Journal of Machine Learning Research 18 (2018) 1-42 Submitted 11/16; Revised 3/18; Published 6/18

Parallelizing Stochastic Gradient Descent for Least Squares Regression: Mini-batching, Averaging, and Model Misspecification

Prateek Jain, Praneeth Netrapalli {PRAJAIN,PRANEETH}@MICROSOFT.COM
Microsoft Research, Bangalore 560001, INDIA

Sham M. Kakade SHAM@CS.WASHINGTON.EDU
Paul G. Allen School of Computer Science and Department of Statistics,
University of Washington, Seattle WA 98195, USA

Rahul Kidambi RKIDAMBI@UW.EDU
Department of Electrical Engineering,
University of Washington, Seattle WA 98195, USA

Aaron Sidford SIDFORD@STANFORD.EDU
Department of Management Science and Engineering,
Stanford University, Palo Alto CA 94305, USA

Editor: Leon Bottou

Abstract

This work characterizes the benefits of averaging techniques widely used in conjunction with stochastic gradient descent (SGD). In particular, this work presents a sharp analysis of: (1) mini-batching, a method of averaging many samples of a stochastic gradient to both reduce the variance of a stochastic gradient estimate and for parallelizing SGD and (2) tail-averaging, a method involving averaging the final few iterates of SGD in order to decrease the variance in SGD's final iterate. This work presents sharp finite sample generalization error bounds for these schemes for the stochastic approximation problem of least squares regression.

Furthermore, this work establishes a precise problem-dependent extent to which mini-batching can be used to yield provable near-linear parallelization speedups over SGD with batch size one. This characterization is used to understand the relationship between learning rate versus batch size when considering the excess risk of the final iterate of an SGD procedure. Next, this mini-batching characterization is utilized in providing a highly parallelizable SGD method that achieves the minimax risk with nearly the same number of serial updates as batch gradient descent, improving significantly over existing SGD-style methods. Following this, a non-asymptotic excess risk bound for model averaging (which is a communication efficient parallelization scheme) is provided.

Finally, this work sheds light on fundamental differences in SGD's behavior when dealing with mis-specified models in the non-realizable least squares problem. This paper shows that maximal stepsizes ensuring minimax risk for the mis-specified case *must* depend on the noise properties.

The analysis tools used by this paper generalize the operator view of averaged SGD (Défossez and Bach, 2015) followed by developing a novel analysis in bounding these operators to characterize the generalization error. These techniques are of broader interest in analyzing various computational aspects of stochastic approximation.

Keywords: Stochastic Gradient Descent, Stochastic Approximation, Least Squares Regression, Parallelization, Mini Batch SGD, Iterate Averaging, Suffix Averaging, Batchsize Doubling, Model Averaging, Parameter Mixing, Mis-specified models, Heteroscedastic Noise, Agnostic Learning

1. Introduction and Problem Setup

With the ever increasing size of modern day datasets, practical algorithms for machine learning are increasingly constrained to spend less time and use less memory. This makes it particularly desirable to employ simple streaming algorithms that generalize well in a few passes over the dataset.

Stochastic gradient descent (SGD) is perhaps the simplest and most well studied algorithm that meets these constraints. The algorithm repeatedly samples an instance from the stream of data and updates the current parameter estimate using the gradient of the sampled instance. Despite its simplicity, SGD has been immensely successful and is the de-facto method for large scale learning problems. The merits of SGD for large scale learning and the associated computation versus statistics tradeoffs is discussed in detail by the seminal work of Bottou and Bousquet (2007).

While a powerful machine learning tool, unfortunately SGD in its simplest forms is inherently serial. Over the past years, as dataset sizes have grown there have been remarkable developments in processing capabilities with multi-core/distributed/GPU computing infrastructure available in abundance. The presence of this computing power has triggered the development of parallel/distributed machine learning algorithms (Mann et al. (2009); Zinkevich et al. (2011); Bradley et al. (2011); Niu et al. (2011); Li et al. (2014); Zhang and Xiao (2015)) that possess the capability to utilize multiple cores/machines. However, despite this exciting line of work, it is yet unclear how to best parallelize SGD and fully utilize these computing infrastructures.

This paper takes a step towards answering this question, by characterizing the behavior of constant stepsize SGD for the problem of strongly convex stochastic least square regression (LSR) under two averaging schemes widely believed to improve the performance of SGD. In particular, this work considers the natural parallelization technique of *mini-batching*, where multiple data-points are processed simultaneously and the current iterate is updated by the average gradient over these samples, and combine it with variance reducing technique of *tail-averaging*, where the average of many of the final iterates are returned as SGD's estimate of the solution.

In this work, parallelization arguments are structured through the lens of a *work-depth* tradeoff: *work* refers to the total computation required to reach a certain generalization error, and *depth* refers to the number of serial updates. Depth, defined in this manner, is a reasonable estimate of the runtime of the algorithm on a large multi-core architecture with shared memory, where there is no communication overhead, and has strong implications for parallelizability on other architectures.

1.1 Problem Setup and Notations

We use boldface small letters ($\mathbf{x}, \mathbf{w}$ etc.) for vectors, boldface capital letters ($\mathbf{A}, \mathbf{H}$ etc.) for matrices and normal script font letters ($\mathcal{M}, \mathcal{T}$ etc) for tensors. We use $\otimes$ to denote the outer product of two vectors or matrices. Loewner ordering between two PSD matrices is represented using $\succeq, \preceq$.
This paper considers the stochastic approximation problem of Least Squares Regression (LSR). Let $L : \mathbb{R}^d \to \mathbb{R}$ be the expected square loss over tuples $(\mathbf{x}, y)$ sampled from a distribution $\mathcal{D}$:

$$L(\mathbf{w}) = \frac{1}{2} \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[(y - \langle \mathbf{w}, \mathbf{x} \rangle)^2] \; \forall \; \mathbf{w} \in \mathbb{R}^d. \tag{1}$$

Let $\mathbf{w}^*$ be a minimizer of the problem (1). Now, let the Hessian of the problem (1) be denoted as:

$$\mathbf{H} \stackrel{\text{def}}{=} \nabla^2 L(\mathbf{w}) = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right].$$

Next, we define the fourth moment tensor $\mathcal{M}$ of the inputs $\mathbf{x}$ as:

$$\mathcal{M} \stackrel{\text{def}}{=} \mathbb{E}\left[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}\right].$$

Let the noise $\epsilon_{\mathbf{x},y}$ in a sample $(\mathbf{x}, y) \sim \mathcal{D}$ with respect to the minimizer $\mathbf{w}^*$ of (1) be denoted as:

$$\epsilon_{\mathbf{x},y} \stackrel{\text{def}}{=} y - \langle \mathbf{w}^*, \mathbf{x} \rangle.$$

Finally, let the noise covariance matrix $\mathbf{\Sigma}$ be denoted as:

$$\mathbf{\Sigma} \stackrel{\text{def}}{=} \mathbb{E}\left[\epsilon_{\mathbf{x},y}^2 \mathbf{x}\mathbf{x}^\top\right].$$

The *homoscedastic* (or, additive noise/well specified) case of LSR refers to the case when $\epsilon_{\mathbf{x},y}$ is mutually independent from $\mathbf{x}$. This is the case, say, when $\epsilon_{\mathbf{x},y}$ sampled from a Gaussian, $N(0, \sigma^2)$ independent of $\mathbf{x}$. In this case, $\mathbf{\Sigma} = \sigma^2 \mathbf{H}$, where, $\sigma^2 = \mathbb{E}\left[\epsilon^2\right]$, where the subscript on $\epsilon_{\mathbf{x},y}$ is suppressed owing to the independence of ϵ on any sample $(\mathbf{x}, y) \sim \mathcal{D}$. On the other hand, the *heteroscedastic* (or, mis-specified) case refers to the setting when $\epsilon_{\mathbf{x},y}$ is correlated with the input $\mathbf{x}$. In this paper, all our results apply to the general mis-specified case of the LSR problem.

1.1.1 ASSUMPTIONS

We make the following assumptions about the problem.

($\mathcal{A}1$) **Finite fourth moment:** The fourth moment tensor $\mathcal{M} = \mathbb{E}\left[\mathbf{x}^{\otimes 4}\right]$ exists and is finite.

($\mathcal{A}2$) **Strong convexity:** The Hessian of $L(\cdot)$, $\mathbf{H} = \mathbb{E}\left[\mathbf{x}\mathbf{x}^\top\right]$ is positive definite i.e., $\mathbf{H} \succ 0$.

($\mathcal{A}1$) is a standard regularity assumption for the analysis of SGD and related algorithms. ($\mathcal{A}2$) is also a standard assumption and guarantees that the minimizer of (1), i.e., $\mathbf{w}^*$ is unique.

1.1.2 IMPORTANT QUANTITIES

In this section, we will introduce some important quantities required to present our results. Let $\mathbf{I}$ denote the $d \times d$ identity matrix. For any matrix $\mathbf{A}$, $\mathcal{M}\mathbf{A} \stackrel{\text{def}}{=} \mathbb{E}\left[\left(\mathbf{x}^\top \mathbf{A} \mathbf{x}\right) \mathbf{x}\mathbf{x}^\top\right]$. Let $\mathcal{H}_{\mathcal{L}} = \mathbf{H} \otimes \mathbf{I}$ and $\mathcal{H}_{\mathcal{R}} = \mathbf{I} \otimes \mathbf{H}$ represent the left and right multiplication operators of the matrix $\mathbf{H}$ so that for any matrix $\mathbf{A}$, we have $\mathcal{H}_{\mathcal{L}}\mathbf{A} = \mathbf{H}\mathbf{A}$ and $\mathcal{H}_{\mathcal{R}}\mathbf{A} = \mathbf{A}\mathbf{H}$.

- **Fourth moment bound:** Let R^2 be the smallest number such that $\mathcal{M}\mathbf{I} \preceq R^2\mathbf{H}$.
- **Smallest eigenvalue:** Let μ be the smallest eigenvalue of $\mathbf{H}$ i.e., $\mathbf{H} \succeq \mu\mathbf{I}$.

The fourth moment bound implies that $\mathbb{E}\left[\|\mathbf{x}\|^2\right] \leq R^2$. Further more, ($\mathcal{A}2$) implies that the smallest eigenvalue μ of $\mathbf{H}$ is strictly greater than zero ($\mu > 0$).

1.1.3 STOCHASTIC GRADIENT DESCENT: MINI-BATCHING AND ITERATE AVERAGING

In this paper, we work with a stochastic first order oracle. This oracle, when queried at $\mathbf{w}$ samples an instance $(\mathbf{x}, y) \sim \mathcal{D}$ and uses this to return an unbiased estimate of the gradient of $L(\mathbf{w})$:

$$\widehat{\nabla L}(\mathbf{w}) = -(y - \langle \mathbf{w}, \mathbf{x} \rangle) \cdot \mathbf{x}; \;\; \mathbb{E}\left[\widehat{\nabla L}(\mathbf{w})\right] = \nabla L(\mathbf{w}).$$

We consider the stochastic gradient descent (SGD) method (Robbins and Monro, 1951), which minimizes $L(\mathbf{w})$ by following the direction opposite to this noisy stochastic gradient estimate, i.e.:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \gamma \cdot \widehat{\nabla L_t}(\mathbf{w}_{t-1}), \text{ with, } \widehat{\nabla L_t}(\mathbf{w}_{t-1}) = -(y_t - \langle \mathbf{w}_{t-1}, \mathbf{x}_t \rangle) \cdot \mathbf{x}_t$$

with $\gamma > 0$ being a constant step size/learning rate; $\widehat{\nabla L_t}(\mathbf{w}_{t-1})$ is the stochastic gradient evaluated using the sample $(\mathbf{x}_t, y_t) \sim \mathcal{D}$ at $\mathbf{w}_{t-1}$. We consider two algorithmic primitives used in conjunction with SGD namely, mini-batching and tail-averaging (also referred to as iterate/suffix averaging).

Mini-batching involves querying the gradient oracle several times and using the average of the returned stochastic gradients to take a single step. That is,

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \gamma \cdot \left(\frac{1}{b} \sum_{i=1}^{b} \widehat{\nabla L_{t,i}}(\mathbf{w}_{t-1}) \right),$$

where, b is the batch size. Note that at iteration t, mini-batching involves repeatedly querying the stochastic gradient oracle at $\mathbf{w}_{t-1}$ for a total of b times. For every query $i = 1, ..., b$ at iteration t, the oracle samples an instance $\{\mathbf{x}_{ti}, y_{ti}\}$ and returns a stochastic gradient estimate $\widehat{\nabla L_{t,i}}(\mathbf{w}_{t-1})$. These estimates $\{\widehat{\nabla L_{t,i}}(\mathbf{w}_{t-1})\}_{i=1}^{b}$ are averaged and then used to perform a single step from $\mathbf{w}_{t-1}$ to $\mathbf{w}_t$. Mini-batching enables the possibility of parallelization owing to the use of cheap matrix-vector multiplication for computing stochastic gradient estimates. Furthermore, mini-batching allows for the possible reduction of variance owing to the effect of averaging several stochastic gradient estimates.

Tail-averaging (or suffix averaging) refers to returning the average of the final few iterates of a stochastic gradient method as a means to improve its variance properties (Ruppert, 1988; Polyak and Juditsky, 1992). In particular, assuming the stochastic gradient method is run for $n-$steps, tail-averaging involves returning

$$\bar{\mathbf{w}} = \frac{1}{n-s} \sum_{t=s+1}^{n} \mathbf{w}_t$$

as an estimate of $\mathbf{w}^*$. Note that s can be interpreted as being cn, with $c < 1$ being some constant.

Typical excess risk bounds (or, generalization error bounds) for the stochastic approximation problem involve the contribution of two error terms namely, (i) the bias, which refers to the dependence on the starting conditions $\mathbf{w}_0$/initial excess risk $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and, (ii) the variance, which refers to the dependence on the noise introduced by the use of a stochastic first order oracle.

1.1.4 Optimal Error Rates for the Stochastic Approximation problem

Under standard regularity conditions often employed in the statistics literature, the minimax optimal rate on the excess risk is achieved by the standard Empirical Risk Minimizer (or, Maximum Likelihood Estimator) (Lehmann and Casella, 1998; van der Vaart, 2000). Given n i.i.d. samples $\mathcal{S}_n = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ drawn from $\mathcal{D}$, define the empirical risk minimization problem as obtaining

$$\mathbf{w}_n^* = \arg\min_{\mathbf{w}} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2.$$

Let us define the noise variance $\widehat{\sigma^2_{\text{MLE}}}$ to represent

$$\widehat{\sigma^2_{\text{MLE}}} = \mathbb{E}\left[\|\widehat{\nabla L}(\mathbf{w}^*)\|^2_{\mathbf{H}^{-1}} \right] = \text{Tr}[\mathbf{H}^{-1}\mathbf{\Sigma}].$$

The asymptotic minimax rate of the Empirical Risk Minimizer $\mathbf{w}_n^*$ on *every problem instance* is $\widehat{\sigma^2_{\text{MLE}}}/n$ (Lehmann and Casella, 1998; van der Vaart, 2000), i.e.,

$$\lim_{n\to\infty} \frac{\mathbb{E}_{\mathcal{S}_n}[L(\mathbf{w}_n^*)] - L(\mathbf{w}^*)}{\widehat{\sigma^2_{\text{MLE}}}/n} = 1.$$

For the well-specified case (i.e., the additive noise case, where, $\mathbf{\Sigma} = \sigma^2\mathbf{H}$), we have $\widehat{\sigma^2_{\text{MLE}}} = d\sigma^2$. Seminal works of Ruppert (1988); Polyak and Juditsky (1992) prove that tail-averaged SGD, with averaging from start, achieves the minimax rate for the *well-specified* case in the limit of $n \to \infty$.

Goal: In this paper, we seek to provide a non-asymptotic understanding of (a) mini-batching and issues of learning rate versus batch-size, (b) tail-averaging, (c) the effect of the model mis-specification, (d) a batch size doubling scheme for parallelizing statistical estimation, (e) a communication efficient parallelization scheme namely, parameter-mixing/model averaging and (f) the behavior of learning rate versus batch size on the final iterate of the mini-batch SGD procedure, on the behavior of excess risk of SGD (in terms of both the bias and the variance terms) for the streaming LSR problem, with the goal of achieving the minimax rate on every problem instance.

1.2 This Paper's Contributions

The main contributions of this paper are as follows:

- This work shows that mini-batching yields near-linear parallelization speedups over the standard serial SGD (i.e. with batch size 1), as long as the mini-batch size is smaller than a problem dependent quantity (which we denote by b_{thresh}). When batch-sizes increase beyond b_{thresh}, mini-batching is inefficient (owing to the lack of serial updates), thus obtaining only sub-linear speedups over mini-batching with a batch size b_{thresh}. A by-product of this analysis sheds light on how the step sizes naturally interpolate from ones used by standard serial SGD (with batch size 1) to ones used by batch gradient descent.

- While the final iterate of SGD decays the bias at a geometric rate but does not obtain minimax rates on the variance, the averaged iterate (Polyak and Juditsky, 1992; Défossez and Bach, 2015) decays the bias at a sublinear rate while achieving minimax rates on the variance. This work rigorously shows that tail-averaging obtains the best of both worlds: decaying the bias at a geometric rate and obtaining near-minimax rates (up to constants) on the variance. This result corroborates with empirical findings (Merity et al., 2017) that indicate the benefits of tail-averaging in general contexts such as training Long-Short term memory models (LSTMs).

- Next, this paper precisely characterizes the tradeoffs of learning rate versus batch size and its effect on the excess risk of the final iterate of an SGD procedure, which provides theoretical evidence to empirical observations (Goyal et al., 2017; Smith et al., 2017) described in the context of deep learning and non-convex optimization.

- Combining the above results, this paper provides a mini-batching and tail-averaging version of SGD that is highly parallelizable: the number of serial steps (which is a proxy for the un-parallelizable time) of this algorithm nearly matches that of *offline gradient descent* and is lower than the serial time of all existing streaming LSR algorithms. See Table 1 for comparison. We note that these results are obtained by providing a tight finite-sample analysis of the effects of mini-batching and tail-averaging with large constant learning rate schemes.

- We provide a non-asymptotic analysis of parameter mixing/model averaging schemes for the streaming LSR problem. Model averaging schemes are an attractive proposition for distributed learning owing to their communication efficient nature, and they are particularly effective in the regime when the estimation error (i.e. variance) is the dominating term in the excess risk. Here, we characterize the excess risk (in terms of both the bias and variance) of the model averaging procedure which sheds light on situations when it is an effective parallelization scheme (in that when this scheme yields linear parallelization speedups).

- All the results in this paper are established for the *general mis-specified* case of the streaming LSR problem. This establishes a fundamental difference in the behavior of SGD when dealing with mis-specified models in contrast to existing analyses that deal with the well-specified case. In particular, this analysis reveals a surprising insight that the maximal stepsizes (that ensure minimax optimal rates) are a function of the noise properties of the mis-specified problem instance. The main takeaway of this analysis is that the maximal step sizes (that permit achieving minimax rates) for the mis-specified case can be *much lower* than ones employed in the well-specified case: indeed, a problem instance that yields such a separation between the maximal learning rates for the well specified and the mis-specified case is presented.

The tool employed in obtaining these results generalizes the operator view of averaged SGD with batch size 1 (Défossez and Bach, 2015) and a clear exposition of the bias-variance decomposition from Jain et al. (2017a) to obtain a sharp bound on the excess risk for mini-batch, tail-averaged constant step-size SGD. Note that the work of Défossez and Bach (2015) does not establish minimax rates while working with large constant step sizes; this shortcoming is remedied by this paper through a novel sharp analysis that rigorously establishes minimax optimal rates while working with large constant step sizes. Furthermore, note that while straightforward operator norm bounds of the matrix operators suffice to show convergence of the SGD method, they turn out to be pretty loose bounds (particularly for bounding the variance). To tighten these bounds, this paper presents a fine grained analysis that bounds the trace of the SGD operators when applied to the relevant matrices. The bounds of this paper and its advantages compared to existing algorithms is indicated in table 1.

While this paper's results focus on strongly convex streaming least square regression, we believe that our techniques and results extend more broadly. This paper aims to serve as the basis for future work on analyzing SGD and parallelization of large scale algorithms for machine learning.

Paper organization: Section 2 presents the related work. Section 3 presents the main results of this work. Section 4 outlines the proof techniques. Section 5 presents experimental simulations to demonstrate the practical utility of the established mini-batching limits and tail-averaging. The proofs of all the claims and theorems are provided in the appendix.

2. Related Work

Stochastic approximation has been the focus of much efforts starting with the work of Robbins and Monro (1951), and has been analyzed in subsequent works including Nemirovsky and Yudin (1983); Kushner and Yin (1987, 2003). These questions and the related issues of computation versus statistics tradeoffs have received renewed attention owing to their relevance in the context of modern large scale machine learning, as highlighted by the work of Bottou and Bousquet (2007).

Geometric Rates on initial error: For *offline optimization* with strongly convex objectives, gradient descent (Cauchy, 1847) and fast gradient methods (Polyak, 1964; Nesterov, 1983) indicate linear

Algorithm	Final error	Runtime/Work	Depth	Streaming	Mis-specified
Gradient Descent (Cauchy, 1847)	$\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$	$\kappa n d \log \frac{n\cdot\Delta_0}{\sigma^2 d}$	$\kappa \log \frac{n\cdot\Delta_0}{\sigma^2 d}$	×	✓
SDCA (Shalev-Shwartz and Zhang, 2012)	$\mathcal{O}\left(\frac{\sigma^2 d}{n}\right)$	$(n+\frac{R^2}{\lambda_{\min}}d)d \cdot \log \frac{n\cdot\Delta_0}{\sigma^2 d}$	$(n+\frac{R^2}{\lambda_{\min}}d) \cdot \log \frac{n\cdot\Delta_0}{\sigma^2 d}$	×	✓
Averaged SGD (Défossez and Bach, 2015)[1]	$\mathcal{O}\left(\frac{1}{\lambda_{\min}^2 n^2\gamma^2}\cdot\Delta_0 + \frac{\sigma^2 d}{n}\right)$	nd	n	✓	×
Streaming SVRG with initial error oracle [2] (Frostig et al., 2015b)	$\mathcal{O}\left(\exp\left(-\frac{n\lambda_{\min}(\mathbf{H})}{R^2}\right)\cdot\Delta_0\right) + \frac{\sigma^2 d}{n}$	nd	$(\frac{R^2}{\lambda_{\min}(\mathbf{H})}) \cdot \log \frac{n\cdot\Delta_0}{\sigma^2 d}$	✓	✓
Algorithm 2 (this paper)	$\mathcal{O}\left(\left(\frac{R^2 t}{\|\mathbf{H}\|_2 n}\right)^{\frac{t}{\kappa\log(\kappa)}}\cdot\Delta_0 + \frac{\sigma^2 d}{n}\right)$	nd	$\frac{t}{t-\kappa\log(\kappa)}\cdot\kappa\log(\kappa)\cdot$ $\log\left(\frac{n\cdot\Delta_0}{\sigma^2 d}\cdot\frac{R^2 t}{\|\mathbf{H}\|_2}\right)$	✓	✓
Algorithm 2 with initial error oracle (this paper)	$\mathcal{O}\left(\exp\left(-\frac{n\lambda_{\min}(\mathbf{H})}{R^2\cdot\log(\kappa)}\right)\cdot\Delta_0 + \frac{\sigma^2 d}{n}\right)$	nd	$\kappa\log(\kappa)\log\frac{n\cdot\Delta_0}{\sigma^2 d}$	✓	✓

Table 1: Comparison of Algorithm 2 with existing algorithms including offline methods such as Gradient Descent, SDCA and streaming methods such as averaged SGD, streaming SVRG given n samples for LSR, with $\Delta_0 = L(\mathbf{w}_0) - L(\mathbf{w}^*)$. The error of offline methods are obtained by running these algorithms so that their final error is $\mathcal{O}(\sigma^2 d/n)$ (which is the minimax rate for the well-specified case). The table is written assuming the additive noise/well specified case; for algorithms which support the mis-specified case, these bounds can be appropriately modified. Refer to Section 1.1 for the definitions of all quantities. We do not consider accelerated variants in this table. Note that the accelerated variants have served to improve running times of the offline algorithms, with the sole exception of Jain et al. (2017b). For Algorithm 2, we require $t \geq 24\kappa\log(\kappa)$. Finally, note that streaming SVRG does not conform to the first order oracle model (Agarwal et al. (2012)).

convergence. However, a multiplicative coupling of number of samples n and condition number in the computational effort is a major drawback in the large scale context. These limitations are addressed through developments in offline stochastic methods (Roux et al., 2012; Shalev-Shwartz and Zhang, 2012; Johnson and Zhang, 2013; Defazio et al., 2014) and their accelerated variants (Shalev-Shwartz and Zhang, 2013a; Frostig et al., 2015a; Lin et al., 2015; Defazio, 2016; Allen-Zhu, 2016) which offer near linear running time in the number of samples and condition number with $\log(n)$ passes over the dataset stored in memory.

For *stochastic approximation* with strongly convex objectives, SGD offers linear rates on the bias without achieving minimax rates on the variance (Bach and Moulines, 2011; Needell et al., 2016; Bottou et al., 2016). In contrast, iterate averaged SGD (Ruppert, 1988; Polyak and Juditsky, 1992) offers a sub-linear $\mathcal{O}(1/n^2)$ rate on the bias (Défossez and Bach, 2015; Dieuleveut and Bach, 2015) while achieving minimax rates on the variance. Note that all these results consider the well-specified (additive noise) case when stating the generalization error bounds. We are unaware of any results that provide sharp non-asymptotic analysis of SGD and the related step size issues in the general mis-specified case. Streaming SVRG (Frostig et al., 2015b) offers a geometric rate on the bias and optimal statistical error rates; we will return to a discussion of Streaming SVRG below. In terms of methods faster than SGD, our own effort (Jain et al., 2017b) provides the first accelerated stochastic approximation method that improves over SGD on every problem instance.

Parallelization of Machine Learning algorithms: In *offline optimization*, Bradley et al. (2011) study parallel co-ordinate descent for sparse optimization. Parallelization via mini-batching has been studied in Cotter et al. (2011); Takác et al. (2013); Shalev-Shwartz and Zhang (2013b); Takác

1. Défossez and Bach (2015)'s bound holds with learning rate $\gamma \to 0$. This work supports these bounds with $\gamma = 1/R^2$.
2. Initial error oracle provides initial excess risk $\Delta_0 = L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and noise level σ^2.

et al. (2015). These results compare worst case upper bounds on the training error to argue parallelization speedups, thus providing weak upper bounds on mini-batching limits. Parameter mixing/Model averaging (Mann et al., 2009) guarantees linear parallelization speedups on the variance but do not improve the bias. Approaches that attempt to re-conciliate communication-computation tradeoffs (Li et al., 2014) indicate increased mini-batching hurts convergence, and this is likely an artifact of comparing weak upper bounds. Hogwild (Niu et al., 2011) indicates near-linear parallelization speedups in the harder asynchronous optimization setting, relying on specific input structures like hard sparsity; these bounds are obtained by comparing worst case upper bounds on training error. Refer to oracle models paragraph below for details on these worst case upper bounds.

In the *stochastic approximation* context, Dekel et al. (2012) study mini-batching in an oracle model that assumes bounded variance of stochastic gradients. These results compare worst case bounds on the generalization error to prescribe mini-batching limits, which renders these limits to be too loose (as mentioned in their paper). Our paper's mini-batching result offers guidelines on batch sizes for linear parallelization speedups by comparing generalization bounds that hold on a per problem basis as opposed to worst case bounds. Refer to the paragraph on oracle models for more details. Finally, parameter mixing in the stochastic approximation context (Rosenblatt and Nadler, 2014; Zhang et al., 2015) offers linear parallelization speedups on the variance error while not improving the bias (Rosenblatt and Nadler, 2014). Finally, Duchi et al. (2015) guarantees asymptotic optimality of asynchronous optimization with linear parallelization speedups on the variance.

Oracle models and optimality: In stochastic approximation, there are at least two lines of thought with regards to oracle models and notions of optimality. One line involves considering the case of bounded noise (Kushner and Yin, 2003; Kushner and Clark, 1978), or, bounded variance of the stochastic gradient, which in the least squares setting amounts to assuming bounds on

$$\widehat{\nabla L}(\mathbf{w}) - \nabla L(\mathbf{w}) = (\mathbf{x}\mathbf{x}^\top - \mathbf{H})(\mathbf{w} - \mathbf{w}^*) - \epsilon\mathbf{x}.$$

This implies additional assumptions are required on compactness of the parameter set (which are enforced via projection steps); such assumptions do not hold in practical implementation of stochastic gradient methods and in the setting considered by this paper. Thus, the mini-batching thresholds in Cotter et al. (2011); Niu et al. (2011); Dekel et al. (2012); Li et al. (2014) present bounds in the above worst-case oracle model by comparing weak upper bounds on the training/test error.

Another view of optimality (Anbar, 1971; Fabian, 1973) considers an objective where the goal is to match the rate of the statistically optimal estimator (referred to as the $M-$estimator) on every problem instance. Polyak and Juditsky (1992) consider this oracle model for the LSR problem and prove that the distribution of the averaged SGD estimator on every problem matches that of the $M-$estimator under certain regularity conditions (Lehmann and Casella, 1998). A recent line of work (Bach and Moulines, 2013; Frostig et al., 2015b) aims to provide non-asymptotic guarantees for SGD and its variants in this oracle model. This paper aims to understand mini-batching and other computational aspects of parallelizing stochastic approximation on every problem instance by working in this practically relevant oracle model. Refer to Jain et al. (2017b) for more details.

Comparing offline and streaming algorithms: Firstly, offline algorithms require performing multiple passes over a dataset stored in memory. Note that results and convergence rates established in the finite sum/offline optimization context do not translate to rates on the generalization error. Indeed, these results require going though concentration and a generalization error analysis for this translation to occur. Refer to Frostig et al. (2015b) for more details.

Comparison to streaming SVRG: Streaming SVRG does not function in the stochastic first order oracle model (Agarwal et al., 2012) satisfied by SGD as run in practice since it requires gradients at two points from a single sample (Frostig et al., 2015b). Furthermore, in contrast to this work, its depth bounds depend on a stronger fourth moment property due to lack of mini-batching.

3. Main Results

We begin by writing out the behavior of the learning rate as a function of batch size.

Maximal Learning Rates: We write out a characterization of the largest learning rate $\gamma^{div}_{b,\max}$ that permits the convergence of the mini-batch Stochastic Gradient Descent update. The following generalized eigenvector problem allows for the computation of $\gamma^{div}_{b,\max}$:

$$\frac{2}{\gamma^{div}_{b,\max}} = \sup_{\mathbf{W} \in \mathcal{S}(d)} \frac{\langle \mathbf{W}, \mathcal{M}\mathbf{W} \rangle + (b-1) \cdot \operatorname{Tr} \mathbf{WHWH}}{b \cdot \operatorname{Tr} \mathbf{WHW}}. \tag{2}$$

This characterization generalizes the divergent stepsize characterization of Défossez and Bach (2015) for batch sizes > 1. The derivation of the above characterization can be found in appendix A.5.1. We note that this characterization sheds light on how the divergent learning rates interpolate from batch size 1 (which is $\leq 2/\operatorname{Tr} \mathbf{H}$) to the batch gradient descent learning rate (setting b to ∞), which turns out to be $2/\lambda_{\max}(\mathbf{H})$. A property of $\gamma^{div}_{b,\max}$ worth noting is that it does not depend on properties of the noise ($\mathbf{\Sigma}$), and depends only on the second and fourth moment properties of the covariate $\mathbf{x}$.

We note that in this paper, our interest does not lie in the non-divergent stepsizes $0 \leq \gamma \leq \gamma^{div}_{b,\max}$, but in the set of (maximal) stepsizes $0 \leq \gamma \leq \gamma_{b,\max}$ ($< \gamma^{div}_{b,\max}$) that are sufficient to guarantee minimax error rates of $\mathcal{O}(\widehat{\sigma^2_{\text{MLE}}}/n)$. For the LSR problem, these maximal learning rates $\gamma_{b,\max}$ are:

$$\gamma_{b,\max} \stackrel{\text{def}}{=} \frac{2b}{R^2 \cdot \rho_{\text{m}} + (b-1)\|\mathbf{H}\|_2}, \text{ where, } \rho_{\text{m}} \stackrel{\text{def}}{=} \frac{d\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{\operatorname{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)}. \tag{3}$$

Note that $\rho_{\text{m}} \geq 1$ captures a notion of "degree" of model mismatch, and how it impacts the learning rate $\gamma_{b,\max}$; for the additive noise/well specified/homoscedastic case, $\rho_{\text{m}} = 1$. Thus, for problems where R^2 and $\|\mathbf{H}\|_2$ is held the same, the well-specified variant of the LSR problem admits a strictly larger learning rate (that achieves minimax rates on the variance) compared to the mis-specified case. Furthermore, in stark contrast to the well-specified case, $\gamma_{b,\max}$ in the mis-specified case depends not just on the second and fourth moment properties of the input, but also on the noise covariance $\mathbf{\Sigma}$. We show that our characterization of $\gamma_{b,\max}$ in the mis-specified case is tight in that there exist problem instances where $\gamma_{b,\max}$ (equation 3) is off the maximal learning rate in the well-specified case (obtained by setting $\rho_{\text{m}} = 1$ in equation 3) by a factor of the dimension d and $\gamma_{b,\max}$ is still the largest step size yielding minimax rates. We also note that there could exist mis-specified problem instances where a step size γ exceeding $\gamma_{b,\max}$ achieves minimax rates. Characterizing the maximal learning rate that achieves minimax rates on *every mis-specified* problem instance is an interesting open question. We return to the characterization of $\gamma_{b,\max}$ in section 3.1.

Note that this paper characterizes the performance of Algorithms 1 and 2 when run with a step size $\gamma \leq \frac{\gamma_{b,\max}}{2}$. The proofs turn out to be significantly complicated for $\gamma \in \left(\frac{\gamma_{b,\max}}{2}, \gamma_{b,\max}\right)$ and can be found in the initial version of this paper Jain et al. (2016b) and these were obtained through generalizing the operator view of analyzing SGD methods introduced by Défossez and Bach (2015). Note that for the well-specified case, this paper's results hold for the same learning rate regimes as

Algorithm 1 Minibatch-TailAveraging-SGD

Input: Initial point $\mathbf{w}_0$, stepsize γ, minibatch size b, initial iterations s, total samples n.

1: **for** $t = 1, 2, .., \lfloor \frac{n}{b} \rfloor$ **do**

2: Sample "b" tuples $\{(x_{ti}, y_{ti})\}_{i=1}^{b} \sim \mathcal{D}^b$

3: $\mathbf{w}_t \leftarrow \mathbf{w}_{t-1} - \frac{\gamma}{b} \sum_{i=1}^{b} \widehat{\nabla L_{ti}}(\mathbf{w}_{t-1})$

Output: $\bar{\mathbf{w}} = \frac{1}{\lfloor \frac{n}{b} \rfloor - s} \sum_{i>s} \mathbf{w}_i$

Bach and Moulines (2013); Frostig et al. (2015b), that are known to admit statistical optimality. We also note that in the additive noise case, we are unaware of a separation between $\gamma_{b,\max}$ and $\gamma_{b,\max}^{div}$; but as we will see, this is not of much consequence given that there exists a strict separation in the learning rate $\gamma_{b,\max}$ between the well-specified and mis-specified problem instances.

Finally, we note that the stochastic process viewpoint allows us to work with learning rates that are significantly larger compared to standard analyses that use function value contraction e.g., Bottou et al. (2016, Theorem 4.6). To the best of our knowledge, all existing works establishing mini-batching thresholds in the stochastic optimization setting e.g., Dekel et al. (2012) work in the worst case (bounded noise) oracle model, with small step sizes, and draw conclusions on mini-batch thresholds and effects by comparing weak upper bounds on the excess risk.

Mini-Batched Tail-Averaged SGD for the mis-specified case: We present our main result, which is the error bound for mini-batch tail-averaged SGD for the general mis-specified LSR problem.

Theorem 1 *Consider the general mis-specified case of the LSR problem 1. Running Algorithm 1 with a batch size $b \geq 1$, step size $\gamma \leq \gamma_{b,max}/2$, number of unaveraged iterations s, total number of samples n, we obtain an iterate $\overline{\mathbf{w}}$ satisfying the following excess risk bound:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2 \mu^2} \cdot \frac{(1-\gamma\mu)^s}{\left(\frac{n}{b} - s\right)^2} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{b \cdot (\frac{n}{b} - s)}. \tag{4}$$

In particular, with $\gamma = \gamma_{b,max}/2$, we have the following excess risk bound:

$$L(\overline{\mathbf{w}}) - L(\mathbf{w}^*) \leq \underbrace{\frac{2\kappa_b^2}{\left(\frac{n}{b} - s\right)^2} \exp\left(-\frac{s}{\kappa_b}\right) \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right)}_{\mathfrak{T}_1} + \underbrace{4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{b(\frac{n}{b} - s)}}_{\mathfrak{T}_2},$$

with $\kappa_b = \frac{R^2 \cdot \rho_m + (b-1)\|\mathbf{H}\|_2}{b\lambda_{min}(\mathbf{H})}$.

Note that the above theorem indicates that the excess risk is composed of two terms, namely the bias ($\mathfrak{T}_1$), which represents the dependence on the initial conditions $\mathbf{w}_0$ and the variance ($\mathfrak{T}_2$), which depends on the statistical noise ($\widehat{\sigma^2_{\text{MLE}}}$); the bias decays geometrically during the "s" unaveraged iterations while the variance is minimax optimal (up to constants) provided $s = \mathcal{O}(n)$. We will understand this geometric decay on the bias more precisely.

Effect of tail-averaging SGD's iterates: To understand tail-averaging, we specialize theorem 1 with a batch size 1 to the well-specified case, i.e., where, $\mathbf{\Sigma} = \sigma^2 \mathbf{H}$, $\widehat{\sigma^2_{\text{MLE}}} = d\sigma^2$ and $\rho_{\text{m}} = 1$.

Corollary 2 *Consider the well-specified (additive noise) case of the streaming LSR problem ($\boldsymbol{\Sigma} = \sigma^2\mathbf{H}$), with a batch size $b = 1$. With a learning rate $\gamma = \frac{\gamma_{1,max}}{2} = \frac{1}{R^2}$, unaveraged iterations s and total samples n, we have the following excess risk bound:*

$$L(\overline{\mathbf{w}}) - L(\mathbf{w}^*) \leq \underbrace{\frac{2\kappa_1^2}{(n-s)^2}\exp\left(-\frac{s}{\kappa_1}\right)\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\}}_{\mathfrak{T}_1} + \underbrace{4\cdot\frac{d\sigma^2}{n-s}}_{\mathfrak{T}_2}, \text{where, } \kappa_1 = R^2/\mu.$$

Tail-averaging allows for a geometric decay of the initial error $\mathfrak{T}_1$, while tail-averaging over $s = c\cdot n$ (with $c < 1$), allows for the variance $\mathfrak{T}_2$ to be minimax optimal (up to constants). We note that the work of Merity et al. (2017), which studies empirical optimization for training non-convex sequence models (e.g. Long-Short term memory models (LSTMs)) also indicate the benefits of tail-averaging.

Note that this particular case (i.e. additive noise/well-specified case with batch size 1) with tail-averaging from start ($s = 0$) is precisely the setting considered in Défossez and Bach (2015), and their result (a) achieves a sub-linear $\mathcal{O}(1/n^2)$ rate on the bias and (b) their variance term is shown to be minimax optimal only with learning rates that approach zero (i.e. $\gamma \to 0$).

3.1 Effects Of Learning Rate, Batch Size and The Role of Mis-specified Models

We now consider the interplay of learning rate, batch size and how model mis-specification plays into the mix. Towards this, we split this section into three parts: (a) understanding learning rate versus mini-batch size in the well-specified case, (b) how model mis-specification leads to a significant difference in the behavior of SGD and (c) how model mis-specification manifests itself when considered in tradeoff between the learning rate versus batch-size.

Effects of mini-batching in the well-specified case: As mentioned previously, in the well-specified case, $\boldsymbol{\Sigma} = \sigma^2\mathbf{H}$ and $\rho_{\text{m}} = 1$. For this case, equation (3) can be specialized as:

$$\gamma_{b,\text{max}} = \frac{2b}{R^2 + (b-1)\|\mathbf{H}\|_2}. \tag{5}$$

Observe that the learning rate $\gamma_{b,\text{max}}$ grows linearly as a function of the batch size b until a batch size $b = b_{\text{thresh}} = 1 + \frac{R^2}{\|\mathbf{H}\|_2}$. In the regime of batch sizes $1 < b \leq b_{\text{thresh}}$, the resulting mini-batch SGD updates offer near-linear parallelization speedups over SGD with a batch size of 1. Furthermore, increasing batch sizes beyond b_{thresh} leads to sub-linear increase in the learning rate, and this implies that we lose the linear parallelization speedup offered by mini-batching with a batch-size $b \leq b_{\text{thresh}}$. Losing the linear parallelization is indicative of the following: consider the case when we double batch-size from $b > b_{\text{thresh}}$ to $2b$. Suppose the bias error $\mathfrak{T}_1$ is larger than the variance $\mathfrak{T}_2$, we require performing the same number of updates with a batch size $2b$ as we did with a batch size b to achieve a similar excess risk bound; this implies we are inefficient in terms of number of samples (or, number of gradient computations) used to achieve a given excess risk. When the estimation error ($\mathfrak{T}_2$) dominates the approximation error ($\mathfrak{T}_1$), we note that larger batch sizes b (with $b > b_{\text{thresh}}$) serves to improve the variance term, thus allowing linear parallelization speedups via mini-batching.

Note that with a batch size of $b = b_{\text{thresh}}$, the learning rate of $\mathcal{O}(1/\lambda_{\text{max}}(\mathbf{H}))$ employed by mini-batch SGD resembles ones used by batch gradient descent. This mini-batching characterization thus allows for understanding tradeoffs of learning rate versus batch size. This behavior is noted in practice (empirically, but with no underlying rigorous theory) for a variety of problems (going beyond linear regression/convex optimization), in the deep learning context (Goyal et al., 2017).

SGD's behaviour with mis-specified models: Next, this paper attempts to shed light on some fundamental differences in the behavior of SGD when dealing with the mis-specified case (as against the well-specified case, which is the focus of existing results (Polyak and Juditsky, 1992; Bach and Moulines, 2013; Dieuleveut and Bach, 2015; Défossez and Bach, 2015)) of the LSR problem. This paper's results in general mis-specified case with batch sizes $b > 1$ specialize to existing results additive noise/well-specified case with batch size 1 (Bach and Moulines, 2013; Dieuleveut and Bach, 2015). To understand these issues better, we consider $\gamma_{b,\max}$ in equation 3 with a batch size 1:

$$\gamma_{1,\max} = \frac{2}{R^2 \cdot \rho_{\mathrm{m}}}. \tag{6}$$

Recounting that $\rho_{\mathrm{m}} \geq 1$, observe that the mis-specified case admits a maximal learning rate (with a view of achieving minimax rates) that is at most as large as the additive noise/well-specified case, where $\rho_{\mathrm{m}} = 1$. Note that when $\mathrm{Tr}\left(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}}\right)^{-1}\mathbf{\Sigma}\big)$ is nearly the same (say, upto constants) as the spectral norm $\left\|\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}}\right)^{-1}\mathbf{\Sigma}\right\|_2$, then $\rho_{\mathrm{m}} = \mathcal{O}(d)$ and $\gamma_{1,\max} = \mathcal{O}(\frac{1}{R^2 d})$. This implies that there exist mis-specified models whose noise properties (captured through the noise covariance matrix $\mathbf{\Sigma}$) prevents SGD from working with large learning rates of $\mathcal{O}(1/R^2)$ used in the well-specified case.

This notion is formalized in the following lemma, which presents an instance working with the mis-specified case, wherein, SGD *cannot* employ large learning rates used by the well-specified variant of the problem, while *retaining minimax optimality*. This behavior is in stark contrast to algorithms such as streaming SVRG (Frostig et al. (2015b)), which work with the same large learning rates in the mis-specified case as in the well-specified case, while guaranteeing minimax optimal rates. The proof of lemma 3 can be found in the appendix A.5.6.

Lemma 3 *Consider a Streaming LSR example with Gaussian covariates (i.e.* $\mathbf{x} \sim \mathcal{N}(0, \mathbf{H})$*) with a diagonal second moment matrix* $\mathbf{H}$ *that is defined by:*

$$\mathbf{H}_{ii} = \begin{cases} 1 & \text{if } i = 1 \\ 1/d & \text{if } i > 1 \end{cases}.$$

Further, let the noise covariance matrix $\mathbf{\Sigma}$ *be diagonal as well, with the following entries:*

$$\mathbf{\Sigma}_{ii} = \begin{cases} 1 & \text{if } i = 1 \\ 1/[(d-1)d] & \text{if } i > 1 \end{cases}.$$

For this problem instance, $\gamma_{1,max} \leq \frac{4}{(d+2)(1+\frac{1}{d})}$ *is necessary for retaining minimax rates, while the well-specified variant of this problem permits a maximal learning rate* $\leq \frac{d}{(d+2)(1+\frac{1}{d})}$*, thus implying an* $\mathcal{O}(d)$ *separation in learning rates between the well-specified and mis-specified case.*

Learning rate versus mini-batch size issues in the mis-specified case: Noting that for the batch size 1, as mentioned in equation 6, the learning rate for the mis-specified case in the most optimistic situation (when $\rho_{\mathrm{m}} =$ constant) can be atmost as large as the learning rate for the well-specified case. Furthermore, we also know from the observations in the mis-specified case that the learning rate tends to grow linearly as a function of the batch size until it hits the limit of $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$. Combining these observations, we will revisit equation 3, which says:

$$\gamma_{b,\max} \stackrel{\text{def}}{=} \frac{2b}{R^2 \cdot \rho_{\mathrm{m}} + (b-1)\|\mathbf{H}\|_2}.$$

This implies that the mini-batching size threshold b_{thresh} can be expressed as:

$$b_{\text{thresh}} \stackrel{\text{def}}{=} 1 + \frac{R^2}{\|\mathbf{H}\|_2} \cdot \rho_{\text{m}}. \tag{7}$$

When $1 < b \leq b_{\text{thresh}}$, we achieve near linear parallelization speedups over running SGD with a batch size 1. Note that this characterization specializes to the batch size threshold b_{thresh} presented in the well-specified case (i.e. where $\rho_{\text{m}} = 1$). Furthermore, this batch size threshold (in the mis-specified case) could be much larger than the threshold in the well-specified case, which is expected since the learning rate for a batch size 1 in the mis-specified case can potentially be much smaller than ones used in the well specified case. Furthermore, with a batch size b_{thresh}, note that the learning rate is $\mathcal{O}(1/\lambda_{\max}(\mathbf{H}))$, resembling ones used with batch gradient descent.

Behavior of the final-iterate: We now present the excess risk bound offered by the final iterate of a stochastic gradient scheme. This result is of much practical relevance in the context of modern machine learning and deep learning, where final iterate is often used, and where the tradeoffs between learning rate and batch sizes are discussed in great detail (Smith et al., 2017). For this discussion, we consider the well-specified case to present our results owing to its ease in presentation. Our framework and results are generic for translating these observations to the mis-specified case.

Lemma 4 *Consider the well-specified case of the LSR problem. Running Algorithm 1 with a step size* $\gamma \leq \frac{\gamma_{b,max}}{2} = \frac{b}{R^2+(b-1)\|\mathbf{H}\|_2}$, *batch size* b, *total samples* n *and with no iterate averaging (i.e. with* $s = n-1$*) yields a result* $\mathbf{w}_{\lfloor n/b \rfloor}$ *that satisfies the following excess risk bound:*

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b (1-\gamma\mu)^{\lfloor n/b \rfloor} \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + \frac{\gamma}{b}\sigma^2 \operatorname{Tr}(\mathbf{H}), \tag{8}$$

where $\kappa_b \stackrel{\text{def}}{=} \frac{R^2+(b-1)\|\mathbf{H}\|_2}{b\mu}$. *In particular, with a step size* $\gamma = \frac{\gamma_{b,max}}{2} = \frac{b}{R^2+(b-1)\|\mathbf{H}\|_2}$, *we have:*

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b \cdot e^{-\frac{\lfloor n/b \rfloor}{\kappa_b}} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + \frac{\sigma^2 \operatorname{Tr}(\mathbf{H})}{R^2 + (b-1)\|\mathbf{H}\|_2}. \tag{9}$$

Remarks: Noting that $\operatorname{Tr}(\mathbf{H}) \leq R^2$, the variance of the final iterate with batch size 1 is $\leq \sigma^2$. Next, with a batch size $b = b_{\text{thresh}}$, the final iterate has a variance $\leq \sigma^2/2$; at cursory glance this may appear interesting, in that by mini-batching, we do not appear to gain much in terms of the variance. This is unsurprising given that in the regime of $b \leq b_{\text{thresh}}$, the $\gamma_{b,\max}$ grows linearly, thus nullifying the effect of averaging multiple stochastic gradients. Furthermore, this follows in accordance with the linear parallelization speedups offered by a batch size $1 < b \leq b_{\text{thresh}}$. Note however, once $b > b_{\text{thresh}}$, any subsequent increase in batch sizes allows the variance of the final iterate to behave as $\mathcal{O}(\sigma^2/b)$. Finally, we note that once $b > b_{\text{thresh}}$, doubling batch sizes b (in equation 9) possesses the same effect as halving the learning rate from γ to $\gamma/2$ (as seen from equation 8), providing theoretical rigor to issues explored in training practical deep models (Smith et al., 2017).

3.2 Parallelization via Doubling Batch Sizes and Model Averaging

We now elaborate on a highly parallelizable stochastic gradient method, which is epoch based and relies on doubling batch sizes across epochs to yield an algorithm that offers the same generalization error as that of offline (batch) gradient descent in nearly the *s*ame number of serial updates as

Algorithm 2 MinibatchDoublingPartialAveragingSGD

Input: Initial point $\mathbf{w}_0$, stepsize γ, initial minibatch size b, number of iterations in each epoch s, number of samples n.

/*Run logarithmic number of epochs where each epoch runs t iterations of minibatch SGD (with out averaging). Double minibatch size after each epoch.*/

for $\ell = 1, 2, \cdots, \log \frac{n}{bt} - 1$ **do**

  $b_\ell \leftarrow 2^{\ell-1} b$

  $\mathbf{w}_\ell \leftarrow$ Minibatch-TailAveraging-SGD$(\mathbf{w}_{\ell-1}, \gamma, b_\ell, t-1, t \cdot b_\ell)$

/*For the last epoch, run tail averaged minibatch SGD with initial point $\mathbf{w}_t$, stepsize γ, mini-batch size $2^{\log \frac{n}{bt} - 1} \cdot b = n/2t$, number of initial iterations $t/2$ and number of samples $n/2$.*/

$\overline{\mathbf{w}} \leftarrow$ Minibatch-TailAveraging-SGD$(\mathbf{w}_s, \gamma, n/2t, t/2, n/2)$

Output: $\overline{\mathbf{w}}$

batch gradient descent, while being a streaming algorithm that does not require storing the entire dataset in memory. Following this, we present a non-asymptotic bound for parameter mixing/model averaging, which is a communication efficient parallelization scheme that has favorable properties when the estimation error (i.e. variance) is the dominating term of the excess risk.

(Nearly) Matching the depth of Batch Gradient Descent: The result of theorem 1 establishes a scalar generalization error bound of Algorithm 1 for the general mis-specified case of LSR and showed that the depth (number of sequential updates in our algorithm) is decreased to n/b. This section builds upon this result to present a simple and intuitive doubling based streaming algorithm that works in epochs and processes a total of $n/2$ points. In each epoch, the minibatch size is increased by a factor of 2 while applying Algorithm 1 (with no tail-averaging) with twice as many samples as the previous epoch. After running over $n/2$ samples using this epoch based approach, we run Algorithm 1 (with tail-averaging) with the remaining $n/2$ points. Intuitively, each of the epoch decays the bias of the previous epoch linearly and halves the statistical error (owing to doubling of mini-batch sizes). The final tail-averaging phase ensures that the variance is small.

The next theorem formalizes this intuition and shows Algorithm 2 improves the depth exponentially from n/b_{thresh} to $\mathcal{O}\left(\kappa \log(d\kappa) \log(n\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\}/\widehat{\sigma^2_{\text{MLE}}})\right)$ in the presence of an error oracle that provides us with the initial excess risk $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and the noise level $\widehat{\sigma^2_{\text{MLE}}}$.

Theorem 5 *Consider the general mis-specified case of LSR. Suppose in Algorithm 2, we use initial batchsize of $b = b_{thresh}$, stepsize $\gamma = \frac{\gamma_{b,max}}{2}$ and number of iterations in each epoch being $t \geq 24\kappa \log(\kappa)$, we obtain the following excess risk bound on $\overline{\mathbf{w}}$:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \left(\frac{2bt}{n}\right)^{\frac{t}{12\kappa \log(\kappa)}} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 80\frac{\widehat{\sigma^2_{MLE}}}{n}.$$

Remarks: The final error again has two parts: the bias term that depends on the initial error $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and the variance term that depends on the statistical noise $\widehat{\sigma^2_{\text{MLE}}}$. Note that the variance error decays at a rate of $\mathcal{O}\left(\widehat{\sigma^2_{\text{MLE}}}/n\right)$ which is minimax optimal up to constant factors.

Algorithm 2 decays the bias at a superpolynomial rate by choosing t large enough. If Algorithm 2 has access to an initial error oracle that provides $L(\mathbf{w}_0) - L(\mathbf{w}^*)$ and $\widehat{\sigma^2_{\text{MLE}}}$, we can run

Algorithm 2 with a batch size b_{thresh} until the excess risk drops to the noise level $\widehat{\sigma^2_{\text{MLE}}}$ and subsequently begin doubling the batch size. Such an algorithm indeed gives geometric convergence with a generalization error bound as:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \exp\left(-(\frac{n\lambda_{\min}}{R^2 \cdot \log(\kappa)}) \cdot \frac{1}{\rho_{\text{m}}}\right)\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\} + 80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n},$$

with a depth of $\mathcal{O}\left(\kappa \log(d\kappa) \log \frac{n\{L(\mathbf{w}_0) - L(\mathbf{w}^*)\}}{\widehat{\sigma^2_{\text{MLE}}}}\right)$. The proof of this claim follows relatively straightforwardly from the proof of Theorem 5. We note that this depth nearly matches (up to $\log$ factors), the depth of standard offline gradient descent despite being a streaming algorithm. This algorithm (aside from tail-averaging in the final epoch) resembles empirically effective schemes proposed in the context of training deep models (Smith et al., 2017).

Parameter Mixing/Model-Averaging: We consider a communication efficient method for distributed optimization which involves running mini-batch tail-averaged SGD independently on P separate machines (each containing their own independent samples) and averaging the resulting solution estimates. This is a well studied scheme for distributed optimization (Mann et al., 2009; Zinkevich et al., 2011; Rosenblatt and Nadler, 2014; Zhang et al., 2015). As mentioned in Rosenblatt and Nadler (2014), these schemes do not appear to offer improvements in the bias error while offering near linear parallelization speedups on the variance. We provide here a non-asymptotic characterization of the behavior of model averaging for the general mis-specified LSR problem.

Theorem 6 *Consider running Algorithm (1), i.e., mini-batch tail-averaged SGD (for the mis-specified LSR problem (1)) independently in P machines, each of which contains N/P samples. Let algorithm (1) be run with a batch size b, learning rate $\gamma \leq \gamma_{b,max}/2$, tail-averaging begun after $s-$iterations, and let each of these machines output $\{\overline{\mathbf{w}}_i\}_{i=1}^P$. The excess risk of the model-averaged estimator $\overline{\mathbf{w}} = \frac{1}{P}\sum_{i=1}^P \overline{\mathbf{w}}_i$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{(1-\gamma\mu)^s}{\gamma^2\mu^2\left(\frac{n}{P\cdot b} - s\right)^2} \cdot \frac{2 + (P-1)(1-\gamma\mu)^s}{P} \cdot \Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big) + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{P \cdot b \cdot \left(\frac{n}{P\cdot b} - s\right)}.$$

In particular, with $\gamma = \gamma_{b,max}/2$, we have the following excess risk bound:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \exp\left(-\frac{s}{\kappa_b}\right) \cdot \frac{\kappa_b^2}{\left(\frac{n}{P\cdot b} - s\right)^2} \cdot \frac{2 + (P-1)\cdot\exp(-s/\kappa_b)}{P} \cdot \big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\big) + 4 \cdot \frac{\widehat{\sigma^2_{MLE}}}{P \cdot b \cdot \left(\frac{n}{P\cdot b} - s\right)}.$$

Remarks: We note that during the iterate-averaged phase (i.e. $t > s$), there is no reduction of the bias, whereas, during the (initial) unaveraged iterations, once $s > \kappa_b \log(P)$, we achieve linear speedups on the bias. We note that model averaging offers linear parallelization speedups on the variance error. Furthermore, when the bias reduces to the noise level, model averaging offers linear parallelization speedups on the overall excess risk. Note that if $s = c \cdot n/(P \cdot b)$, with $c < 1$, then the excess risk is minimax optimal. Finally, we note that the theorem can be generalized in a straightforward manner to the situation when each machine has different number of examples.

4. Proof Outline

We present here the framework for obtaining the results described in this paper; the framework has been introduced in the work of Défossez and Bach (2015). Towards this purpose, we begin by introducing some notations. We begin by defining the centered estimate $\boldsymbol{\eta}_t$ as:

$$\boldsymbol{\eta}_t \stackrel{\text{def}}{=} \mathbf{w}_t - \mathbf{w}^*.$$

Mini-batch SGD (with a batch size b) moves $\boldsymbol{\eta}_{t-1}$ to $\boldsymbol{\eta}_t$ using the following update:

$$\boldsymbol{\eta}_t = \left(\mathbf{I} - \frac{\gamma}{b}\cdot\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\boldsymbol{\eta}_{t-1} + \frac{\gamma}{b}\sum_{i=1}^{b}\epsilon_{ti}\mathbf{x}_{ti} = (\mathbf{I}-\gamma\widehat{\mathbf{H}}_{tb})\boldsymbol{\eta}_{t-1} + \gamma\cdot\boldsymbol{\xi}_{tb},$$

where, $\widehat{\mathbf{H}}_{tb} = \frac{1}{b}\sum_{i=1}^{b}\mathbf{x}_{ti}\otimes\mathbf{x}_{ti}$ and $\boldsymbol{\xi}_{tb} = \frac{1}{b}\sum_{i=1}^{b}\epsilon_{ti}\mathbf{x}_{ti}$. Next, the tail-averaged iterate $\bar{\mathbf{x}}_{s,n}$ is associated with its own centered estimate $\bar{\boldsymbol{\eta}}_{s,n} = \frac{1}{n-s}\sum_{i=s+1}^{n}\boldsymbol{\eta}_i$. The analysis proceeds by tracking the covariance of the centered estimates $\boldsymbol{\eta}_t$, i.e. by tracking $\mathbb{E}\left[\boldsymbol{\eta}_t\otimes\boldsymbol{\eta}_t\right]$.

Bias-Variance decomposition: The main results of this paper are derived by going through the bias-variance decomposition, which is well known in the context of Stochastic Approximation (Bach and Moulines, 2011, 2013; Frostig et al., 2015b). The bias-variance decomposition allows for us to bound the generalization error by analyzing two sub-problems, namely, (i) The *bias* sub-problem, which analyzes the noiseless/realizable (or the consistent linear system) problem, by setting the noise $\epsilon_{ti} = 0\ \forall\ t, i$, $\boldsymbol{\eta}_0^{\text{bias}} = \boldsymbol{\eta}_0$ and (ii) the *variance* sub-problem, which involves starting at the solution, i.e., $\boldsymbol{\eta}_0^{\text{variance}} = 0$ and allowing the noise ϵ_{ti} to drive the resulting process. The corresponding tail-averaged iterates are associated with their centered estimates $\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}$ and $\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}$ respectively. The bias-variance decomposition for the square loss establishes the following relation:

$$\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}\otimes\bar{\boldsymbol{\eta}}_{s,n}\right] \preceq 2\cdot\left(\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\right] + \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\right]\right). \tag{10}$$

Using the bias-variance decomposition, we obtain an estimate of the generalization error as

$$\begin{aligned}\mathbb{E}\left[L(\bar{\mathbf{x}}_{s,n})\right] - L(\mathbf{x}^*) &= \tfrac{1}{2}\cdot\left\langle\mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}\otimes\bar{\boldsymbol{\eta}}_{s,n}\right]\right\rangle \\ &\leq \mathrm{Tr}\left(\mathbf{H}\cdot\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{bias}}\right]\right) + \mathrm{Tr}\left(\mathbf{H}\cdot\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\otimes\bar{\boldsymbol{\eta}}_{s,n}^{\text{variance}}\right]\right).\end{aligned}$$

We now provide a few lemmas that help us bound the behavior of the bias and variance error.

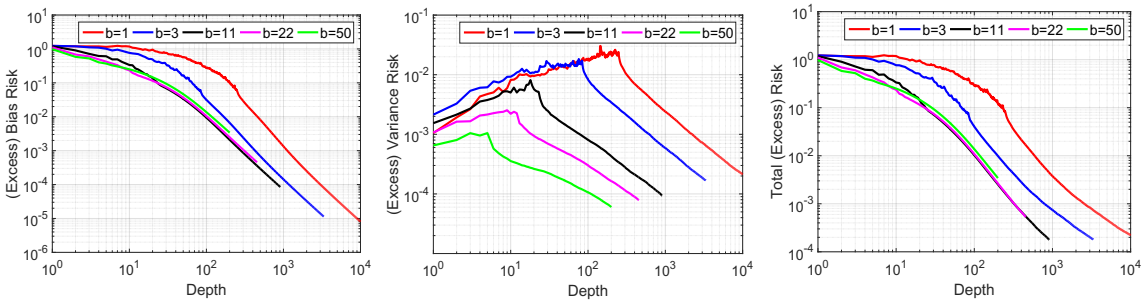
Lemma 7 *With a batch size b, step size $\gamma = \gamma_{b,max}/2$, the centered bias estimate $\boldsymbol{\eta}_t^{bias}$ exhibits the following per step contraction:*

$$\left\langle\mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{bias}\otimes\boldsymbol{\eta}_t^{bias}\right]\right\rangle \leq c_{\kappa_b}\left\langle\mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_{t-1}^{bias}\otimes\boldsymbol{\eta}_{t-1}^{bias}\right]\right\rangle,$$

where, $c_{\kappa_b} = 1 - 1/\kappa_b$, where $\kappa_b = \frac{R^2\cdot\rho_m + (b-1)\|\mathbf{H}\|_2}{b\mu}$.

Lemma (7) ensures that the bias decays at a geometric rate during the burn-in iterations when the iterates are not averaged; this rate holds only when the excess risk is larger than the noise level σ^2.

We now turn to bounding the variance error. It turns out that it suffices to understand the behavior of limiting centered variance $\mathbb{E}\left[\boldsymbol{\eta}_\infty^{\text{variance}}\otimes\boldsymbol{\eta}_\infty^{\text{variance}}\right]$.



(a) Bias Risk (b) Variance Risk (c) Total Risk

Figure 1: Effect of increased batch sizes on the Algorithm's generalization error. The variance decreases monotonically with increasing batch size. The bias indicates that the rate of decay increases till the optimal b_{thresh}. With $b = b_{\text{thresh}}$, mini-batch SGD obtains the same generalization error as batchsize 1 using smaller number of iterations (i.e. smaller depth) compared to larger batch sizes.

Lemma 8 *Consider the well-specified case of the streaming LSR problem. With a batch size b, step size $\gamma = \gamma_{b,max}/2$, the limiting centered variance $\boldsymbol{\eta}_\infty^{variance}$ has an expected covariance that is upper bounded in a psd sense as:*

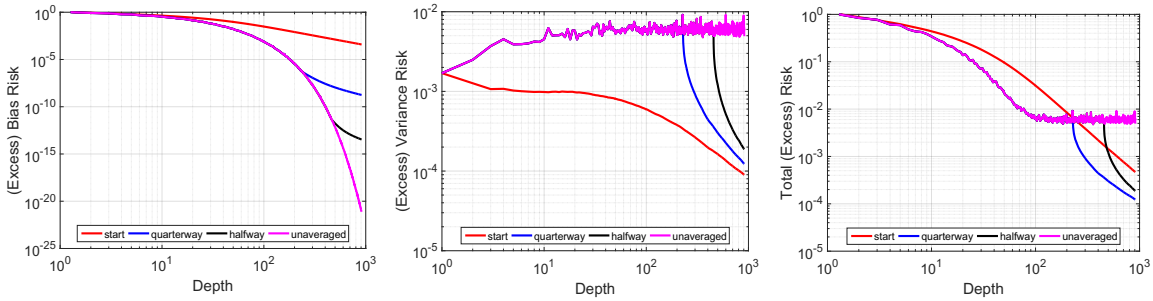
$$\mathbb{E}\left[\boldsymbol{\eta}_\infty^{variance} \otimes \boldsymbol{\eta}_\infty^{variance}\right] \preceq \frac{1}{R^2 + (b-1)\|\mathbf{H}\|_2} \cdot \sigma^2 \cdot \mathbf{I}.$$

Characterizing the behavior of the final iterate is crucial towards obtaining bounds on the behavior of the tail-averaged iterate. In particular, the final iterate having a excess variance risk $\mathcal{O}(\sigma^2)$ (as is the case with lemma (8)) appears crucial towards achieving minimax rates of the averaged iterate.

5. Experimental Simulations

We conduct experiments using a synthetic example to illustrate the implications of our theoretical results on mini-batching and tail-averaging. The data is sampled from a $50-$ dimensional Gaussian with eigenvalues decaying as $\{\frac{1}{k}\}_{k=1}^{50}$ (condition number $\kappa = 50$), and the variance σ^2 of the (additive noise) noise is 0.01. In this case, our estimated batch size according to Theorem 1 is $b_{\text{thresh}} = 11$. Our results are presented by averaging over 100 independent runs of the Algorithm, and each run employs 200κ samples. All plots are log-log with x-axis being the depth, and y-axis the excess risk. For our plots, we assume that each iteration takes constant time for all batch sizes; this is done to present evidence regarding the tightness of our mini-batching characterization limits that yield linear parallelization speedups over SGD with mini-batch size of 1.

We consider the effect of mini-batching (in figure 1) with batch sizes of 1, 3, $b_{\text{thresh}} = 11$, $2 \cdot b_{\text{thresh}} = 22$ and $d = 50$. Averaging begins after observing a fixed number of samples (set as 5κ). We see that the rate of bias decay (figure 1a) increases until reaching a mini-batch size of b_{thresh}, saturating thereafter; this implies we are inefficient in terms of sample size. As expected, the rate of decay of variance (figure 1b) is monotonic as a function of mini-batch size. Finally, the overall error (figure 1c) shows the tightness of our mini-batching characterization: with a batch size of b_{thresh}, we obtain a generalization error that is the *same* as using batch size of 1 with the number of (serial) iterations (i.e. depth) that is an order of magnitude smaller. Subsequently, we note that larger batch sizes worsen generalization error thus depicting the tightness of our characterization of b_{thresh}.



(a) Bias Risk (b) Variance Risk (c) Total Risk

Figure 2: [Zoom in to see detail] Effect of tail-averaging with mini-batch size of $b_{thresh} = 11$.

In the next experiment, we fix batch size $= b_{\text{thresh}}$ and consider the effect of when tail-averaging begins (figure 2). We consider averaging iterates from the start (as prescribed by Défossez and Bach (2015)), after a quarter/half of total number of iterations, and unaveraged SGD as well. We see that the bias (figure 2a) exhibits a geometric decay in the unaveraged phase while switching to an slower $\mathcal{O}(\frac{1}{t^2})$ rate with averaging. The variance (figure 2b) tends to increase and stabilize at $\mathcal{O}(\frac{\sigma^2}{b_{\text{thresh}}})$ in the absence of averaging, while switching to a $\mathcal{O}(\frac{1}{N})$ decay rate when averaging begins. The overall generalization error (figure 2c) shows the superiority of the scheme where averaging after a burn-in period allows the bias to decay towards the noise level at a geometric rate, following which tail-averaging allows us to decay the variance term, providing credence to our theoretical results that tail-averaged SGD allows us to obtain better generalization error as a function of sample size.

6. Concluding Remarks

This paper analyzes several algorithmic primitives often used in practice in conjunction with vanilla SGD for the stochastic approximation problem. In particular, this paper provides a sharp non-asymptotic treatment of (a) mini-batching, (b) tail-averaging, (c) effects of model mismatch, (d) behaviour of the final iterate, (e) highly parallel SGD method based on doubling batch sizes and (f) model-averaging/parameter mixing schemes for the strongly convex streaming LSR problem.

The effect of mini-batching and other algorithmic primitives mentioned above can be understood for a variety of models and/or algorithms. In particular, future directions could include understanding these issues for stochastic approximation with the Logistic Loss (Bach, 2014), streaming PCA (Jain et al., 2016a), and other algorithms such as streaming SVRG (Frostig et al., 2015b).

Acknowledgments

Sham Kakade acknowledges funding from Washington Research Foundation Fund for Innovation in Data-Intensive Discovery and National Science Foundation (NSF) through awards CCF-1703574 and CCF-1740551. Rahul Kidambi thanks James Saunderson for useful discussions on matrix operator theory.

Appendix A. Appendix

We begin with a note on the organization:

- Section A.1 introduces notations necessary for the rest of the appendix.
- Section A.2 derives the mini-batch SGD update and provides the bias-variance decomposition and reasons about its implication in bounding the generalization error.
- Section A.3 provides lemmas that are used to bound the bias error.
- Section A.4 provides lemmas that are used to bound the variance error.
- Section A.5 uses the results of the previous sections to obtain the main results of this paper.

A.1 Notations

We begin by introducing the centered iterate $\boldsymbol{\eta}_t$ i.e.:

$$\boldsymbol{\eta}_t \stackrel{\text{def}}{=} \mathbf{w}_t - \mathbf{w}^*.$$

In a manner similar to $\mathbf{w}_t$, the tail-averaged iterate $\overline{\mathbf{w}}_{t,N}$ is associated with its corresponding centered estimate $\bar{\boldsymbol{\eta}}_{t,N} \stackrel{\text{def}}{=} \overline{\mathbf{w}}_{t,N} - \mathbf{w}^* = \frac{1}{N}\sum_{s=t}^{t+N-1}(\mathbf{w}_s - \mathbf{w}^*) = \frac{1}{N}\sum_{s=t}^{t+N-1}\boldsymbol{\eta}_s$. Next, let $\boldsymbol{\Phi}_t$ denote the expected covariance of the centered estimate $\boldsymbol{\eta}_t$, i.e.

$$\boldsymbol{\Phi}_t \stackrel{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\eta}_t \otimes \boldsymbol{\eta}_t\right],$$

and in a similar way as the final iterate $\mathbf{w}_t$, the tail-averaged estimate $\overline{\mathbf{w}}_{t,N}$ is associated with its expected covariance, i.e. $\bar{\boldsymbol{\Phi}}_{t,N} \stackrel{\text{def}}{=} \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{t,N} \otimes \bar{\boldsymbol{\eta}}_{t,N}\right]$.

A.2 Mini-Batch Tail-Averaged SGD: Bias-Variance Decomposition

In section A.2.1, we derive the basic recursion governing the evolution of the iterates $\mathbf{w}_t$ and the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}$. In section A.2.2 we provide the bias-variance decomposition of the final iterate. In section A.2.3, we provide the bias-variance decomposition of the tail-averaged iterate.

A.2.1 THE BASIC RECURSION

At each iteration t of Algorithm 1, we are provided with b fresh samples $\{(\mathbf{x}_{ti}, y_{ti})\}_{i=1}^{b}$ drawn i.i.d. from the distribution $\mathcal{D}$. We start by recounting the mini-batch gradient descent update rule that allows us to move from iterate $\mathbf{w}_{t-1}$ to $\mathbf{w}_t$:

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \frac{\gamma}{b}\sum_{i=1}^{b}(\langle \mathbf{w}_{t-1}, \mathbf{x}_{ti}\rangle - y_{ti})\mathbf{x}_{ti},$$

where, $0 < \gamma < \gamma_{b,\max}$ is the constant step size that is set to a value less than the maximum allowed learning rate $\gamma_{b,\max}$. We also recount the definition of $\overline{\mathbf{w}}_{t,N}$ which is the iterate obtained by averaging for N iterations starting from the t^{th} iteration, i.e.,

$$\overline{\mathbf{w}}_{t,N} = \frac{1}{N}\sum_{s=t}^{t+N-1}\mathbf{w}_s.$$

Let us first denote the residual error term by $\epsilon_i = y_i - \langle \mathbf{w}^*, \mathbf{x}_i \rangle$. By the first order optimality conditions of $\mathbf{w}^*$, we observe that ϵ and $\mathbf{x}$ are orthogonal, i.e, $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\epsilon \cdot \mathbf{x}] = 0$. For any estimate $\mathbf{w}$, the excess risk/generalization error can be written as:

$$L(\mathbf{w}) - L(\mathbf{w}^*) = \frac{1}{2} \operatorname{Tr}\left(\mathbf{H} \cdot (\boldsymbol{\eta} \otimes \boldsymbol{\eta}) \right), \text{ with } \boldsymbol{\eta} = \mathbf{w} - \mathbf{w}^*. \tag{11}$$

We now write out the main recursion governing the mini-batch SGD updates in terms of $\boldsymbol{\eta}_\cdot$:

$$\begin{aligned}
\boldsymbol{\eta}_t &= \Big(\mathbf{I} - \frac{\gamma}{b} \sum_{i=1}^{b} \mathbf{x}_{ti} \otimes \mathbf{x}_{ti} \Big) \boldsymbol{\eta}_{t-1} + \frac{\gamma}{b} \sum_{i=1}^{b} \epsilon_{ti} \mathbf{x}_{ti} \\
&= \Big(\mathbf{I} - \frac{\gamma}{b} \sum_{i=1}^{b} \mathbf{x}_{ti} \otimes \mathbf{x}_{ti} \Big) \boldsymbol{\eta}_{t-1} + \frac{\gamma}{b} \sum_{i=1}^{b} \boldsymbol{\xi}_{ti} \\
&= \mathbf{P}_{tb} \boldsymbol{\eta}_{t-1} + \gamma \boldsymbol{\zeta}_{tb},
\end{aligned} \tag{12}$$

where, $\mathbf{P}_{tb} \overset{\text{def}}{=} \left(\mathbf{I} - \frac{\gamma}{b} \sum_{i=1}^{b} \mathbf{x}_{ti} \otimes \mathbf{x}_{ti} \right)$ and $\boldsymbol{\zeta}_{tb} \overset{\text{def}}{=} \frac{1}{b} \sum_{i=1}^{b} \boldsymbol{\xi}_{ti} = \frac{1}{b} \sum_{i=1}^{b} \epsilon_{ti} \mathbf{x}_{ti}$. Equation 12 automatically brings out the "operator" view of analyzing the (expected) covariance of the centered estimate $\boldsymbol{\Phi}_t = \mathbb{E}\left[\boldsymbol{\eta}_t \otimes \boldsymbol{\eta}_t \right]$ to provide an estimate of the generalization error. We now note the following about the covariance of $\boldsymbol{\zeta}_{tb}$:

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\zeta}_{tb} \otimes \boldsymbol{\zeta}_{t'b}] &= \frac{1}{b^2} \sum_{i,j} \mathbb{E}[\boldsymbol{\xi}_{ti} \otimes \boldsymbol{\xi}_{t'j}] \\
&= \Big[\frac{1}{b^2} \sum_{i=1}^{b} \mathbb{E}[\boldsymbol{\xi}_{ti} \otimes \boldsymbol{\xi}_{ti}] \Big] \mathbb{1}[t = t'] = \frac{1}{b} \boldsymbol{\Sigma} \ \mathbb{1}[t = t'],
\end{aligned} \tag{13}$$

where, $\mathbb{1}[.]$ is the indicator function, and equals 1 if the argument inside $[.]$ is true and 0 otherwise. We note that the expectation of the cross terms in equation 13 is zero owing to independence of the samples $\{\mathbf{x}_{ti}, y_{ti}\}_{i=1}^{b}$ as well as between $\{\mathbf{x}_{ti}, y_{ti}\}_{i=1}^{b}$, $\{\mathbf{x}_{t'i}, y_{t'i}\}_{i=1}^{b}$ $\forall\, t \neq t'$ and owing to the first order optimality conditions. Owing to the invariance of $\boldsymbol{\zeta}_{tb}$ on the iteration t, context permitting, we sometimes drop the iteration index t from $\boldsymbol{\zeta}_{tb}$ and simply refer to it as $\boldsymbol{\zeta}_b$.

Next we expand out the recurrence (12). Let $\mathbf{Q}_{j,t} = (\prod_{k=j}^{t} \mathbf{P}_{kb})^T$ with the convention that $\mathbf{Q}_{t',t} = \mathbf{I}\ \forall\, t' > t$. With this notation we have:

$$\begin{aligned}
\boldsymbol{\eta}_t &= \mathbf{P}_{tb} \boldsymbol{\eta}_{t-1} + \gamma \boldsymbol{\zeta}_{tb} \\
&= \mathbf{P}_{tb} \mathbf{P}_{t-1,b} \ldots \mathbf{P}_{1,b} \boldsymbol{\eta}_0 + \gamma \sum_{j=0}^{t-1} \{\mathbf{P}_{tb} \ldots \mathbf{P}_{t-j+1,b}\} \boldsymbol{\zeta}_{t-j,b} \\
&= \mathbf{Q}_{1,t} \boldsymbol{\eta}_0 + \gamma \sum_{j=0}^{t-1} \mathbf{Q}_{t-j+1,t} \boldsymbol{\zeta}_{t-j,b} \\
&= \mathbf{Q}_{1,t} \boldsymbol{\eta}_0 + \gamma \sum_{j=1}^{t} \mathbf{Q}_{j+1,t} \boldsymbol{\zeta}_{j,b} \\
&= \boldsymbol{\eta}_t^{\text{bias}} + \boldsymbol{\eta}_t^{\text{variance}},
\end{aligned} \tag{14}$$

where, we note that

$$\boldsymbol{\eta}_t^{\text{bias}} \stackrel{\text{def}}{=} \mathbf{Q}_{1,t}\boldsymbol{\eta}_0 \tag{15}$$

relates to understanding the behavior of SGD on the noiseless problem (i.e. $\boldsymbol{\zeta}_{\cdot,\cdot} = 0$ a.s.) and aims to quantify the dependence on the initial conditions. Further,

$$\boldsymbol{\eta}_t^{\text{variance}} \stackrel{\text{def}}{=} \gamma \sum_{j=1}^{t} \mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \tag{16}$$

relates to the behavior of SGD when begun at the solution (i.e. $\boldsymbol{\eta}_0 = 0$) and allowing the noise $\boldsymbol{\zeta}_{\cdot,\cdot}$ to drive the process.

Furthermore, considering the tail-averaged iterate obtained by averaging the iterates of the SGD procedure for N iterations starting from a certain number of iterations "s", i.e., we examine the quantity $\bar{\boldsymbol{\eta}}_{s+1,N} = \overline{\mathbf{w}}_{s+1,N} - \mathbf{w}^*$, where $\overline{\mathbf{w}}_{s+1,N} = \frac{1}{N}\sum_{t=s+1}^{s+N} \mathbf{w}_t$. We write out the expression for $\bar{\boldsymbol{\eta}}_{s+1,N}$ starting out from equation 14:

$$\begin{aligned}
\bar{\boldsymbol{\eta}}_{s+1,N} &= \frac{1}{N}\sum_{t=s+1}^{s+N} \boldsymbol{\eta}_t \\
&= \frac{1}{N}\sum_{t=s+1}^{s+N} \left(\boldsymbol{\eta}_t^{\text{bias}} + \boldsymbol{\eta}_t^{\text{variance}}\right) && \text{(from equation 14)} \\
&= \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{bias}} + \bar{\boldsymbol{\eta}}_{s+1,N}^{\text{variance}}.
\end{aligned} \tag{17}$$

A.2.2 THE FINAL ITERATE: BIAS-VARIANCE DECOMPOSITION

The behavior of the final iterate is considered to be of great practical interest and we hope to shed light on the behavior of this final iterate and the tradeoffs between the learning rate and batch size. Since the generalization error of any iterate $\mathbf{w}_N$ obtained by running mini-batch SGD with a batch size b for a total of N iterations can be estimated by tracking $\mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right]$, where, $\boldsymbol{\eta}_N = \mathbf{w}_N - \mathbf{w}^*$, we provide a simple psd upper bound on the outer product of interest, i.e.:

$$\begin{aligned}
\mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right] &= \mathbb{E}\left[\left(\boldsymbol{\eta}_N^{\text{bias}} + \boldsymbol{\eta}_N^{\text{variance}}\right) \otimes \left(\boldsymbol{\eta}_N^{\text{bias}} + \boldsymbol{\eta}_N^{\text{variance}}\right)\right] \quad \text{(by substituting equation 14)} \\
&\preceq 2 \cdot \left(\mathbb{E}\left[\left(\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right)\right] + \mathbb{E}\left[\left(\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right)\right]\right).
\end{aligned}$$

Using this expression, we now write out the expression for the excess risk of the final iterate:

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}_N)\right] - L(\mathbf{w}^*) &= \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N \otimes \boldsymbol{\eta}_N\right]\rangle \\
&\leq \frac{1}{2}\langle \mathbf{H}, 2 \cdot \left(\mathbb{E}\left[\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right] + \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right]\right)\rangle \\
&\leq 2 \cdot \left(\frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{bias}} \otimes \boldsymbol{\eta}_N^{\text{bias}}\right]\rangle + \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\boldsymbol{\eta}_N^{\text{variance}} \otimes \boldsymbol{\eta}_N^{\text{variance}}\right]\rangle\right) \\
&= 2 \cdot \left(\left(\mathbb{E}\left[L(\mathbf{w}_N^{\text{bias}})\right] - L(\mathbf{w}^*)\right) + \left(\mathbb{E}\left[L(\mathbf{w}_N^{\text{variance}})\right] - L(\mathbf{w}^*)\right)\right).
\end{aligned} \tag{18}$$

A.2.3 THE TAIL-AVERAGED ITERATE: BIAS-VARIANCE DECOMPOSITION

Now, considering the fact that the excess risk/generalization error (equation 11) involves tracking $\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]$, we see that the quantity of interest can be bounded by considering the behavior of SGD on bias and variance sub-problem. In particular, writing out the outerproduct of equation 17, we see the following inequality holds through a straightforward application of Cauchy-Shwartz inequality:

$$\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right] \preceq 2 \cdot \left(\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}^{\text{bias}}_{s+1,N}\right] + \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}^{\text{variance}}_{s+1,N}\right]\right). \tag{19}$$

The above equation is referred to as the bias-variance decomposition and is well known from previous work on Stochastic Approximation (Bach and Moulines, 2013; Frostig et al., 2015b; Défossez and Bach, 2015). This implies that an upper bound on the generalization error (equation 11) is:

$$\begin{aligned} L(\overline{\mathbf{w}}_{s+1,N}) - L(\mathbf{w}^*) &= \frac{1}{2}\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]\rangle \\ &\leq \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}^{\text{bias}}_{s+1,N}\right]\rangle + \langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}^{\text{variance}}_{s+1,N}\right]\rangle. \end{aligned} \tag{20}$$

Here, we adopt the proof approach of Jain et al. (2017a). In particular, Jain et al. (2017a) provide a clean way to simplify the expression corresponding to the tail-averaged iterate. Let us consider $\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N} \otimes \bar{\boldsymbol{\eta}}_{s+1,N}\right]$ and simplify the resulting expression: in particular,

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}\bar{\boldsymbol{\eta}}_{s+1,N}^\top\right] &= \frac{1}{N^2}\sum_{l=s+1}^{s+N}\sum_{k=s+1}^{s+N}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right] \\
&= \frac{1}{N^2}\cdot\left(\sum_{l\geq k}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right] + \sum_{l<k}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right]\right) \\
&\preceq \frac{1}{N^2}\cdot\left(\sum_{l\geq k}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right] + \sum_{l\leq k}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_k\right]\right) \qquad (*) \\
&= \frac{1}{N^2}\cdot\left(\sum_{l\geq k}(\mathbf{I}-\gamma\mathbf{H})^{l-k}\mathbb{E}\left[\boldsymbol{\eta}_k \otimes \boldsymbol{\eta}_k\right] + \sum_{l\leq k}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}\right) \quad (**) \\
&= \frac{1}{N^2}\cdot\sum_{l\leq k}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l} + (\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&= \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=l}^{s+N}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l} + (\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&= \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=l}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l} + (\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&\quad - \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=s+N+1}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l} + (\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right) \\
&= \frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right](\gamma\mathbf{H})^{-1} + (\gamma\mathbf{H})^{-1}\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right)
\end{aligned}$$

$$-\frac{1}{N^2}\sum_{l=s+1}^{s+N}\sum_{k=s+N+1}^{\infty}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right) \qquad (***), \tag{21}$$

where, $(*)$ is a valid PSD upper bound as we add and subtract the diagonal terms $\{\mathbb{E}\left[\boldsymbol{\eta}_k\boldsymbol{\eta}_k^\top\right]\}_{k=s+1}^{s+N}$. $(**)$ follows because of the following (assume $l>k$; the other case follows similarly):

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_k\right]&=\mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1}+\gamma\boldsymbol{\zeta}_{lb}\right)\otimes\boldsymbol{\eta}_k\right]\\&=\mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1}+\gamma\boldsymbol{\zeta}_{lb}\right)\otimes\boldsymbol{\eta}_k|\mathcal{F}_{l-1}\right]\right]\\&=\mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{lb}\boldsymbol{\eta}_{l-1}+\gamma\boldsymbol{\zeta}_{lb}\right)|\mathcal{F}_{l-1}\right]\otimes\boldsymbol{\eta}_k\right]\\&=(\mathbf{I}-\gamma\mathbf{H})\mathbb{E}\left[\boldsymbol{\eta}_{l-1}\otimes\boldsymbol{\eta}_k\right],\end{aligned}$$

where, the final equation follows since $\mathbb{E}\left[\mathbf{P}_{lb}|\mathcal{F}_{l-1}\right]=\mathbb{E}\left[\mathbf{I}-\frac{\gamma}{b}\sum_{i=1}^{b}\mathbf{x}_{li}\otimes\mathbf{x}_{li}|\mathcal{F}_{l-1}\right]=\mathbf{I}-\gamma\mathbf{H}$ and $\mathbb{E}\left[\boldsymbol{\zeta}_{lb}|\mathcal{F}_{l-1}\right]=0$ from first order optimality conditions. Recursing over l yields the result. $(***)$ follows from summing a (convergent) geometric series.

This implies that the excess risk corresponding to the bias/variance term can be obtained from equation 21 by taking an inner product with $\mathbf{H}$, i.e.:

$$\begin{aligned}\left\langle\mathbf{H},\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}\otimes\bar{\boldsymbol{\eta}}_{s+1,N}\right]\right\rangle&\leq\frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\left(\left\langle\mathbf{H},\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\gamma\mathbf{H})^{-1}+(\gamma\mathbf{H})^{-1}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right\rangle\right)\\&\quad-\frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\sum_{k=s+N+1}^{\infty}\left(\left\langle\mathbf{H},\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}\right.\right.\\&\qquad\left.\left.+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right\rangle\right)\\&\leq\frac{1}{N^2}\cdot\sum_{l=s+1}^{s+N}\left(\left\langle\mathbf{H},\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\gamma\mathbf{H})^{-1}+(\gamma\mathbf{H})^{-1}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right\rangle\right)\\&=\frac{2}{\gamma N^2}\cdot\sum_{l=s+1}^{s+N}\mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right).\end{aligned} \tag{22}$$

The upper bound on the final line follows because each term within the summation in the second line is negative owing to the following argument. Consider say,

$$\begin{aligned}&\left\langle\mathbf{H},\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right](\mathbf{I}-\gamma\mathbf{H})^{k-l}+(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right\rangle\\&=2\,\mathrm{Tr}\left[\mathbf{H}(\mathbf{I}-\gamma\mathbf{H})^{k-l}\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]\right]\geq 0.\end{aligned}$$

Note that $\mathbf{H}$ and $(\mathbf{I}-\gamma\mathbf{H})$ commute and both are psd, implying that $\mathbf{H}(\mathbf{I}-\gamma\mathbf{H})^{k-l}$ is PSD. Finally, the trace of the product of two PSD matrices is positive with $\mathbf{H}(\mathbf{I}-\gamma\mathbf{H})^{k-l}$ being one of these PSD matrices and $\mathbb{E}\left[\boldsymbol{\eta}_l\otimes\boldsymbol{\eta}_l\right]$ being the other, thus yielding the claimed bound in equation 22.

This implies that the overall error (through equation 11) can be upperbounded as:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N})\right]-L(\mathbf{w}^*)=\frac{1}{2}\cdot\left\langle\mathbf{H},\mathbb{E}\left[\bar{\boldsymbol{\eta}}_{s+1,N}\otimes\bar{\boldsymbol{\eta}}_{s+1,N}\right]\right\rangle$$

$$\leq \frac{1}{\gamma N^2} \sum_{l=s+1}^{s+N} \mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l \otimes \boldsymbol{\eta}_l\right]\right)$$
$$\leq \frac{2}{\gamma N^2} \cdot \sum_{l=s+1}^{s+N} \left(\mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{bias}} \otimes \boldsymbol{\eta}_l^{\text{bias}}\right]\right) + \mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{variance}} \otimes \boldsymbol{\eta}_l^{\text{variance}}\right]\right)\right), \tag{23}$$

where the final line follows from equation 19. We will now bound each of these terms to precisely characterize the excess risk of mini-batch tail-averaged SGD. We refer to the bias error of the tail-averaged iterate as the following:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{\text{bias}})\right] - L(\mathbf{w}^*) \stackrel{\text{def}}{=} \frac{2}{\gamma N^2} \sum_{l=s+1}^{s+N} \mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{bias}} \otimes \boldsymbol{\eta}_l^{\text{bias}}\right]\right). \tag{24}$$

Similarly, we refer to the variance error of the tail-averaged iterate as the following:

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{\text{variance}})\right] - L(\mathbf{w}^*) \stackrel{\text{def}}{=} \frac{2}{\gamma N^2} \sum_{l=s+1}^{s+N} \mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{\eta}_l^{\text{variance}} \otimes \boldsymbol{\eta}_l^{\text{variance}}\right]\right). \tag{25}$$

A.3 Lemmas For Bounding The Bias Error

Lemma 9 *With $\gamma \leq \frac{\gamma_{b,max}}{2} = \frac{b}{R^2 \cdot \rho_m + (b-1)\|\mathbf{H}\|_2}$, the following bound holds:*

$$\left\| \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li}) \right] \right\|_2 \leq 1 - \gamma\mu.$$

Proof This lemma generalizes one appearing in Jain et al. (2017a) to the mini-batch size b case. Denote by $\mathbf{U}$ the matrix of interest and consider the following:

$$\mathbf{U} = \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{j=1}^{b} \mathbf{x}_{li} \otimes \mathbf{x}_{li}) \right]$$
$$= \mathbf{I} - \gamma\mathbf{H} - \gamma\mathbf{H} + \left(\frac{\gamma}{b}\right)^2 \cdot \left(b\mathbb{E}\left[\|\mathbf{x}\|^2 \mathbf{x}\mathbf{x}^\top\right] + b(b-1)\mathbf{H}^2 \right)$$
$$\preceq \mathbf{I} - 2\gamma\mathbf{H} + \frac{\gamma^2}{b} \cdot \left(R^2\mathbf{H} + (b-1)\|\mathbf{H}\|_2\right)\mathbf{H}$$
$$= \mathbf{I} - \gamma\mathbf{H},$$

from which a spectral norm bound implied by the lemma naturally follows. ■

Lemma 10 *For any learning rate $\gamma \leq \gamma_{b,max}/2$, the bias error of the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}^{bias}$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{bias})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2 N^2 \mu^2}(1-\gamma\mu)^{s+1} \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right).$$

Proof Before writing out the proof of the bound in the lemma, we require to bound the per step contraction properties of an SGD update in the case of the bias error (i.e. $\boldsymbol{\zeta}_\cdot = 0$):

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{\eta}_l\|^2\right] &= \mathbb{E}\left[\boldsymbol{\eta}_{l-1}^\top (\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})\boldsymbol{\eta}_{l-1}\right] \\
&= \mathbb{E}\left[\boldsymbol{\eta}_{l-1}^\top \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{li}\otimes\mathbf{x}_{li})\middle|\mathcal{F}_{l-1}\right]\boldsymbol{\eta}_{l-1}\right] \\
&\leq (1-\gamma\mu)\mathbb{E}\left[\|\boldsymbol{\eta}_{l-1}\|^2\right] \quad \text{(using lemma 9)}.
\end{aligned}$$

This implies that a recursive application of the above bound yields $\mathbb{E}\left[\|\boldsymbol{\eta}_l\|^2\right] \leq (1-\gamma\mu)^l \mathbb{E}\left[\|\boldsymbol{\eta}_0\|^2\right]$.

Next, we consider the bias error from equation 24:

$$\begin{aligned}
\mathbb{E}\left[L(\overline{\mathbf{w}}^{\text{bias}}_{s+1,N})\right] - L(\mathbf{w}^*) &= \frac{2}{\gamma N^2}\sum_{t=s+1}^{s+N}\mathbb{E}\left[\|\boldsymbol{\eta}_t\|^2\right] \\
&\leq \frac{2}{\gamma N^2}\sum_{t=s+1}^{\infty}\mathbb{E}\left[\|\boldsymbol{\eta}_t\|^2\right] \\
&\leq \frac{2}{\gamma N^2}\sum_{t=s+1}^{\infty}(1-\gamma\mu)^t\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma N^2}(\gamma\mu)^{-1}(1-\gamma\mu)^{s+1}\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma^2\mu N^2}(1-\gamma\mu)^{s+1}\|\boldsymbol{\eta}_0\|^2 \\
&= \frac{2}{\gamma^2\mu^2 N^2}(1-\gamma\mu)^{s+1}\cdot\left(\mu\cdot\|\boldsymbol{\eta}_0\|^2\right) \\
&\leq \frac{2}{\gamma^2\mu^2 N^2}(1-\gamma\mu)^{s+1}\cdot\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right),
\end{aligned}$$

where in the final line, we use the fact that $\mu\mathbf{I} \preceq \mathbf{H}$. This proves the claimed bound. ■

Lemma 11 *For any learning rate $\gamma \leq \gamma_{b,max}/2$, the bias error of the* **final** *iterate $\mathbf{w}_N^{bias}$ is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{bias})\right] - L(\mathbf{w}^*) \leq \frac{\kappa}{2}\cdot(1-\gamma\mu)^N\cdot\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right).$$

Proof Similar to the tail-averaged case, we require to bound the per step contraction properties of an SGD update in the case of the bias error (i.e. $\boldsymbol{\zeta}_\cdot = 0$):

$$\begin{aligned}
\mathbb{E}\left[\|\boldsymbol{\eta}_N\|^2\right] &= \mathbb{E}\left[\boldsymbol{\eta}_{N-1}^\top (\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})\boldsymbol{\eta}_{N-1}\right] \\
&= \mathbb{E}\left[\boldsymbol{\eta}_{N-1}^\top \mathbb{E}\left[(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{Ni}\otimes\mathbf{x}_{Ni})\middle|\mathcal{F}_{N-1}\right]\boldsymbol{\eta}_{N-1}\right]
\end{aligned}$$

$$\leq (1-\gamma\mu)\mathbb{E}\left[\|\boldsymbol{\eta}_{N-1}\|^2\right] \quad \text{(using lemma 9)}.$$

This implies that a recursive application of the above bound yields $\mathbb{E}\left[\|\boldsymbol{\eta}_N\|^2\right] \leq (1-\gamma\mu)^N\mathbb{E}\left[\|\boldsymbol{\eta}_0\|^2\right]$. Then,

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}_N^{\text{bias}})\right] - L(\mathbf{w}^*) &= \frac{1}{2}\operatorname{Tr}\left((\boldsymbol{\eta}_N^{\text{bias}})^\top \mathbf{H}\boldsymbol{\eta}_N^{\text{bias}}\right) \\
&\leq \frac{\lambda_{\text{max}}(\mathbf{H})}{2}\operatorname{Tr}\left(\|\boldsymbol{\eta}_N^{\text{bias}}\|^2\right) \\
&\leq \frac{\lambda_{\text{max}}(\mathbf{H})(1-\gamma\mu)^N}{2\lambda_{\text{min}}(\mathbf{H})}\operatorname{Tr}\left(\lambda_{\text{min}}(\mathbf{H})\|\boldsymbol{\eta}_0\|^2\right) \\
&\leq \frac{\lambda_{\text{max}}(\mathbf{H})(1-\gamma\mu)^N}{2\lambda_{\text{min}}(\mathbf{H})}\Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big) \qquad \text{(since, } \mathbf{w}_0 = \mathbf{w}_0^{\text{bias}}\text{)}. \\
&\leq \frac{\kappa}{2}\cdot(1-\gamma\mu)^N\Big(L(\mathbf{w}_0) - L(\mathbf{w}^*)\Big).
\end{aligned}$$

■

A.4 Lemmas For Bounding The Variance Error

Now, we seek to understand the behavior of the variance error of the tail-averaged iterate $\overline{\mathbf{w}}_{s+1,N}$. We begin by noting here that the variance error is analyzed by beginning the optimization at the solution, i.e. $\boldsymbol{\eta}_0^{\text{variance}} = 0$ and allowing the noise to drive the process. In particular, we write out the recursive updates that characterize the variance error:

$$\boldsymbol{\eta}_t^{\text{variance}} = \mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}, \text{ with } \boldsymbol{\eta}_0^{\text{variance}} = 0.$$

This implies that by defining $\boldsymbol{\Phi}_t^{\text{variance}} \stackrel{\text{def}}{=} \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}} \otimes \boldsymbol{\eta}_t^{\text{variance}}\right]$, we have:

$$\begin{aligned}
\boldsymbol{\Phi}_t^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}} \otimes \boldsymbol{\eta}_t^{\text{variance}}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\left(\mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}\right) \otimes \left(\mathbf{P}_{tb}\boldsymbol{\eta}_{t-1}^{\text{variance}} + \gamma\boldsymbol{\zeta}_{tb}\right)|\mathcal{F}_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbf{P}_{tb}\boldsymbol{\Phi}_{t-1}^{\text{variance}}\mathbf{P}_{tb}^\top\right] + \frac{\gamma^2}{b}\boldsymbol{\Sigma}. \qquad (26)
\end{aligned}$$

where, $\mathcal{F}_{t-1}$ is the filtration defined using the samples $\{\mathbf{x}_{ji}, y_{ji}\}_{j=1,i=1}^{j=t-1,i=b}$. Furthermore cross terms are zero since $\mathbb{E}\left[\boldsymbol{\zeta}_{tb}|\mathcal{F}_{t-1}\right] = 0$ owing to first order optimality conditions. Recounting that $\mathbf{P}_{tb} = \mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}$, we express equation 26 using a linear operator as follows:

$$\begin{aligned}
\mathbb{E}\left[\mathbf{P}_{tb}\boldsymbol{\Phi}_{t-1}^{\text{variance}}\mathbf{P}_{tb}^\top\right] &= \mathbb{E}\left[\left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\boldsymbol{\Phi}_{t-1}^{\text{variance}}\left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^b \mathbf{x}_{ti}\otimes\mathbf{x}_{ti}\right)\right] \\
&\stackrel{\text{def}}{=} (\mathcal{I} - \gamma\mathcal{T}_b)\boldsymbol{\Phi}_{t-1}^{\text{variance}},
\end{aligned}$$

with $\mathcal{T}_b$ representing the following linear operator:

$$\mathcal{T}_b = \mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}} - \frac{\gamma}{b}\mathcal{M} - \gamma\frac{b-1}{b}\mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}},$$

with $\mathcal{M} = \mathbb{E}[\mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x} \otimes \mathbf{x}]$, $\mathcal{H}_{\mathcal{L}} = \mathbf{H} \otimes \mathbf{I}$ and $\mathcal{H}_{\mathcal{R}} = \mathbf{I} \otimes \mathbf{H}$ representing the left and right multiplication linear operators corresponding to the matrix $\mathbf{H}$. Given this notation, we consider $\mathbf{\Phi}_t^{\text{variance}}$:

$$\begin{aligned}\mathbf{\Phi}_t^{\text{variance}} &= (\mathcal{I} - \gamma\mathcal{T}_b)\mathbf{\Phi}_{t-1}^{\text{variance}} + \frac{\gamma^2}{b}\mathbf{\Sigma} \\ &= \frac{\gamma^2}{b}\left(\sum_{k=0}^{t-1}(\mathcal{I} - \gamma\mathcal{T}_b)^k\right)\mathbf{\Sigma}. \end{aligned} \tag{27}$$

Before bounding the variance error, we will describe a lemma that shows that the expected covariance of the variance error $\mathbf{\Phi}_t^{\text{variance}}$ initialized at 0 grows monotonically to its steady state value (in a PSD sense).

Lemma 12 *The sequence of centered variance iterates $\boldsymbol{\eta}_t^{variance}$ have expected covariances that monotonically grow in a PSD sense, i.e.:*

$$0 = \mathbf{\Phi}_0^{variance} \preceq \mathbf{\Phi}_1^{variance} \preceq \mathbf{\Phi}_2^{variance} .... \preceq \mathbf{\Phi}_\infty^{variance}.$$

Proof This lemma generalizes the lemma appearing in Jain et al. (2017a,b). We begin by recounting the t^{th} variance iterate, i.e.:

$$\boldsymbol{\eta}_t^{\text{variance}} = \gamma\sum_{j=1}^{t}\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b}.$$

This implies in particular that

$$\begin{aligned}\mathbf{\Phi}_t^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}} \otimes \boldsymbol{\eta}_t^{\text{variance}}\right] \\ &= \gamma^2\sum_{j=1}^{t}\sum_{l=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{l,b}\mathbf{Q}_{l+1,b}^\top\right] \quad \text{(from equation 14)} \\ &= \gamma^2\sum_{j=1}^{t}\sum_{l=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\mathbb{E}\left[\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{l,b}|\mathcal{F}_{j-1}\right]\mathbf{Q}_{l+1,b}^\top\right] \\ &= \gamma^2\sum_{j=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\boldsymbol{\zeta}_{j,b} \otimes \boldsymbol{\zeta}_{j,b}\mathbf{Q}_{j+1,t}^\top\right] \\ &= \frac{\gamma^2}{b}\sum_{j=1}^{t}\mathbb{E}\left[\mathbf{Q}_{j+1,t}\mathbf{\Sigma}\mathbf{Q}_{j+1,t}^\top\right].\end{aligned}$$

where, the third line follows since $\mathbb{E}\left[\boldsymbol{\zeta}_{l,b} \otimes \boldsymbol{\zeta}_{j,b}\right] = 0$ for $j \neq l$, similar to arguments in equation 13. This immediately reveals that the sequence of covariances grows as a function of time, since,

$$\mathbf{\Phi}_{t+1}^{\text{variance}} - \mathbf{\Phi}_t^{\text{variance}} = \frac{\gamma^2}{b}\mathbb{E}\left[\mathbf{Q}_{2,t+1}\mathbf{\Sigma}\mathbf{Q}_{2,t+1}^\top\right] \succeq 0.$$

■

This lemma leads to a natural upper bound on the variance error, as expressed below:

Lemma 13 *With $\gamma < \frac{\gamma_{b,max}}{2}$, the variance error of the tail-averaged iterate $\overline{\mathbf{w}}^{variance}_{s+1,N}$ is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}^{variance}_{s+1,N})\right] - L(\mathbf{w}^*) \leq \frac{2}{Nb}\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).$$

Proof Considering the variance error of tail-averaged iterate from equation 25:

$$\begin{aligned}
\mathbb{E}\left[L(\overline{\mathbf{w}}^{\text{variance}}_{s+1,N})\right] - L(\mathbf{w}^*) &= \frac{2}{\gamma N^2}\cdot\sum_{l=s+1}^{s+N}\left(\operatorname{Tr}\left(\mathbb{E}\left[\mathbf{\Phi}^{\text{variance}}_l\right]\right)\right) \\
&\leq \frac{2}{\gamma N}\cdot\operatorname{Tr}\left(\mathbb{E}\left[\mathbf{\Phi}^{\text{variance}}_\infty\right]\right) \qquad \text{(from lemma 12)} \\
&= \frac{2}{\gamma N}\cdot\frac{\gamma^2}{b}\cdot\operatorname{Tr}\left(\sum_{k=0}^{\infty}(\mathcal{I}-\gamma\mathcal{T}_b)^k\mathbf{\Sigma}\right) \qquad \text{(from equation 14)} \\
&= \frac{2}{Nb}\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).
\end{aligned}$$

■

Lemma 14 *With $\gamma < \frac{\gamma_{b,max}}{2}$, the variance error of the* **final** *iterate $\mathbf{w}^{variance}_N$, obtained by running mini-batch SGD for N steps is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}^{variance}_N)\right] - L(\mathbf{w}^*) \leq \frac{\gamma}{2b}\operatorname{Tr}\left(\mathbf{H}\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).$$

Proof We note that since we deal with the square loss case,

$$\begin{aligned}
\mathbb{E}\left[L(\mathbf{w}^{\text{variance}}_N)\right] - L(\mathbf{w}^*) &= \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}\mathbf{\Phi}^{\text{variance}}_N\right) \\
&\leq \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}\mathbf{\Phi}^{\text{variance}}_\infty\right) \quad \text{(using lemma 12)} \\
&= \frac{\gamma^2}{2b}\operatorname{Tr}\left(\mathbf{H}\sum_{j=0}^{\infty}(\mathcal{I}-\gamma\mathcal{T}_b)^j\mathbf{\Sigma}\right) \\
&= \frac{\gamma}{2b}\operatorname{Tr}\left(\mathbf{H}\mathcal{T}_b^{-1}\mathbf{\Sigma}\right).
\end{aligned}$$

■

Lemma 15 *Denoting the assumption (A) $\gamma \leq \gamma_{b,max}/2$,*

1. *With (A) in place, $\mathcal{T}_b \succeq 0$.*
2. *With (A) in place, $\mathcal{T}_b^{-1}\mathbf{W} \succeq 0$ for every $\mathbf{W} \in \mathcal{S}(d),\ \mathbf{W} \succeq 0$.*
3. $\operatorname{Tr}\left((\mathcal{H}_{\mathcal{R}}+\mathcal{H}_{\mathcal{L}})^{-1}\mathbf{A}\right) = \frac{1}{2}\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right)\ \forall\ \mathbf{A} \in \mathcal{S}^+(\mathbb{R}^d)$.

4. *With (A) in place,*

$$\operatorname{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq 2\operatorname{Tr}\left(\mathbf{H}^{-1}\mathbf{\Sigma}\right).$$

Proof

Proof of claim 1 in Lemma 15: $\mathcal{T}_b \succeq 0$ implies that for all symmetric matrices $\mathbf{A} \in \mathcal{S}(d)$, we have $\operatorname{Tr}\left(\mathbf{A}\mathcal{T}_b\mathbf{A}\right) \geq 0$, and this is true owing to the following inequalities:

$$\begin{aligned}
\langle \mathbf{A}, \mathcal{T}_b\mathbf{A}\rangle &= 2\operatorname{Tr}\left(\mathbf{AHA}\right) - \frac{\gamma}{b}\mathbb{E}\left[\langle \mathbf{x}, \mathbf{Ax}\rangle^2\right] - \frac{\gamma(b-1)}{b}\langle \mathbf{H}, \mathbf{AHA}\rangle \\
&\geq 2\operatorname{Tr}\left(\mathbf{AHA}\right) - \frac{\gamma}{b}\mathbb{E}\left[\|\mathbf{x}\|^2\|\mathbf{Ax}\|^2\right] - \frac{\gamma(b-1)}{b}\|\mathbf{H}\|\operatorname{Tr}\left(\mathbf{AHA}\right) \\
&\geq 2\operatorname{Tr}\left(\mathbf{AHA}\right) - \frac{\gamma}{b}R^2\mathbb{E}\left[\|\mathbf{Ax}\|^2\right] - \frac{\gamma(b-1)}{b}\|\mathbf{H}\|\operatorname{Tr}\left(\mathbf{AHA}\right) \\
&\geq \left(2 - \frac{\gamma}{b}\left(R^2 + (b-1)\|\mathbf{H}\|\right)\right)\operatorname{Tr}\left(\mathbf{AHA}\right).
\end{aligned}$$

Using the definition of $\gamma_{b,\max}$ completes the proof of the claim.

Proof of claim 2 in Lemma 15: We require to prove $\mathcal{T}_b^{-1}$ operating on a PSD matrix produces a PSD matrix, or in other words, $\mathcal{T}_b^{-1}$ is a PSD map.

$$\begin{aligned}
\mathcal{T}_b^{-1} &= [\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})]^{-1} \\
&= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{\frac{1}{2}}[\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})]^{-1}\cdot \\
&\qquad\qquad (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{\frac{1}{2}}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \\
&= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}[\mathcal{I} - \frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}]^{-1}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}.
\end{aligned} \tag{28}$$

Now, we prove that $\|\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\| < 1$. Given $\gamma < \gamma_{b,\max}/2$, we employ claim 1 to note that $\mathcal{T}_b \succ 0$.

$$\begin{aligned}
&\mathcal{T}_b \succ 0 \\
&\quad\Rightarrow \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}) \succ 0 \\
&\quad\Rightarrow \frac{\gamma}{b}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}) \prec \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} \\
&\quad\Rightarrow \frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \prec \mathcal{I} \\
&\quad\Rightarrow \|\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\| < 1.
\end{aligned}$$

With this fact in place, we employ Taylor series to expand $\mathcal{T}_b^{-1}$ in equation 28, i.e.:

$$\begin{aligned}
\mathcal{T}_b^{-1} &= (\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}\sum_{i=0}^{\infty}(\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R})(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}})^i(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-\frac{1}{2}} \\
&= \sum_{i=0}^{\infty}(\frac{\gamma}{b}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}(\mathcal{M} + (b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}))^i(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}.
\end{aligned}$$

The proof completes by employing the following facts: Using Lyapunov's theorem (Bhatia (2007) proposition A 1.2.6), we know $(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}$ is a PSD map, i.e. if $(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})^{-1}(A) = B$, then, if A is PSD $\implies$ B is PSD. Furthermore, $\mathcal{M}$ is also a PSD map, i.e. if A_1 is PSD, $\mathcal{M}(A_1) = E[(x^T A_1 x)x \otimes x]$ is PSD as well. Finally, $\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}$ is also a PSD map, since, if A_2 is PSD, then, $\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}(A_2) = HA_2H$ which is PSD as well. With all these facts in place, we note that each term in the Taylor's expansion above is a PSD map implying the overall map is PSD as well, thus rounding up the proof to claim 2 in Lemma 15.

Proof of claim 3 in Lemma 15:

We know that the operator $(\mathcal{H}_\mathcal{R} + \mathcal{H}_\mathcal{L})^{-1}$ is a PSD map, i.e, it maps PSD matrices to PSD matrices. Since $\mathbf{A} \succeq 0$, we replace this condition with $\mathbf{U} = (\mathcal{H}_\mathcal{R} + \mathcal{H}_\mathcal{L})^{-1}\mathbf{A} \succeq 0$ implying, we need to show the following:

$$\mathrm{Tr}\left(\mathbf{U}\right) = \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right) \ \forall\, \mathbf{U} \succeq 0.$$

Examining the right hand side, we see the following:

$$\begin{aligned}\frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{A}\right) &= \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}(\mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R})\mathbf{U}\right) \\ &= \frac{1}{2}\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{H}\mathbf{U} + \mathbf{H}^{-1}\mathbf{U}\mathbf{H}\right) \\ &= \mathrm{Tr}\left(\mathbf{U}\right).\end{aligned}$$

thus wrapping up the proof of claim 4.

Proof of claim 4 in Lemma 15: Let $\mathcal{U} = \mathcal{H}_\mathcal{L} + \mathcal{H}_\mathcal{R} - \frac{\gamma}{b}\cdot(b-1)\mathcal{H}_\mathcal{L}\mathcal{H}_\mathcal{R}$. Then,

$$\begin{aligned}\mathcal{T}_b^{-1}\mathbf{\Sigma} &= \left(\mathcal{U} - \frac{\gamma}{b}\mathcal{M}\right)^{-1}\mathbf{\Sigma} \\ &= \sum_{i=0}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i\mathcal{U}^{-1}\mathbf{\Sigma}.\end{aligned}$$

Let $\mathbf{A} = \mathcal{U}^{-1}\mathbf{\Sigma}$, $\mathbf{A}' = \mathcal{U}^{-1}\mathbf{H}$. Then,

$$\mathcal{T}_b^{-1}\mathbf{\Sigma} = \sum_{i=1}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i\mathbf{A}.$$

The $i = 0$ term is just $= \mathbf{A}$. Now, considering $i = 1$, we have:

$$\begin{aligned}\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\mathbf{A} &\preceq \frac{\gamma}{b}\|\mathbf{A}\|_2\mathcal{U}^{-1}\mathcal{M}\mathbf{I} \\ &\preceq \frac{\gamma}{b}\|\mathbf{A}\|_2R^2\mathcal{U}^{-1}\mathbf{H} = \frac{\gamma}{b}\|\mathbf{A}\|_2R^2\mathbf{A}'.\end{aligned}$$

Next, considering $i = 2$, we have:

$$\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^2\mathbf{A} = \left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\cdot\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\mathbf{A}$$

$$\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right) \cdot \left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)\mathbf{A}'$$
$$\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right) \cdot \left(\frac{\gamma}{b}\mathcal{U}^{-1}\right) \cdot \|\mathbf{A}'\|_2 \cdot R^2\mathbf{H}$$
$$\preceq \left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right) \cdot \left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right) \cdot \mathbf{A}'.$$

Noting this recursive structure, we see that:

$$\begin{aligned}
\mathcal{T}_b{}^{-1}\mathbf{\Sigma} &= \sum_{i=0}^{\infty}\left(\frac{\gamma}{b}\mathcal{U}^{-1}\mathcal{M}\right)^i \mathbf{A} \\
&\preceq \mathbf{A} + \sum_{i=1}^{\infty}\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right) \cdot \left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)^{i-1} \cdot \mathbf{A}' \\
&= \mathbf{A} + \frac{\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)}{1 - \left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)} \cdot \mathbf{A}'.
\end{aligned}$$

Note that this summation is finite iff $\gamma \leq \frac{b}{R^2\|\mathbf{A}'\|_2}$. Further, applying the trace operator on both sides, we have:

$$\operatorname{Tr}\left(\mathcal{T}_b{}^{-1}\mathbf{\Sigma}\right) \leq \operatorname{Tr}\left(\mathbf{A}\right) + \frac{\left(\frac{\gamma}{b}\|\mathbf{A}\|_2 R^2\right)}{1 - \left(\frac{\gamma}{b}\|\mathbf{A}'\|_2 R^2\right)} \operatorname{Tr}\left(\mathbf{A}'\right). \tag{29}$$

Now, for any psd matrix $\mathbf{B} \succeq 0$, let us upperbound $\mathcal{U}^{-1}\mathbf{B}$:

$$\mathcal{U}^{-1}\mathbf{B} = \sum_{j=0}^{\infty}\left(\gamma \cdot \frac{b-1}{b} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1} \cdot \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}}\right)^i (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}.$$

The recursion can be bounded by analyzing $i = 1$:

$$\begin{aligned}
&\gamma \cdot \frac{b-1}{b} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1} \cdot \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{B} \\
&\preceq \|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{B}\|_2 \cdot \gamma \cdot \frac{b-1}{b} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1} \cdot \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}} \cdot \mathbf{I} \\
&\preceq \|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{B}\|_2 \cdot \gamma \cdot \frac{b-1}{b} \cdot (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1} \cdot \|\mathbf{H}\|_2\mathbf{H} \\
&= \|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{B}\|_2 \cdot \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2 \cdot \mathbf{I}.
\end{aligned}$$

This indicates the means to recurse for bounding terms $i \geq 2$:

$$\mathcal{U}^{-1}\mathbf{B} \preceq \sum_{j=0}^{\infty}\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{B}\|_2\left(\gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2\right)^j \cdot \mathbf{I}$$

$$= \frac{\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{B}\|_2}{1 - \gamma \cdot \frac{(b-1)\|\mathbf{H}\|_2}{2b}} \cdot \mathbf{I}.$$

The upperbound above is true as long as $\gamma < \frac{2b}{(b-1)\|\mathbf{H}\|_2}$. This now allows us to obtain bounds on $\|\mathbf{A}\|_2, \|\mathbf{A}'\|_2, \mathrm{Tr}\left(\mathbf{A}'\right)$:

$$\|\mathbf{A}\|_2 \leq \frac{\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{1 - \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2}$$

$$\|\mathbf{A}'\|_2 \leq \frac{1/2}{1 - \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2}$$

$$\mathrm{Tr}\left(\mathbf{A}'\right) \leq \frac{d/2}{1 - \gamma \cdot \frac{b-1}{2b} \cdot \|\mathbf{H}\|_2}.$$

Substituting these in equation 29:

$$\mathrm{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq \mathrm{Tr}\left(\mathbf{A}\right) + \frac{\frac{\gamma R^2}{2b} \cdot d\|(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{\left(1 - \frac{\gamma}{2b} \cdot (R^2 + (b-1)\|\mathbf{H}\|_2)\right) \cdot \left(1 - \gamma \cdot \frac{b-1}{2b}\|\mathbf{H}\|_2\right)}. \tag{30}$$

with the conditions on γ being: $\gamma \leq \frac{2b}{(b-1)\|\mathbf{H}\|_2}$, $\gamma \leq \frac{2b}{R^2+(b-1)\|\mathbf{H}\|_2}$, $\gamma \leq \frac{2b}{R^2}$. These are combined using $\gamma \leq \frac{2b}{R^2+(b-1)\|\mathbf{H}\|_2}$. Once this condition is satisfied, the denominator of the second term can be upperbounded by atmost a constant. Next, looking at the numerator of the second term, we see that $\gamma \leq \frac{2b}{R^2 \cdot \frac{d\|(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\|_2}{\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)}} = \frac{2b}{R^2\rho_{\mathrm{m}}}$ allows for the second term to be upperbounded by $\mathcal{O}(\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right))$. This is clearly satisfied if $\gamma \leq \frac{2b}{R^2 \cdot \rho_{\mathrm{m}} + (b-1)\|\mathbf{H}\|_2}$. In particular, setting γ to be half of this maximum, we have:

$$\mathrm{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) \leq \mathrm{Tr}\left(\mathbf{A}\right) + 2\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right). \tag{31}$$

Denoting $\hat{\mathbf{\Sigma}} = (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}} - \gamma \cdot \frac{b-1}{b} \cdot \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}$, in order to bound $\mathrm{Tr}\left(\mathbf{A}\right)$, we require comparing $\mathrm{Tr}\left(\hat{\mathbf{\Sigma}}\right)$ with $\mathrm{Tr}\left(\tilde{\mathbf{\Sigma}}\right) = \mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right)$. For this, without loss of generality, we can consider $\mathbf{H}$ to be diagonal, and this implies that comparing the diagonal elements of $\hat{\mathbf{\Sigma}}_{ii} = \mathbf{\Sigma}_{ii}/(2\lambda_i - \gamma\frac{b-1}{b}\lambda_i^2)$ while $\tilde{\mathbf{\Sigma}}_{ii} = \mathbf{\Sigma}_{ii}/2\lambda_i$. Comparing these, we see that

$$\begin{aligned}\mathrm{Tr}\left(\hat{\mathbf{\Sigma}}\right) = \mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}} - \gamma \cdot \frac{b-1}{b} \cdot \mathcal{H}_{\mathcal{L}}\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right) &\leq \frac{1}{1 - \gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\,\mathrm{Tr}\left(\tilde{\mathbf{\Sigma}}\right) \\ &= \frac{1}{1 - \gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

Noting that $\mathrm{Tr}\left(\mathbf{A}\right) = \mathrm{Tr}\left(\hat{\mathbf{\Sigma}}\right)$, we see that substituting the above in equation 31, we have:

$$\begin{aligned}\mathrm{Tr}\left(\mathcal{T}_b^{-1}\mathbf{\Sigma}\right) &\leq \frac{1}{1 - \gamma\frac{b-1}{2b}\|\mathbf{H}\|_2}\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right) + 2\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right) \\ &\leq 4\,\mathrm{Tr}\left((\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right) = 2\,\mathrm{Tr}\left(\mathbf{H}^{-1}\mathbf{\Sigma}\right).\end{aligned}$$

■

Corollary 16 *Consider the mis-specified case of the streaming LSR problem. With* $\gamma \leq \frac{\gamma_{b,max}}{2}$*, the variance error of the tail-averaged iterate* $\overline{\mathbf{w}}_{s+1,N}^{variance}$ *is upper bounded as:*

$$\mathbb{E}\left[L(\overline{\mathbf{w}}_{s+1,N}^{variance})\right] - L(\mathbf{w}^*) \leq \frac{4}{Nb} \cdot \widehat{\sigma^2_{MLE}}.$$

Proof The result follows in a straightforward manner by noting that $\gamma \leq \frac{\gamma_{b,\max}}{2}$ implying that $\operatorname{Tr}\left(\mathcal{T}_b^{-1}\boldsymbol{\Sigma}\right) \leq 2\operatorname{Tr}\left(\mathbf{H}^{-1}\boldsymbol{\Sigma}\right)$ and by substituting into the result of lemma 13. ■

Corollary 17 *With* $\gamma \leq \frac{\gamma_{b,max}}{2}$*,* $\boldsymbol{\Sigma} = \sigma^2\mathbf{H}$ *the variance error of the* **final** *iterate* $\mathbf{w}_N^{variance}$*, obtained by running mini-batch SGD for* N *steps is upper bounded as:*

$$\mathbb{E}\left[L(\mathbf{w}_N^{variance})\right] - L(\mathbf{w}^*) \leq \frac{\gamma\sigma^2}{2b}\operatorname{Tr}\mathbf{H}.$$

Proof This follows from the fact that $\mathcal{T}_b^{-1}\boldsymbol{\Sigma} \preceq \sigma^2\mathbf{I}$, implying that $\mathbf{H}\mathcal{T}_b^{-1}\boldsymbol{\Sigma} \preceq \sigma^2\mathbf{H}$ and then applying the trace operator on the result of lemma 14. ■

A.5 Main Results

A.5.1 DERIVATION OF DIVERGENT LEARNING RATE

A necessary condition for the convergence of Stochastic Gradient Updates is $\mathcal{T}_b \succeq 0$, and this by definition implies,

$$\begin{aligned}
&\langle \mathbf{W}, \mathcal{T}_b\mathbf{W}\rangle \geq 0, \quad \mathbf{W} \in \mathcal{S}(d) \\
\implies &2\operatorname{Tr}(\mathbf{WHW}) - \frac{\gamma}{b}\operatorname{Tr}(\mathbf{W}\mathcal{M}\mathbf{W}) - \gamma\left(\frac{b-1}{b}\right)\operatorname{Tr}(\mathbf{WHWH}) \geq 0 \\
\implies &\frac{2}{\gamma} \geq \frac{\operatorname{Tr}(\mathbf{W}\mathcal{M}\mathbf{W}) + (b-1)\operatorname{Tr}(\mathbf{WHWH})}{b\operatorname{Tr}(\mathbf{WHW})} \\
\implies &\frac{2}{\gamma^{div}_{b,\max}} = \sup_{\mathbf{W}\in\mathcal{S}(d)} \frac{\operatorname{Tr}(\mathbf{W}\mathcal{M}\mathbf{W}) + (b-1)\operatorname{Tr}(\mathbf{WHWH})}{b\operatorname{Tr}(\mathbf{WHW})}.
\end{aligned}$$

A.5.2 PROOF OF THEOREM 1

Proof [proof of Theorem 1] The proof of theorem 1 follows from characterizing bias-variance decomposition for the tail-averaged iterate in section A.2.3 with equation 23.

The bias error of the tail-averaged iterate (equation 24) is bounded with lemma 9 and lemma 10 in section A.3.

The variance error of the tail-averaged iterate (equation 25) is bounded with lemma 12, lemma 13, lemma 15 and corollary 16 in section A.4.

The final expression follows through substituting the result of lemma 10 and corollary 16 into equation 23, with appropriate parameters of the problem, i.e., with a batch size b, number of burn-in iterations s, number of tail-averaged iterations $n/b - s$ to provide the claimed excess risk bound of

Algorithm 1:

$$\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) \leq \frac{2}{\gamma^2\mu^2(\frac{n}{b}-s)^2} \cdot (1-\gamma\mu)^s \cdot \left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{b \cdot (\frac{n}{b}-s)}.$$

■

A.5.3 PROOF OF LEMMA 4

Proof [proof of Lemma 4] The proof of lemma 4 follows from characterizing bias-variance decomposition for the final iterate in section A.2.2 with equation 18.

The bias error of the final iterate is bounded with lemma 9 and lemma 11 in section A.3.

The variance error of the final iterate is bounded with lemma 12, lemma 14, lemma 15 and corollary 17 in section A.4.

The final expression follows through substituting the result of lemma 11 and corollary 17 into equation 18, with appropriate parameters of the problem, i.e., with a batch size b, number of samples n and number of iterations $\lfloor n/b \rfloor$, to provide the claimed excess risk bound:

$$\mathbb{E}\left[L(\mathbf{w}_{\lfloor n/b \rfloor})\right] - L(\mathbf{w}^*) \leq \kappa_b(1-\gamma\mu)^{\lfloor n/b \rfloor}\left(L(\mathbf{w}_0) - L(\mathbf{w}^*)\right) + \frac{\gamma}{b}\sigma^2 \operatorname{Tr}(\mathbf{H}).$$

■

A.5.4 PROOF OF THEOREM 5

Proof Let $\widetilde{L_e} = \mathbb{E}\left[L(\mathbf{w}_e)\right] - L(\mathbf{w}^*)$. We will first provide a recursive bound for $\widetilde{L_e}$ for $e \leq \log\left(\frac{n}{bt}\right) - 1$ using theorem 1, with a mini-batch size of $b_e = 1 + 2^{e-1}b$, where, $b = b_{\text{thresh}} - 1$, $n_e = b_e \cdot t$, $s = t - 1$:

$$\begin{aligned}
\widetilde{L_e} &\leq 2\kappa_{b_e}^2 \exp\left(-\frac{n_e}{b_e \cdot \kappa_{b_e}}\right)\widetilde{L_{e-1}} + 4\frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e} \\
&\leq \exp\left(-\frac{n_e}{3b_e\kappa_e \log(\kappa_e)}\right) \cdot \widetilde{L_{e-1}} + 4 \cdot \frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e}.
\end{aligned}$$

Next, denote $\kappa = \|\mathbf{H}\|_2 / \mu$; now, let us bound κ_{b_e}:

$$\begin{aligned}
\kappa_{b_e} &= \frac{R^2 \cdot \frac{d\left\|(\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right\|_2}{\operatorname{Tr}((\mathcal{H}_{\mathcal{L}}+\mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma})} + (b_e - 1)\|\mathbf{H}\|_2}{b_e\mu} \\
&= \kappa \cdot \frac{b_{\text{thresh}} - 1 + b_e - 1}{b_e} = \kappa \cdot \frac{b_{\text{thresh}} - 1 + 2^{e-1}(b_{\text{thresh}} - 1)}{2^{e-1}(b_{\text{thresh}} - 1)} \\
&= \kappa \cdot \frac{1 + 2^{e-1}}{2^{e-1}} \leq 2\kappa.
\end{aligned}$$

This implies $\kappa_{b_e}\log(\kappa_{b_e}) \leq 4\kappa\log(\kappa)$. This implies, revisiting the recursion on $\widetilde{L_e}$, we have:

$$\begin{aligned}
\widetilde{L_e} &\leq \exp\Big(-\frac{n_e}{12b_e\kappa\log(\kappa)}\Big)\cdot\widetilde{L_{e-1}} + 4\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{b_e}\\
&\leq \exp\Big(-\frac{t}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_{e-1}} + 4\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{2^{e-1}b}\\
&\leq \exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + \frac{4\widehat{\sigma^2_{\text{MLE}}}}{b}\cdot\sum_{j=1}^{e}\frac{\exp\Big(-\frac{t(j-1)}{12\kappa\log(\kappa)}\Big)}{2^{e-j}}\\
&\leq \exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + \frac{4\widehat{\sigma^2_{\text{MLE}}}}{b}\cdot\frac{1/2^{e-1}}{1-2\cdot\exp\big(-\frac{t}{12\kappa\log\kappa}\big)}\\
&\leq \exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + \frac{12\widehat{\sigma^2_{\text{MLE}}}}{2^e b} \quad (\text{since } t > 24\kappa\log(\kappa))\\
&= \exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + \frac{12\widehat{\sigma^2_{\text{MLE}}}}{b\cdot n}\cdot(4bt) \quad (\text{since } 2^e = n/(4bt))\\
&= \exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + 48\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}t}{n}. \qquad (32)
\end{aligned}$$

Next, for the final epoch, we have $b = n/2t$, $s = t/2$, and a total of $n/2$ samples, implying:

$$\begin{aligned}
\widetilde{L_{e+1}} &\leq \frac{2\kappa_b^2}{\Big(t/2\Big)^2}\cdot\exp\big(-\frac{t}{2\kappa_b}\big)\widetilde{L_e} + 4\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{b\cdot\big(n/4b\big)} = \frac{8\kappa_b^2}{t^2}\cdot\exp\big(-\frac{t}{2\kappa_b}\big)\cdot\widetilde{L_e} + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\leq \frac{32\kappa^2}{t^2}\cdot\exp\big(-\frac{t}{4\kappa}\big)\cdot\widetilde{L_e} + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n} \quad (\text{since } \kappa_b \leq 2\kappa)\\
&\leq \frac{32\kappa^2}{t^2}\cdot\exp\big(-\frac{t}{4\kappa}\big)\cdot\bigg(\exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + 48\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}t}{n}\bigg) + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\leq \frac{32\kappa^2}{t^2}\cdot\exp\big(-\frac{t}{4\kappa}\big)\cdot\exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + 64\kappa\exp\big(-t/4\kappa\big)\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n} + 16\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\leq \frac{32\kappa^2}{t^2}\cdot\exp\big(-\frac{t}{4\kappa}\big)\cdot\exp\Big(-\frac{te}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + 80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&\leq \exp\Big(-\frac{t(e+1)}{12\kappa\log(\kappa)}\Big)\cdot\widetilde{L_0} + 80\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}\\
&= \bigg(\frac{2bt}{n}\bigg)^{\frac{t}{12\kappa\log(\kappa)}}\widetilde{L_0} + 80\cdot\frac{\widehat{\sigma^2_{\text{MLE}}}}{n}, \qquad (33)
\end{aligned}$$

which rounds up the proof of the theorem. ■

A.5.5 PROOF OF THEOREM 6

Proof For analyzing the parameter mixing scheme, we require tracking the progress of the i^{th} machine's SGD updates using its centered estimate $\boldsymbol{\eta}_k^{(i)}$. Furthermore, the tail-averaged iterate for the i^{th} machine is represenTeed as $\bar{\boldsymbol{\eta}}^{(i)} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{k=s+1}^{s+N} \boldsymbol{\eta}_k^{(i)}$. Finally, the model averaged estimate is represented with its own centered estimate defined as $\bar{\boldsymbol{\eta}} = \frac{1}{P}\sum_{i=1}^{P} \bar{\boldsymbol{\eta}}^{(i)}$. Now, in a manner similar to standard mini-batch tail-averaged SGD on a single machine, the model averaged iterate admits its own bias variance decomposition, through which $\bar{\boldsymbol{\eta}} = \bar{\boldsymbol{\eta}}^{\text{bias}} + \bar{\boldsymbol{\eta}}^{\text{variance}}$ and an upperbound on the excess risk is written as:

$$\begin{aligned}\mathbb{E}\left[L(\overline{\mathbf{w}})\right] - L(\mathbf{w}^*) &= \mathbb{E}\left[\frac{1}{2}\langle(\overline{\mathbf{w}} - \mathbf{w}^*), \mathbf{H}(\overline{\mathbf{w}} - \mathbf{w}^*)\rangle\right] = \mathbb{E}\left[\frac{1}{2}\langle\bar{\boldsymbol{\eta}}, \mathbf{H}\bar{\boldsymbol{\eta}}\rangle\right] \\ &\leq \mathbb{E}\left[\langle\bar{\boldsymbol{\eta}}^{\text{bias}}, \mathbf{H}\bar{\boldsymbol{\eta}}^{\text{bias}}\rangle\right] + \mathbb{E}\left[\langle\bar{\boldsymbol{\eta}}^{\text{variance}}, \mathbf{H}\bar{\boldsymbol{\eta}}^{\text{variance}}\rangle\right].\end{aligned}$$

We will first handle the variance since it is straightforward given that the noise $\boldsymbol{\zeta}$ is independent for different machines SGD runs. What this implies is the following:

$$\begin{aligned}\bar{\boldsymbol{\eta}}^{\text{variance}} &= \frac{1}{P}\sum_{i=1}^{P} \bar{\boldsymbol{\eta}}^{(i),\text{variance}} \\ \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{\text{variance}}\right] &= \frac{1}{P^2}\sum_{i,j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{variance}}\right] \\ &= \frac{1}{P^2}\left(\sum_{i}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{(i),\text{variance}}\right] + \sum_{i\neq j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{variance}}\right]\right) \\ &= \frac{1}{P}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(1),\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{(1),\text{variance}}\right]. \end{aligned} \tag{34}$$

Where, the final line follows because $\forall\ i \neq j$, the terms are in expectation equal to zero since in expectation each of the noise terms is zero (from first order optimality conditions). The other observation is that the only terms left are P independent runs of tail-averaged SGD in each of the machine, whose risk is straightforward to bound from corollary 16. This implies

$$\langle\mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{variance}} \otimes \bar{\boldsymbol{\eta}}^{\text{variance}}\right]\rangle \leq \frac{4}{PNb}\cdot\widehat{\sigma^2_{\text{MLE}}}. \quad \text{(using corollary 16)} \tag{35}$$

Next, let us consider the bias error:

$$\begin{aligned}\bar{\boldsymbol{\eta}}^{\text{bias}} &= \frac{1}{P}\sum_{i=1}^{P} \bar{\boldsymbol{\eta}}^{(i),\text{bias}} \\ \implies \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{\text{bias}}\right] &= \frac{1}{P^2}\sum_{i,j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] \\ &= \frac{1}{P^2}\Bigg(\sum_{i=1}^{P}\underbrace{\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(i),\text{bias}}\right]}_{\text{independent runs of tail-averaged SGD}} + \sum_{i\neq j}\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right]\Bigg),\end{aligned} \tag{36}$$

which implies that we require bounding $\forall\ i \neq j$, $\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right]$.

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] &= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_k^{(i),\text{bias}} \otimes \boldsymbol{\eta}_l^{(j),\text{bias}}\right] \\
&= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\boldsymbol{\eta}_k^{(i),\text{bias}}\right] \otimes \mathbb{E}\left[\boldsymbol{\eta}_l^{(j),\text{bias}}\right] \\
&= \frac{1}{N^2}\sum_{k,l=s+1}^{s+N} \mathbb{E}\left[\mathbf{Q}_{1:k}^{(i)}\boldsymbol{\eta}_0\right] \otimes \mathbb{E}\left[\mathbf{Q}_{1:l}^{(j)}\boldsymbol{\eta}_0\right] \quad \text{(from equation 15)} \\
&= \frac{1}{N^2}\Bigg(\sum_{k=s+1}^{s+N}(\mathbf{I}-\gamma\mathbf{H})^k\Bigg)\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0\Bigg(\sum_{l=s+1}^{s+N}(\mathbf{I}-\gamma\mathbf{H})^l\Bigg) \\
&\preceq \frac{1}{N^2}\Bigg(\sum_{k=s+1}^{\infty}(\mathbf{I}-\gamma\mathbf{H})^k\Bigg)\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0\Bigg(\sum_{l=s+1}^{\infty}(\mathbf{I}-\gamma\mathbf{H})^l\Bigg) \\
&= \frac{1}{\gamma^2 N^2}\mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}.
\end{aligned}$$

This implies that,

$$\begin{aligned}
\mathbb{E}\left[\bar{\boldsymbol{\eta}}^{(i),\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{(j),\text{bias}}\right] &\leq \frac{1}{\gamma^2 N^2}\cdot\langle \mathbf{H}, \mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0 \otimes \boldsymbol{\eta}_0(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}\rangle \\
&= \frac{1}{\gamma^2 N^2}\cdot\boldsymbol{\eta}_0^\top(\mathbf{I}-\gamma\mathbf{H})^{s+1}\mathbf{H}^{-1}\mathbf{H}\mathbf{H}^{-1}(\mathbf{I}-\gamma\mathbf{H})^{s+1}\boldsymbol{\eta}_0 \\
&\leq \frac{(1-\gamma\mu)^{2s+2}}{\mu\gamma^2 N^2}\|\boldsymbol{\eta}_0\|^2 \leq \frac{(1-\gamma\mu)^{2s+2}}{\mu^2\gamma^2 N^2}\cdot\Big(L(\mathbf{w}_0)-L(\mathbf{w}^*)\Big). \qquad (37)
\end{aligned}$$

Combining the bound for the cross terms in equation 37 and lemma 10 for the self-terms, we get:

$$\langle \mathbf{H}, \mathbb{E}\left[\bar{\boldsymbol{\eta}}^{\text{bias}} \otimes \bar{\boldsymbol{\eta}}^{\text{bias}}\right]\rangle \leq \frac{(1-\gamma\mu)^{s+1}}{\mu^2\gamma^2 N^2}\cdot\frac{2+(1-\gamma\mu)^{s+1}\cdot(P-1)}{P}\cdot\Big(L(\mathbf{w}_0)-L(\mathbf{w}^*)\Big). \qquad (38)$$

The proof wraps up by substituting the relation $N = n/(P\cdot b) - s$ in equations 35 and 38. ■

A.5.6 PROOF OF LEMMA 3

For this problem instance, we begin by noting that $(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}$ is diagonal as well, with entries:

$$\{(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}\}_{ii} = \frac{1}{2}\{\mathbf{H}^{-1}\boldsymbol{\Sigma}\}_{ii} = \begin{cases} 1/2 & \text{if } i=1 \\ 1/2(d-1) & \text{if } i>1 \end{cases}.$$

Let us consider the case with batch size $b=1$. With the appropriate choice of step size γ that ensure contracting operators, we require considering $\text{Tr}\left(\mathcal{T}_b{}^{-1}\boldsymbol{\Sigma}\right)$ as in equation 29, which corresponds to bounding the leading order term in the variance. We employ the taylor's expansion (just as in claim 2 of lemma 15) to expand the term of interest $\mathcal{T}_b{}^{-1}\boldsymbol{\Sigma}$:

$$\mathcal{T}_b{}^{-1}\boldsymbol{\Sigma} = \sum_{i=0}^{\infty}\left(\gamma(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\mathcal{M}\right)^i(\mathcal{H}_\mathcal{L}+\mathcal{H}_\mathcal{R})^{-1}\boldsymbol{\Sigma}$$

$$= (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma} + \sum_{i=1}^{\infty} \left(\gamma(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathcal{M}\right)^i (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}$$

$$\Rightarrow \operatorname{Tr} \mathcal{T}_b^{-1}\mathbf{\Sigma} = \operatorname{Tr}(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma} + \sum_{i=1}^{\infty} \operatorname{Tr}\left[\left(\gamma(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathcal{M}\right)^i (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right]$$

$$\operatorname{Tr} \mathcal{T}_b^{-1}\mathbf{\Sigma} = \frac{1}{2}\operatorname{Tr} \mathbf{H}^{-1}\mathbf{\Sigma} + \sum_{i=1}^{\infty} \operatorname{Tr}\left[\left(\gamma(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathcal{M}\right)^i (\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}\right].$$

We observe that the term corresponding to $i = 0$ works out regardless of the choice of stepsize γ; we then switch our attention to the second term, i.e., the term corresponding to $i = 1$:

$$\operatorname{Tr}\left(\gamma(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathcal{M}\right)(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma} = \frac{d+2}{4} \cdot \operatorname{Tr}(\mathbf{\Sigma}).$$

We require that this term should be $\leq \operatorname{Tr}(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}$, implying,

$$\gamma < \frac{4\operatorname{Tr}(\mathcal{H}_{\mathcal{L}} + \mathcal{H}_{\mathcal{R}})^{-1}\mathbf{\Sigma}}{(d+2)\operatorname{Tr}(\mathbf{\Sigma})}.$$

For this example, we observe that this yields $\gamma < \frac{4}{(d+2)(1+\frac{1}{d})}$, which clearly is off by a factor d compared to the well-specified case which requires $\gamma < \frac{d}{(d+2)(1+\frac{1}{d})}$, establishing a clear separation between the step sizes required by SGD for the well-specified and mis-specified cases.

A.5.7 PROOFS OF SUPPORTING LEMMAS

Proof of lemma 7

Proof [Proof of lemma 7] We begin by considering $\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{bias}} \otimes \boldsymbol{\eta}_t^{\text{bias}}\right]\rangle$:

$$\begin{aligned}
\langle \mathbf{I}, \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{bias}} \otimes \boldsymbol{\eta}_t^{\text{bias}}\right]\rangle &= \mathbb{E}\left[\|\boldsymbol{\eta}_t^{\text{bias}}\|^2\right] \\
&= \mathbb{E}\left[(\boldsymbol{\eta}_{t-1}^{\text{bias}})^\top \left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b} \mathbf{x}_{ti}\mathbf{x}_{ti}^\top\right)\left(\mathbf{I} - \frac{\gamma}{b}\sum_{i=1}^{b} \mathbf{x}_{ti}\mathbf{x}_{ti}^\top\right)\boldsymbol{\eta}_{t-1}^{\text{bias}}\right] \\
&\leq (1 - \gamma\mu) \cdot \mathbb{E}\left[\|\boldsymbol{\eta}_{t-1}^{\text{bias}}\|^2\right] \qquad \text{(from lemma 9)},
\end{aligned}$$

from where the lemma follows through substitution of $\gamma = \gamma_{b,\max}/2$. ■

Proof of lemma 8

Proof [Proof of lemma 8] From equation 27, we have that:

$$\begin{aligned}
\mathbf{\Phi}_t^{\text{variance}} &= \mathbb{E}\left[\boldsymbol{\eta}_t^{\text{variance}} \otimes \boldsymbol{\eta}_t^{\text{variance}}\right] \\
&= \frac{\gamma^2}{b}\left(\sum_{k=0}^{t-1}(\mathcal{I} - \gamma\mathcal{T}_b)^k\right)\mathbf{\Sigma}.
\end{aligned}$$

Allowing $t \to \infty$, we have:

$$\mathbf{\Phi}_\infty^{\text{variance}} = \frac{\gamma}{b}\mathcal{T}_b^{-1}\mathbf{\Sigma} \preceq \frac{\gamma}{b} \cdot \sigma^2\mathbf{I} \quad \text{(from claim 4 in lemma 15 since } \gamma \leq \gamma_{b,\max}/2,\ \mathbf{\Sigma} = \sigma^2\mathbf{H}).$$

Substituting $\gamma = \gamma_{b,\max}/2$, the result follows. ■

References

Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 2012.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *CoRR*, abs/1603.05953, 2016.

Dan Anbar. *On Optimal Estimation Methods Using Stochastic Approximation Procedures*. University of California, 1971.

Francis Bach and Eric Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Neural Information Processing Systems (NIPS) 24*, 2011.

Francis R. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *In Journal of Machine Learning Research (JMLR)*, volume 15, 2014.

Francis R. Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Neural Information Processing Systems (NIPS) 26*, 2013.

Rajendra Bhatia. *Positive Definite Matrices*. Princeton Series in Applied Mathematics. Princeton University Press, 2007.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Neural Information Processing Systems (NIPS) 20*, 2007.

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

Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for l1-regularized loss minimization. In *International Conference on Machine Learning (ICML)*, 2011.

Louis Augustin Cauchy. Méthode générale pour la résolution des systémes d'équations simultanees. *C. R. Acad. Sci. Paris*, 1847.

Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Neural Information Processing Systems (NIPS) 24*, 2011.

Aaron Defazio. A simple practical accelerated method for finite sums. In *Neural Information Processing Systems (NIPS) 29*, 2016.

Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Neural Information Processing Systems (NIPS) 27*, 2014.

Alexandre Défossez and Francis R. Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Artifical Intelligence and Statistics (AISTATS)*, 2015.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research (JMLR)*, volume 13, 2012.

Aymeric Dieuleveut and Francis Bach. Non-parametric stochastic approximation with large step sizes. *The Annals of Statistics*, 2015.

John C. Duchi, Sorathan Chaturapruek, and Christopher Ré. Asynchronous stochastic convex optimization. *CoRR*, abs/1508.00882, 2015.

Vaclav Fabian. Asymptotically efficient stochastic approximation; the RM case. *Annals of Statistics*, 1(3), 1973.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning (ICML)*, 2015a.

Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on Learning Theory (COLT)*, 2015b.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Prateek Jain, Chi Jin, Sham M. Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming pca: Matching matrix bernstein and near-optimal finite sample guarantees for oja's algorithm. In *Conference on Learning Theory (COLT)*, 2016a.

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

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, Venkata Krishna Pillutla, and Aaron Sidford. A markov chain theory approach to characterizing the minimax optimality of stochastic gradient descent (for least squares). *arXiv preprint arXiv:1710.09430*, 2017a.

Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent. *arXiv preprint arXiv:1704.08227*, 2017b.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Neural Information Processing Systems (NIPS) 26*, 2013.

Harold J. Kushner and Dean S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag, 1978.

Harold J. Kushner and G. Yin. Asymptotic properties of distributed and communicating stochastic approximation algorithms. *SIAM Journal on Control and Optimization*, 25(5):1266–1290, 1987.

Harold J. Kushner and G. Yin. Stochastic approximation and recursive algorithms and applications. *Springer-Verlag*, 2003.

Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer Texts in Statistics. Springer, 1998.

Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J. Smola. Efficient mini-batch training for stochastic optimization. In *Knowledge Discovery and Data Mining (KDD)*, 2014.

Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. A universal catalyst for first-order optimization. In *Neural Information Processing Systems (NIPS)*, 2015.

Gideon Mann, Ryan T. McDonald, Mehryar Mohri, Nathan Silberman, and Dan Walker. Efficient large-scale distributed training of conditional maximum entropy models. In *Neural Information Processing Systems (NIPS) 22*, 2009.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. *Mathematical Programming*, volume 155, 2016.

Arkadi S. Nemirovsky and David B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley, 1983.

Yurii E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Doklady AN SSSR*, 269, 1983.

Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Neural Information Processing Systems (NIPS) 24*, 2011.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4, 1964.

Boris T. Polyak and Anatoli B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J Control Optim*, volume 30, 1992.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Stat.*, vol. 22, 1951.

Jonathan Rosenblatt and Boaz Nadler. On the optimality of averaging in distributed statistical learning. *CoRR*, abs/1407.2724, 2014.

Nicolas Le Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In *Neural Information Processing Systems (NIPS) 25*, 2012.

David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. *Tech. Report, ORIE, Cornell University*, 1988.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *CoRR*, abs/1209.1873, 2012.

Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Neural Information Processing Systems (NIPS) 26*, 2013a.

Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *Neural Information Processing Systems (NIPS) 26*, 2013b.

Samuel L Smith, Pieter-Jan Kindermans, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

Martin Takác, Avleen Singh Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for SVMs. In *International Conference on Machine Learning (ICML)*, volume 28, 2013.

Martin Takác, Peter Richtárik, and Nati Srebro. Distributed mini-batch sdca. *CoRR*, abs/1507.08322, 2015.

Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge University Publishers, 2000.

Yuchen Zhang and Lin Xiao. Disco: Distributed optimization for self-concordant empirical loss. In *International Conference on Machine Learning (ICML)*, 2015.

Yuchen Zhang, John C. Duchi, and Martin Wainwright. Divide and conquer ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research (JMLR)*, volume 16, 2015.

Martin A. Zinkevich, Alex Smola, Markus Weimer, and Lihong Li. Parallelized stochastic gradient descent. In *Neural Information Processing Systems (NIPS) 24*, 2011.