inclusion operator $\mathcal{S}_\rho : H \to L^2_{\rho_X}$.

Proposition 1. *We have that for any $t \in [T]$,*

$$\mathbb{E}\mathcal{E}(\bar{f}_t) - \mathcal{E}(f_\rho) = \mathbb{E}\|\mathcal{S}_\rho \bar{h}_t - f_\rho\|_\rho^2 + \mathbb{E}[\|\mathcal{S}_\rho(\bar{g}_t - \bar{h}_t)\|_\rho^2] + \mathbb{E}\|\mathcal{S}_\rho(\bar{f}_t - \bar{g}_t)\|_\rho^2. \tag{23}$$

The error decomposition is similar as the one given in (Lin & Rosasco, 2017b) for classic multi-pass SGM. There are three terms in the right-hand side of (23). The first term depends on the regularity of the regression function (Assumption 3) and it is called as *bias*. The second term depends on the noise level σ^2 from (9) and it is called as *sample variance*. The last term is caused by the random estimates of the full gradients and it is called as *computational variance*. In the appendix, we will estimate these three terms separately. Total error bounds can be thus derived by substituting these estimates into the error decomposition.

4. Discussion

We briefly review convergence results for SGM. SGM (Robbins & Monro, 1951) has been widely used in convex optimization and machine learning, see e.g. (Cesa-Bianchi et al., 2004; Nemirovski et al., 2009; Bottou et al., 2016) and references therein. In what follows, we will briefly recall some recent works on generalization error for nonparametric regression on a RKHS considering the square loss. We will use the term "online learning algorithm" (OL) to mean one-pass SGM, i.e, SGM that each sample can be used only once. Different variants of OL, either with or without regularization, have been studied. Most of them take the form

$$f_{t+1} = (1 - \lambda_t) f_t - \eta_t (f_t(x_t) - y_t) K_{x_t}, t = 1 \cdots, N.$$

Here, the regularization parameter λ_t could be zero (Zhang, 2004; Ying & Pontil, 2008), or a positive (Smale & Yao, 2006; Ying & Pontil, 2008) and possibly time-varying constant (Tarres & Yao, 2014). Particularly, (Tarres & Yao, 2014) studied OL with time-varying regularization parameters and convergence rate of order $O(N^{\frac{-2\zeta}{2\zeta+1}})$ $(\zeta \in [\frac{1}{2}, 1])$ in high probability was proved. (Ying & Pontil, 2008) studied OL without regularization and convergence rate of order $O(N^{-\frac{2\zeta}{2\zeta+1}})$ in expectation was shown. Both convergence rates from (Ying & Pontil, 2008; Tarres & Yao, 2014) are capacity-independently optimal and they do not take the capacity assumption into account. Considering an averaging step (Polyak & Juditsky, 1992) and a proof technique motivated by (Bach & Moulines, 2013), (Dieuleveut & Bach, 2016) proved capacity-dependently optimal rate $O(N^{-\frac{2\zeta}{(2\zeta+\gamma)\vee 1}})$ for OL in the case that $\zeta \le 1$. Recently, (Lin & Rosasco, 2017b) studied (multi-pass) SGM, i.e, Algorithm 1 with $m = 1$. They showed that SGM with suitable parameter choices, achieves convergence rate of order $O(N^{-\frac{2\alpha}{(2\alpha+\gamma)\vee 1}} \log^\beta N)$ with $\beta = 2$ when $2\alpha + \gamma > 1$ or $\beta = 4$ otherwise, after some number of iterations. In comparisons, the derived results for SGM in Theorem 2 are better than those from (Lin & Rosasco, 2017b), and the convergence rates are the same as those from (Dieuleveut & Bach, 2016) for averaging OL when

$\zeta \leq 1$ and $2\zeta + \gamma \geq 1$. For the case $2\zeta + \gamma \leq 1$, the convergence rate $O(N^{-2\zeta}(1 \vee \log N^{\gamma}))$ for SGM in Theorem 2 is worser than $O(N^{-2\zeta})$ in (Dieuleveut & Bach, 2016) for averaging OL. However, averaging OL saturates for $\zeta > 1$, while SGM does not.

To meet the challenge of large-scale learning, a line of research focus on designing learning algorithms with Nyström subsampling, or more generally sketching. Interestingly, the latter has also been applied to compressed sensing, low rank matrix recovery and kernel methods, see e.g. (Candès et al., 2006; Yurtsever et al., 2017; Yang et al., 2012) and references therein. The basic idea of Nyström subsampling is to replace a standard large matrix with a smaller matrix obtained by subsampling (Smola & Schölkopf, 2000; Williams & Seeger, 2000). For kernel methods, Nyström subsampling has been successfully combined with KRR (Alaoui & Mahoney, 2015; Rudi et al., 2015; Yang et al., 2017) and SGM (Lu et al., 2016; Lin & Rosasco, 2017a). Generalization error bounds of order $O(N^{\frac{-2\zeta}{2\zeta+\gamma}})$ (Rudi et al., 2015; Lin & Rosasco, 2017a) were derived, provided that the subsampling level is suitably chosen, considering the case $\zeta \in [\frac{1}{2}, 1]$. Computational advantages of these algorithms were highlighted. Here, we summarize their computational costs in Table 1, from which we see that distributed SGM has advantages on both memory and time.

Table 1. Summary of assumptions and costs for distributed SGM (DSGM), KRR, GM, one-pass SGM with averaging (AveOL), SGM, Nyström KRR (NyKRR), Nyström SGM (NySGM), and distributed KRR (DKRR).

Alg.	Ass. (ζ/γ)	Space/Time
KRR	$[\frac{1}{2}, 1],]0, 1]$	$N^2 \& N^3$
GM	$[0, \infty[, [0, 1]$	$N \ \& \ N^2 N^{\frac{1}{2\zeta+2}}$
AveOL	$[0, 1], [0, 1]$	$N \ \& \ N^2$
SGM	$[0, \infty[, [0, 1]$	$N \ \& \ N^2 N^{\frac{1-\gamma}{2\zeta+\gamma}}$
NyKRR	$[\frac{1}{2}, 1],]0, 1]$	$N^{\frac{2\zeta+\gamma+1}{2\zeta+\gamma}} \ \& \ N^{\frac{2\zeta+2+\gamma}{2\zeta+\gamma}}$
NySGM	$[\frac{1}{2}, 1],]0, 1]$	$N^{\frac{2}{2\zeta+\gamma} \vee 1} \ \& \ N^{\frac{2\zeta+2}{2\zeta+\gamma}}$
DKRR	$[\frac{1}{2}, 1],]0, 1]$	$N^{\frac{2\zeta+2\gamma+1}{2\zeta+\gamma}} \ \& \ N^{\frac{2\zeta+2+3\gamma}{2\zeta+\gamma}}$
DSGM	$[0, 1], [0, 1]$	$\boldsymbol{N \ \& \ N^{\frac{2\zeta+\gamma+1}{2\zeta+\gamma}}}$

Note: 1) For AveOL and DSGM, $2\zeta + \gamma > 1$. 2) The costs here for the distributed algorithms are the costs of running the distributed algorithms on a single machine.

Another line of research for large-scale learning focus on distributed (parallelizing) learning. Distributed learning, based on a divide-and-conquer approach, has been used for, e.g., perceptron-based algorithms (Mcdonald et al., 2009), parametric smooth convex optimization problems (Zhang et al., 2012), and sparse regression (Lee et al., 2017). Recently, this approach has been successfully applied to learning algorithms with kernel methods, such as KRR (Zhang et al., 2015), and SRA (Guo et al., 2017; Blanchard & Mucke, 2016a). (Zhang et al., 2015) first studied distributed KRR and showed that distributed KRR retains optimal rates $O(N^{-\frac{2\zeta}{2\zeta+\gamma}})$ (for $\zeta \in [\frac{1}{2}, 1]$) provided the partition level is not too large. The number of partition to retain optimal rate shown in (Zhang et al., 2015) for distributed KRR depends on some conditions which may be less well understood and thus potentially leads to a suboptimal partition number. (Lin et al., 2017) provided an alternative and refined analysis for distributed KRR, leading to a less strict condition on the partition number. (Guo et al., 2017) extended the analysis to distributed SRA, an proved optimal convergence rate for the case $\zeta \geq 1/2$, if the number of partitions $m \leq N^{\frac{2\zeta-1}{2\zeta+\gamma}}$. In comparison, the condition on partition number from Corollary 2 for distributed SGM is less strict. Moreover, Corollary 2 shows that distributed SGM can retain optimal rate even in the non-attainable case. According to Corollary 2, distributed SGM with appropriate choices of parameters can achieve optimal rate if the partition number is not too large. In comparison of the derived results for distributed KRR with those for distributed SGM, we see from Table 1 that the latter has advantages on both memory and time. The most related to our works are (Zinkevich et al., 2010; Jain et al., 2016). (Zinkevich et al., 2010) studied distributed OL for optimization problems over a finite-dimensional domain, and proved convergence results assuming that the objective function is strongly convex. (Jain et al., 2016) considered distributed OL with averaging for least square regression problems over a finite-dimension space and proved certain convergence results that may depend on the smallest eigenvalue of the covariance matrix. These results do not apply to our cases, as we consider distributed multi-pass SGM for nonparametric regression over a RKHS and our objective function is not strongly convex. We finally remark that using a partition approach (Thomann et al., 2016; Tandon et al., 2016), one can also scale up the kernel methods, with a computational advantage similar as those of using distributed learning technique.

We conclude this section with some further questions. First, in this paper, we assume that all parameter choices are given priorly. In practice, these parameters can be possibly tuned by cross-validation method. Second, the derived rate for SGM in the case $2\zeta + \gamma \leq 1$ is $O(N^{-2\zeta}(1 \vee \log N^{\gamma}))$, which is worser than $O(N^{-2\zeta})$ of averaging OL (Dieuleveut & Bach, 2016). It would be interesting to improve the rate, or to derive a minimax rate for the case $2\zeta + \gamma \leq 1$. Third, all results stated in this paper are in expectation, and it would be interesting to derive high-probability results in the future (and possibly by a proof technique from (London, 2017)).

Acknowledgements

This work was sponsored by the Department of the Navy, Office of Naval Research (ONR) under a grant number N62909-17-1-2111. It has also received funding from Hasler Foundation Program: Cyber Human Systems (project number 16066), and from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program (grant agreement n 725594-time-data).

References

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

Bach, Francis and Moulines, Eric. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Advances in Neural Information Processing Systems*, pp. 773–781, 2013.

Bauer, Frank, Pereverzev, Sergei, and Rosasco, Lorenzo. On regularization algorithms in learning theory. *Journal of Complexity*, 23(1):52–72, 2007.

Blanchard, Gilles and Krämer, Nicole. Optimal learning rates for kernel conjugate gradient regression. In *Advances in Neural Information Processing Systems*, pp. 226–234, 2010.

Blanchard, Gilles and Mucke, Nicole. Optimal rates for regularization of statistical inverse learning problems. *arXiv preprint arXiv:1604.04054*, 2016a.

Blanchard, Gilles and Mucke, Nicole. Parallelizing spectral algorithms for kernel learning. *arXiv preprint arXiv:1610.07487*, 2016b.

Bottou, Leon, Curtis, Frank E, and Nocedal, Jorge. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*, 2016.

Candès, Emmanuel J, Romberg, Justin, and Tao, Terence. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information theory*, 52(2):489–509, 2006.

Caponnetto, Andrea. Optimal learning rates for regularization operators in learning theory. *Technical report*, 2006.

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

Cesa-Bianchi, Nicolo, Conconi, Alex, and Gentile, Claudio. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Cucker, Felipe and Zhou, Ding Xuan. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.

Dicker, Lee H, Foster, Dean P, and Hsu, Daniel. Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators. *Electronic Journal of Statistics*, 11(1):1022–1047, 2017.

Dieuleveut, Aymeric and Bach, Francis. Nonparametric stochastic approximation with large step-sizes. *The Annals of Statistics*, 44(4):1363–1399, 2016.

Engl, Heinz Werner, Hanke, Martin, and Neubauer, Andreas. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

Fujii, Junichi, Fujii, Masatoshi, Furuta, Takayuki, and Nakamoto, Ritsuo. Norm inequalities equivalent to heinz inequality. *Proceedings of the American Mathematical Society*, 118(3):827–830, 1993.

Gerfo, L Lo, Rosasco, Lorenzo, Odone, Francesca, De Vito, Ernesto, and Verri, Alessandro. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008.

Guo, Zheng-Chu, Lin, Shao-Bo, and Zhou, Ding-Xuan. Learning theory of distributed spectral algorithms. *Inverse Problems*, 2017.

Jain, Prateek, Kakade, Sham M, Kidambi, Rahul, Netrapalli, Praneeth, and Sidford, Aaron. Parallelizing stochastic approximation through mini-batching and tail-averaging. *arXiv preprint arXiv:1610.03774*, 2016.

Lee, Jason D, Liu, Qiang, Sun, Yuekai, and Taylor, Jonathan E. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(5):1–30, 2017.

Lin, Junhong and Rosasco, Lorenzo. Optimal rates for learning with Nyström stochastic gradient methods. *arXiv preprint arXiv:1710.07797*, 2017a.

Lin, Junhong and Rosasco, Lorenzo. Optimal rates for multi-pass stochastic gradient methods. *Journal of Machine Learning Research*, 18(97):1–47, 2017b.

Lin, Shao-Bo, Guo, Xin, and Zhou, Ding-Xuan. Distributed learning with regularized least squares. *Journal of Machine Learning Research*, 18(92):1–31, 2017.

London, Ben. A PAC-bayesian analysis of randomized learning with application to stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 2935–2944, 2017.

Lu, Jing, Hoi, Steven CH, Wang, Jialei, Zhao, Peilin, and Liu, Zhi-Yong. Large scale online kernel learning. *Journal of Machine Learning Research*, 17(47):1, 2016.

Mathé, Peter and Pereverzev, Sergei V. Moduli of continuity for operator valued functions. 2002.

Mcdonald, Ryan, Mohri, Mehryar, Silberman, Nathan, Walker, Dan, and Mann, Gideon S. Efficient large-scale distributed training of conditional maximum entropy models. In *Advances in Neural Information Processing Systems*, pp. 1231–1239, 2009.

Minsker, Stanislav. On some extensions of bernstein's inequality for self-adjoint operators. *arXiv preprint arXiv:1112.5448*, 2011.

Nemirovski, Arkadi, Juditsky, Anatoli, Lan, Guanghui, and Shapiro, Alexander. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19 (4):1574–1609, 2009.

Pinelis, IF and Sakhanenko, AI. Remarks on inequalities for large deviation probabilities. *Theory of Probability & Its Applications*, 30(1):143–148, 1986.

Polyak, Boris T and Juditsky, Anatoli B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Rosasco, Lorenzo and Villa, Silvia. Learning with incremental iterative regularization. In *Advances in Neural Information Processing Systems*, pp. 1630–1638, 2015.

Rudi, Alessandro, Camoriano, Raffaello, and Rosasco, Lorenzo. Less is more: Nyström computational regularization. *Advances in Neural Information Processing Systems*, pp. 1657–1665, 2015.

Schölkopf, Bernhard and Smola, Alexander J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Shawe-Taylor, John and Cristianini, Nello. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Smale, Steve and Yao, Yuan. Online learning algorithms. *Foundations of Computational Mathematics*, 6(2):145–170, 2006.

Smale, Steve and Zhou, Ding-Xuan. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26(2):153–172, 2007.

Smola, Alex J and Schölkopf, Bernhard. Sparse greedy matrix approximation for machine learning. 2000.

Steinwart, Ingo and Christmann, Andreas. *Support vector machines*. Springer Science & Business Media, 2008.

Tandon, Rashish, Si, Si, Ravikumar, Pradeep, and Dhillon, Inderjit. Kernel ridge regression via partitioning. *arXiv preprint arXiv:1608.01976*, 2016.

Tarres, Pierre and Yao, Yuan. Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence. *IEEE Transactions on Information Theory*, 60(9): 5716–5735, 2014.

Thomann, Philipp, Steinwart, Ingo, Blaschzyk, Ingrid, and Meister, Mona. Spatial decompositions for large scale SVMs. *arXiv preprint arXiv:1612.00374*, 2016.

Tropp, Joel A. User-friendly tools for random matrices: An introduction. Technical report, DTIC Document, 2012.

Williams, Christopher KI and Seeger, Matthias. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pp. 661–667. MIT press, 2000.

Yang, Tianbao, Li, Yu-Feng, Mahdavi, Mehrdad, Jin, Rong, and Zhou, Zhi-Hua. Nyström method vs random fourier features: A theoretical and empirical comparison. In *Advances in Neural Information Processing Systems*, pp. 476–484, 2012.

Yang, Yun, Pilanci, Mert, Wainwright, Martin J, et al. Randomized sketches for kernels: Fast and optimal nonparametric regression. *The Annals of Statistics*, 45(3):991–1023, 2017.

Yao, Yuan, Rosasco, Lorenzo, and Caponnetto, Andrea. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Ying, Yiming and Pontil, Massimiliano. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8(5):561–596, 2008.

Yurtsever, Alp, Udell, Madeleine, Tropp, Joel Aaron, and Cevher, Volkan. Sketchy decisions: Convex low-rank matrix optimization with optimal storage. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Zhang, Tong. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *International Conference on Machine learning*, pp. 116. ACM, 2004.

Zhang, Tong. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

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

Zhang, Yuchen, Wainwright, Martin J, and Duchi, John C. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pp. 1502–1510, 2012.

Zhang, Yuchen, Duchi, John C, and Wainwright, Martin J. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015.

Zinkevich, Martin, Weimer, Markus, Li, Lihong, and Smola, Alex J. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pp. 2595–2603, 2010.