Mini-Batching Theory: Matrix Product Concentration Methods

by

Mael Macuglia

A thesis presented for the degree of
Master of Computational Science

Supervisors: Petar Nizić-Nikolac, Prof. Afonso S. Bandeira

Department of Mathematics
Institute for Operations Research
ETH Zurich
Zurich
20.08.24

Abstract

In modern data science and machine learning, particularly for large-scale problems, optimization plays a crucial role. Mini-batching has emerged as a practical approach, yet its theoretical underpinnings remain incompletely understood. This work aims to bridge this gap by providing rigorous theoretical analysis of mini-batch stochastic gradient descent (SGD) methods. We prove that mini-batch SGD, applied to consistent least squares problems, converges at the same rate as its deterministic counterpart, given a sufficiently large batch size. Our analysis accommodates versatile sampling procedures, encompassing both standard SGD with uniform sampling and averaging Kaczmarz methods. We provide both expectation and high-probability bounds, leveraging novel concentration results for products of matrices—a departure from traditional optimization proof techniques. Building on Bollapragada et al.'s work [4], which showed expected convergence for mini-batch SGD with heavy ball momentum, we extend their results to provide high-probability bounds under the same conditions. Additionally, we present expectation bounds for μ-strongly convex and L-smooth functions that closely approximate quadratics, demonstrating that mini-batch SGD in the interpolation regime converges similarly to gradient descent, given an adequately large batch size. Our results not only advance the theoretical understanding of mini-batching but also offer practical insights for algorithm selection and tuning in large-scale optimization scenarios. By bridging the gap between stochastic and deterministic methods, this work contributes to the foundation of efficient, scalable optimization techniques for modern machine learning applications.

Acknowledgements

I would like to thank my supervisor Petar Nizić-Nikolac for his support during my thesis work. My sincere thanks to Prof. Afonso Bandeira for the opportunity to conduct my master′s thesis research under his mentorship. Special thanks go to my friend Deepak Narayanan for the many insightful discussions we shared. I would also like to extend my appreciation to Ilyas Fatkhullin for sharing his knowledge from the optimization community.

CONTENTS

CHAPTER 1

INTRODUCTION

In recent years, the field of numerical linear algebra and optimization has witnessed significant advancements in the analysis of iterative methods. Particularly noteworthy is the development of new tools for non-asymptotic growth and concentration bounds for products of independent random matrices. These tools mirror the spirit of results previously established for sums of independent random matrices [41], yet they open up new avenues for analysis in contemporary applications. One such application where products of random matrices arise naturally is in Kaczmarz iterative methods [40]. Recent work by Huang et al. [22] has provided novel tools specifically tailored for analyzing products of random matrices in the context of iterative methods. A key strength of these new techniques is their applicability to matrices without specific structural assumptions, instead relying on conditions similar to those found in Bernstein-type inequalities, such as bounded first and second moments. This thesis aims to take a fundamentally different approach from the existing literature and common proof techniques for iterative algorithms. Our primary objective is to investigate whether these new tools for products of random matrices can yield novel results and bridge gaps in the current body of knowledge. By leveraging these advanced analytical techniques, we seek to deepen our understanding of iterative methods and potentially uncover new insights that have eluded traditional approaches. Furthermore, our methods have yielded novel applications and results in mini-batching theory, expanding the scope and impact of our research beyond initial expectations.

To appreciate the novelty of our approach, it's important to understand the standard method for deriving convergence of stochastic first-order schemes [34, 15, 21, 37, 40, 6]. These schemes generally follow an iterative update rule of the form $x_{k+1} = \Phi\big[x, f(x), \nabla f(x)\big]$, where $f(x)$ is the objective function to minimise or maximise. This can be summarised in three key steps:

Step 1: Take the expectation of the square error conditioned on the history of iterates:

$$\mathbb{E}_{\underline{\boldsymbol{x}}_k|x_{k-1},\dots,x_1}\|\underline{\boldsymbol{x}}_{k+1}-x^*\|^2 = \mathbb{E}_{\underline{\boldsymbol{x}}_k|x_{k-1},\dots,x_1}\|\Phi\big[\underline{\boldsymbol{x}}_k, f(\underline{\boldsymbol{x}}_k), \nabla f(\underline{\boldsymbol{x}}_k)\big]-x^*\|^2 \tag{1.1}$$

Step 2: Use the geometry of the L_2 inner product together with properties of the objective function f (strong convexity, smoothness, etc.) to upper bound the conditional expectation:

$$\mathbb{E}_{\underline{\boldsymbol{x}}_k|x_{k-1},\dots,x_1}\|\Phi\big[\underline{\boldsymbol{x}}_k, f(\underline{\boldsymbol{x}}_k), \nabla f(\underline{\boldsymbol{x}}_k)\big]-x^*\|^2 \leq L\cdot\|x_{k-1}-x^*\|^2+\text{Noise} \tag{1.2}$$

where L corresponds to the linear convergence rate.

Step 3: Use steps 1 and 2 recursively together with the law of total expectation to obtain a bound of the form:

$$\mathbb{E}\|\underline{\boldsymbol{x}}_{k+1}-x^*\|^2 \leq L^k\|x_0-x^*\|^2+\widetilde{\text{Noise}} \tag{1.3}$$

Following the approach proposed by Bollapragada et al. [4], we restrict our analysis to the class of linear iterative operators satisfying:

$$\Phi\big[\underline{\boldsymbol{x}}_k, f(\underline{\boldsymbol{x}}_k), \nabla f(\underline{\boldsymbol{x}}_k)\big] = \underline{\boldsymbol{T}}_k\cdot\underline{\boldsymbol{x}}_k \quad \text{s.t.} \quad \Phi\big[x^*, f(x^*), \nabla f(x^*)\big]=0 \quad , \quad x^* = \text{Minimizer of } f \tag{1.4}$$

Here, $\underline{\boldsymbol{T}}_k$ is a random matrix. This formulation, which explicitly involves products of random matrices, serves as the primary motivation for studying linear iterative operators through this lens. Our analysis of these iterative operators, grounded in random matrix theory, follows these key steps:

Step 1: Unfold the recursion of the error norm and upper bound it by the sub-multiplicative property of norms:

$$\|\underline{\boldsymbol{x}}_k-x^*\| = \|\underline{\boldsymbol{T}}_k\cdots\underline{\boldsymbol{T}}_1\cdot(x_0-x^*)\| \leq \|\underline{\boldsymbol{T}}_k\cdots\underline{\boldsymbol{T}}_1\|\cdot\|x_0-x^*\| \tag{1.5}$$

Step 2: Control the expectation or the tail (with probability δ) of the spectral norm of the product of independent random matrices:

$$\mathbb{E}\|\underline{\boldsymbol{T}}_k\cdots\underline{\boldsymbol{T}}_1\| \leq \text{Constant}\cdot\big(\,\cdot\,\big)^k \quad \text{or} \quad \mathbb{P}\Big\{\|\underline{\boldsymbol{T}}_k\cdots\underline{\boldsymbol{T}}_1\| \leq \text{Constant}\cdot\big(\,\cdot\,\big)^k\Big\} \leq 1-\delta \tag{1.6}$$

Several well-known algorithms belong to the family of first-order iterative methods satisfying equation (1.4), especially in the context of solving consistent least squares problem

$$\big\{f(x)=\|Ax-b\|^2; \quad A\in\mathbb{R}^{n\times d}, b\in Range(A)\big\} \tag{1.7}$$

These include :

- Randomized Kaczmarz (RK) methods [40] and their accelerated modifications (Accelerated Randomized Kaczmarz (ARK)) [28, 33, 47], along with numerous related techniques, which are comprehensively summarised in the survey by [14].
- Stochastic Gradient Descent (SGD) with importance sampling, which has a direct link to the RK algorithm [33].
- Adding Heavy Ball Momentum (HBM) to the SGD and RK.
- Batching methods, a common practice in modern optimization to improve speed while managing per-iteration costs. In the context of SGD or RK, we refer to these as mini-SGD, and when HBM is added, mini-HBM. For RK algorithms, mini-SGD (and mini-HBM) correspond to RK with block averaging (plus momentum) [17, 15, 37].

This thesis partially builds upon the work of Bollapragada et al. [4], which demonstrated that, in expectation, mini-HBM converges as fast as its deterministic version in terms of the condition number of A^TA denoted κ. Their results show that for achieving ϵ-error iterates, we need at least $\mathcal{O}\left(\sqrt{\kappa}\log(1/\epsilon)\right)$ iterations, provided the batch size is large enough. This rate is known to be the informationally theoretically optimal [34]. We extend their work by providing a stronger version of their result, showing that we can attain the optimal rate not only in expectation but with high probability. This implies that our bounds can be used for confidence intervals. We also provide results for mini-SGD, both in expectation and with high probability for completeness. These results indicate that mini-SGD requires a smaller batch size than mini-HBM to converge at a rate close to its deterministic version (GD). Inspired by the importance of optimization in modern data science and machine learning, we also demonstrate that mini-SGD converges as fast as GD on μ-strongly convex and L-smooth families of objective functions for finite sum problems ($f(x) = \sum_i f_i(x)$) under interpolation regime, given the batch is large enough.

It's worth noting that in the context of Empirical Risk Minisation (ERM), several well-known learning paradigms, such as logistic regression, yield optimization objectives that fall into this category.

1.1 Contributions

1.1.1 Consistent Linear System of Equations

In this section, we consider the consistent least squares problem with system matrix $A \in \mathbb{R}^{n\times d}$, formally defined in problem 2.2.3. We focus on the setting where the number of rows is much larger than the number of columns, i.e., $n \gg d$. For all results in Table 1.1.1 we make the following assumption (which is stated again in chapter 3 Assumption: 3.1.1 for completeness). The mini-batch stochastic gradient descent (mini-SGD) is defined in Definition 2.3.1, and the mini-batch heavy ball momentum (mini-HBM) is defined in Definition 2.3.2.

We assume that for some $\eta \geq 1$ the sampling probabilities p_j from mini-SGD satisfies

$$\eta p_j \geq \frac{\|a_j\|^2}{\|A\|_F^2} \quad \forall j \in [n] \tag{1.8}$$

Convergence Type	Algorithm	Iterations	#Rows Prod/Iter	References
-	HBM	$\sqrt{\kappa}$	n	[27, 21, 37]
$\mathbb{E}$	mini-HBM	$\sqrt{\kappa}$	$B \gtrsim d \cdot \log(d) \cdot \overline{\kappa} \cdot \sqrt{\kappa}$	[4]
$\mathbb{P}$	mini-HBM	$\sqrt{\kappa}$	$B \gtrsim d \cdot \log(d/\tilde{\delta}) \cdot \overline{\kappa} \cdot \sqrt{\kappa}$	(cor 3.4.1)
-	GD	κ	n	[15, 34]
$\mathbb{E}$	mini-SGD	κ	$B \gtrsim d \cdot \log(d) \cdot \overline{\kappa}$	(cor 3.1.2)
$\mathbb{P}$	mini-SGD	κ	$B \gtrsim d \cdot \log(d/\tilde{\delta}) \cdot \overline{\kappa}$	(cor 3.2.1)
$\mathbb{E}$	SGD/RK	$d \cdot \overline{\kappa}$	1	[33, 40]
$\mathbb{E}$	ARK	$d \cdot \sqrt{\overline{\kappa}}$	1	[28]

Table 1.1: *Runtime comparisons for mini-batch versions of Kaczmarz method / stochastic gradient descent with weighted sampling, both with and without heavy ball momentum, applied to consistent linear systems. Here, κ corresponds to the condition number with respect to the 2-norm of the matrix $A^T A$, and $\overline{\kappa}$ represents the corresponding smoothed condition number, where A is an $n \times d$ matrix.The results that hold in expectation with respect to the stochasticity of the algorithm are denoted by the symbol $\mathbb{E}$, whereas stronger versions of the results (i.e., those that hold with high probability) are denoted by the symbol $\mathbb{P}$.*

Comparison and Literature Review of the Results

For mini-SGD with uniform probabilities, which is the most commonly used setting for solving machine learning algorithms, our assumption covers this case with $p_j := 1/n$ and $\eta = n \cdot \max_j \|a_j\|^2 / \|A\|_F^2$. In the case where $\eta = 1$, implying $p_j = \|a_j\|^2/\|A\|_F^2$, the mini-SGD recovers the average block Kaczmarz method [31]. To compare, the RK algorithm can be reformulated as the cyclic RK [23], which is shown to converge in $d \cdot \overline{\kappa}$ iterations. Note that $\overline{\kappa} = \frac{\lambda_{avg}}{\lambda_{min}}$ corresponds to the *smoothed* condition number of $A^T A$, with $\lambda_{avg} = \frac{1}{d} \sum_i \lambda_i$ and λ_{min} being the smallest eigenvalue. We draw the following important relation between the smoothed and standard condition numbers:

$$\overline{\kappa} \leq \kappa \leq d \cdot \overline{\kappa} \tag{1.9}$$

Hence, our results show that mini-SGD (or average block RK) outperforms RK at the cost of more row iterations. From a practical point of view, our results demonstrate a significant improvement over RK if parallelization is allowed [8]. Moreover, we also provide a convergence analysis with high probability, which can be used for establishing confidence intervals, implementing early stopping, managing resources efficiently, and providing stronger guarantees. To the best of our knowledge, this is the first high probability bound for average block RK with guarantees.

The work of Ma et al. [30] demonstrates the existence of a critical batch size for mini-batch stochastic gradient descent with uniform sampling on quadratics. For batch sizes exceeding this critical value, the convergence rate approaches that of full-batch gradient descent. While their analysis focuses on the overparameterized setting ($d \gg n$), our results extend to importance sampling and primarily address the scenario where $n \gg d$. Additionally, we provide a high-probability bound, which is not present in their work.

For mini-HBM, several studies have contributed to our understanding of its convergence properties and performance: Loizou and Richárik [29] demonstrated a linear rate of convergence for the linear regression problem. However, the convergence rates they obtained are slower than the deterministic rate of HBM. Can et al. [7] provided various guarantees for mini-HBM. Their analysis, however, requires bounded variance of the stochastic gradient, which is not applicable in the context of Kaczmarz methods. Lee et al. [26] showed that mini-HBM converges in expectation at a rate equivalent to deterministic HBM for a specific batch size. Their result improves upon the work of Bollapragada et al. [4] by a factor of κ. However, their analysis relies on strong assumptions about the system matrix A, specifically orthogonal invariance. In contrast, our approach and that of Bollapragada et al. [4] do not impose such constraints on the system matrix. Bollapragada et al. [4] provided results in expectation convergence. Our work extends their findings by establishing a stronger result with high probability, as presented in result 1.1.1.

Combining our findings on mini-SGD (for both in expectation and high-probability cases) and mini-HBM (for high-probability cases) with the in expectation-case results from Bollapragada et al. [4], we observe that both methods yield the same total computational cost when considering the product of iterations and row operations. However, it's important to note that such theoretical analyses may not directly translate to practical applications, as the actual computational cost heavily depends on implementation-specific factors such as memory allocation and parallelization strategies. Nevertheless, Table 1.1.1, which summarizes our findings, provides insight into the trade-offs between mini-SGD and mini-HBM. These trade-offs primarily involve the number of iterations versus the number of row operations per iteration. From a parallelization perspective, using mini-HBM to achieve convergence comparable to HBM requires $\sqrt{\kappa}$ more threads than mini-SGD. This difference can be significant for large values of κ, highlighting the importance of method selection based on the specific computational resources available and the characteristics of the problem at hand.

1.1.2 $\mu-$Strongly Convex and $L-$Smooth on Finite Sum Problems with Interpolation

In this section, we transition from the quadratic setting to what we loosely define as "almost quadratic." This category encompasses finite sum problems in the interpolation regime that are μ-strongly convex and L-smooth. A formal definition is provided in problem 2.2.4. We assume that the iterates remain within a region where the objective function behaves approximately quadratically, which is a common assumption for analysis purposes [19, 30]. In this context, a key step in our analysis involves replacing the (stochastic) gradient in the mini-SGD algorithm with its second-order Taylor approximation (see section 3.2). Furthermore, we make specific assumptions about the probability distribution used in mini-SGD. For completeness, this assumption is restated in section 3.2 as assumption 3.3.1.

Assume for some $\eta > 1$ that the following condition holds for the probabilities p_i in mini-SGD

$$\frac{\|H_i\|}{p_i} \leq \eta\|H\| \quad \forall i \in [n] \tag{1.10}$$

where $H := \nabla^2 f(x^*)$ and $H_i := \nabla^2 f_i(x^*)$ are the respective Hessians of the sum of functions f and the individual summands f_i evaluated at the interpolation point x^*. Similar to the assumption made in the context of consistent least squares, we aim to allow for a more general form of sampling than the uniform distribution typically used in machine learning.

Convergence Type	Algorithm	Iterations	#Rows Prod/Iter	References
-	GD	κ	n	[17], [6], [34]
$\mathbb{E}$	mini-SGD	κ	$B \gtrsim \log(d)\kappa$	(cor 3.3.1)
$\mathbb{E}$	mini-SGD	$\tilde{\kappa}$	$B \in [1, n]$	[15], [18]
$\mathbb{E}$	SGD	κ_{max}	1	[34], [17]

Table 1.2: *Runtime comparisons for stochastic, mini-batch, and plain gradient descent applied to finite sum problems in the interpolation regime, for which the underlying function is μ-strongly convex and L-smooth. Here, $\kappa := \frac{L}{\mu}$, whereas $\kappa_{\max} := \frac{\max_i L_i}{\mu}$. The term $\tilde{\kappa}$ interpolates between κ and $\kappa_{\max}$. The results hold in expectation with respect to the stochasticity of the algorithm, encoded by the symbol $\mathbb{E}$.*

Comparison and Literature Review of the Results

Ma et al. [30] identify a saturation point for batch size, beyond which mini-SGD with uniform sampling converges approximately at the same rate as its deterministic counterpart on μ-strongly convex and L-smooth functions for finite sum problems under interpolation, in the overparameterized regime $d \gg n$. Our results focus on the regime $n \gg d$. While their regime is highly relevant from a modern machine learning perspective, where overparameterized models predominate in practice, our results could potentially be extended from interpolation to noisy settings, as demonstrated for the quadratic case by Bollapragada et al. [4]. This represents an initial attempt to explore the value of the proof technique beyond the context of linear systems. The mini-SGD algorithm on μ-strongly convex and L-smooth settings has been extensively studied. We refer to the results of a recent survey by Garrigos et al. [15]. Their findings demonstrate a linear convergence of mini-SGD on $\mathcal{F}_{\mu,L}$ of the order:

$$\tilde{\kappa} = \left(\frac{n(B-1)}{B(n-1)}\right)\kappa + \left(\frac{n-B}{B(n-1)}\right)\kappa_{max} \tag{1.11}$$

where $\kappa = L/\mu$ represents the ratio of smoothness to strong convexity parameters of the sum of functions, while κ_{max} corresponds to $\max_i L_i/\mu$, where L_i is the smoothness parameter of the summand function f_i in the finite sum problem (2.2.2). For $B = 1$, the convergence rate is of order κ_{max}, whereas for full batch, it is of order κ. Note that κ_{max} can be significantly larger than κ. Our results indicate that beyond a certain batch size threshold, the rate is of order κ. We illustrate this comparison in figure 1.1.

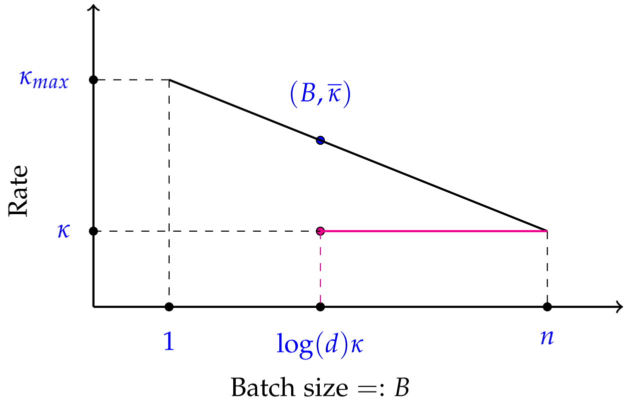


Figure 1.1: *Illustration of the result presented in Table 1.2.*

Remark 1.1.1 It is important to note that, for the sake of our analysis, we approximate the gradient $\nabla f_i(x)$ for all $i \in [n]$ by its second-order Taylor approximation at the interpolation point x^*. The error generated from this approximation is not accounted for in the error bounds we present.

1.2 Notation

Notation: We use the following conventions:

- Random variables (scalar) are denoted by bold underlined lowercase letters, e.g., $\underline{\boldsymbol{x}}$
- Random matrices of size $n \times d$ are denoted by bold underlined uppercase letters, e.g., $\underline{\boldsymbol{X}}$
- Variables with an asterisk correspond to minimizers of the optimization problem, e.g., x^*
- We use the ℓ_2-norm $\|\cdot\|$ for vectors and the spectral norm $\|\cdot\|$ for matrices
- The Frobenius norm is denoted by $\|\cdot\|_F$
- Throughout, $A \in \mathbb{R}^{n \times d}$ is assumed to be full rank
- The row rank of the matrix A is denoted $R(A)$

For matrix $A^T A$:

- $\kappa(A^T A)$ denotes the condition number with respect to the 2-norm
- $\lambda(A^T A)_i$ represents the eigenvalues
- $\lambda_{\max}(A^T A)$ and $\lambda_{\min}(A^T A)$ denote the largest and smallest eigenvalues, respectively

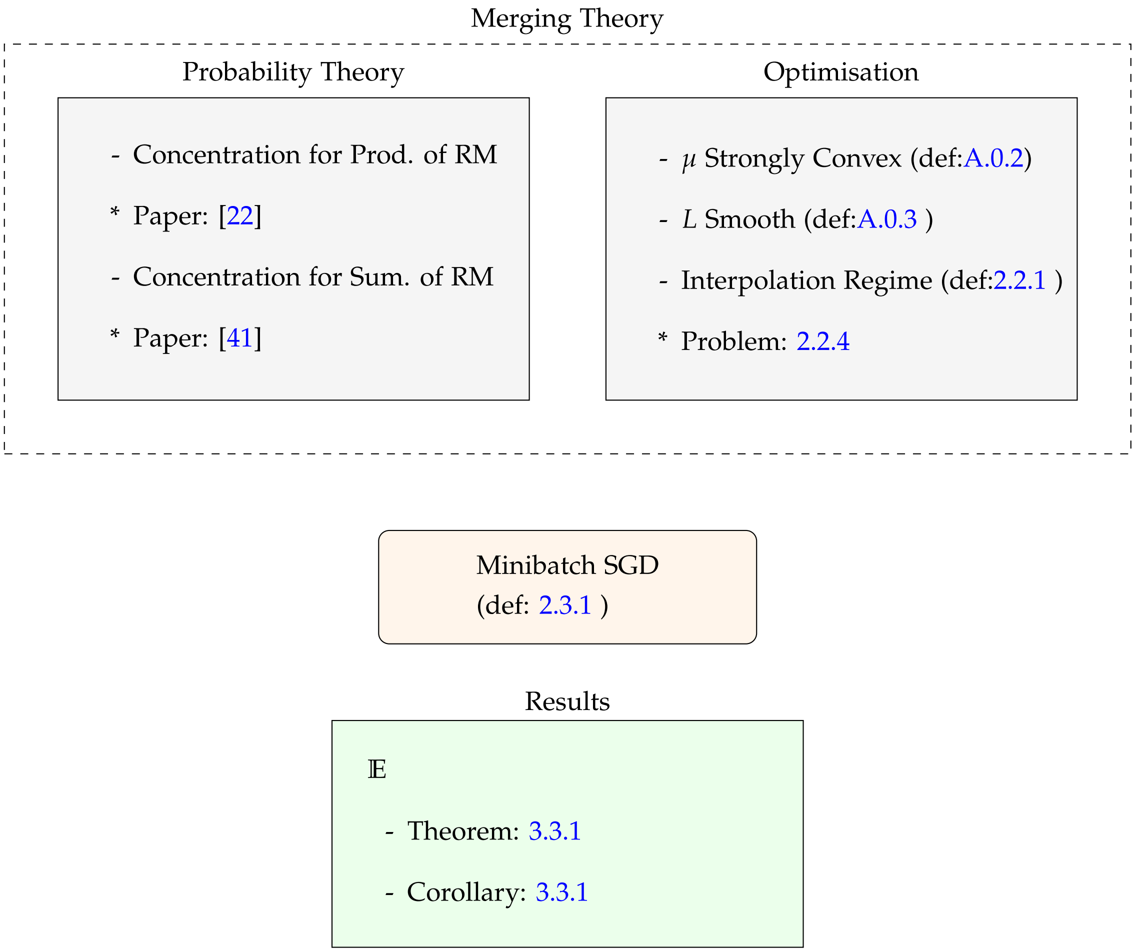
For matrix A:

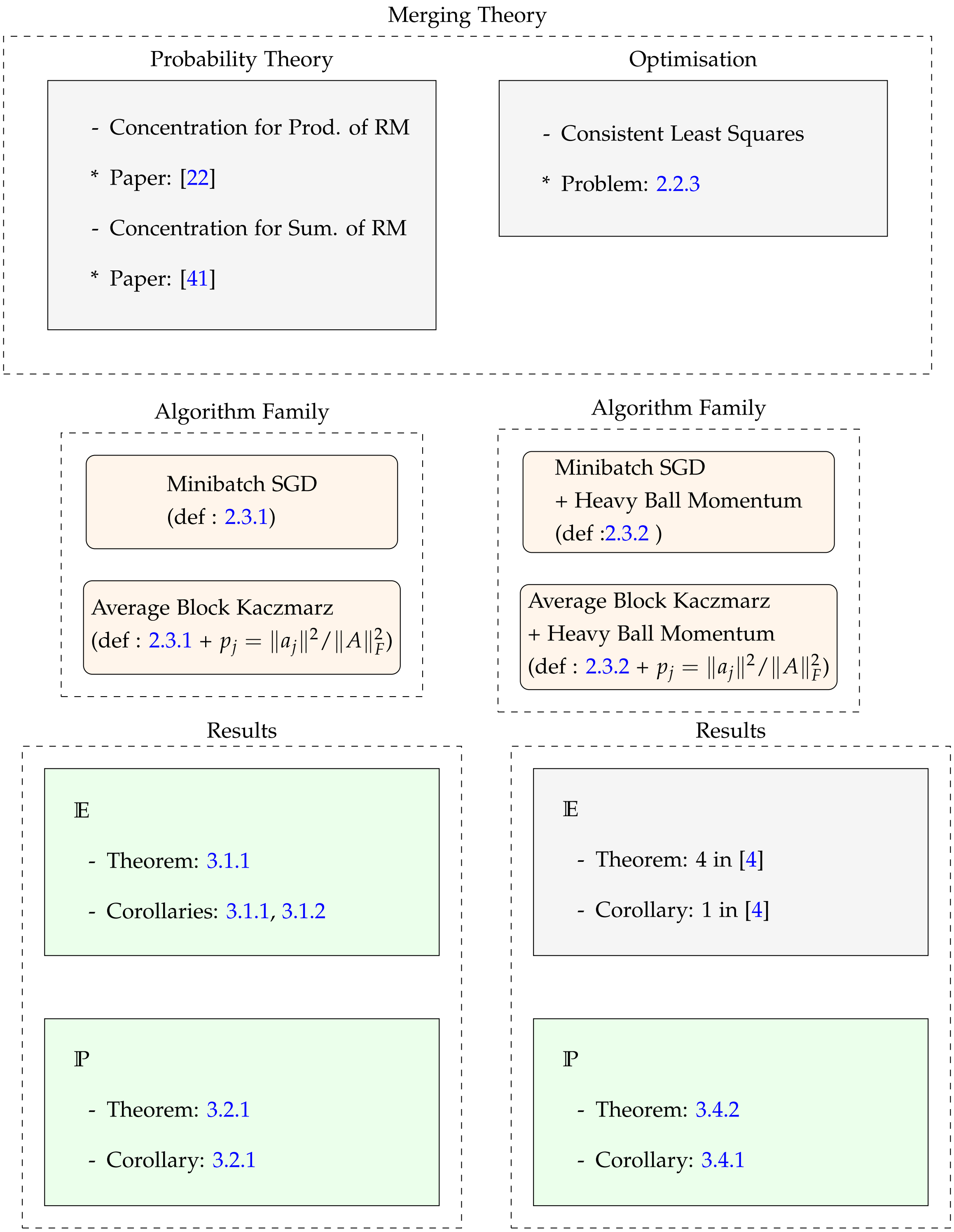
- $\sigma(A)_i$ represents the singular values, ordered as $\sigma_1(A) \geq \cdots \geq \sigma_d(A)$

The smoothed condition number is defined as $\overline{\kappa} = \lambda_{\text{avg}} / \lambda_{\min}$, where $\lambda_{\text{avg}} = \frac{1}{d} \sum_{i=1}^{d} \lambda_i$.

1.3 Thesis Mindmap Overview

In this section, we present a simplified overview of the work conducted in this thesis through the lens of two flow charts: one for solving consistent linear least squares problems, and another for optimising μ-strongly convex and L-smooth functions under the interpolation regime. These flow charts provide visual links to different parts of the thesis. Our novel contributions are highlighted in light green. The symbol $\mathbb{E}$ encodes that the results of the iterates error norm are in expectation, whereas the symbol $\mathbb{P}$ encodes that the results are with high probability.




Merging Theory
Probability Theory
- Concentration for Prod. of RM
* Paper: [22]
- Concentration for Sum. of RM
* Paper: [41]
Optimisation
- Consistent Least Squares
* Problem: 2.2.3
Algorithm Family
Minibatch SGD
(def : 2.3.1)
Average Block Kaczmarz
(def : 2.3.1 + $p_j = \|a_j\|^2 / \|A\|_F^2$)
Algorithm Family
Minibatch SGD
+ Heavy Ball Momentum
(def :2.3.2)
Average Block Kaczmarz
+ Heavy Ball Momentum
(def : 2.3.2 + $p_j = \|a_j\|^2 / \|A\|_F^2$)
Results
$\mathbb{E}$
- Theorem: 3.1.1
- Corollaries: 3.1.1, 3.1.2
$\mathbb{P}$
- Theorem: 3.2.1
- Corollary: 3.2.1
Results
$\mathbb{E}$
- Theorem: 4 in [4]
- Corollary: 1 in [4]
$\mathbb{P}$
- Theorem: 3.4.2
- Corollary: 3.4.1

CHAPTER

2

PRELIMINARIES

2.1 Concentration of Measure: A Cornerstone of Modern Probability

This chapter aims to provide the reader with a "minimal overview" of the background in concentration theory necessary to understand the novelty of the tools borrowed from recent advancements in probability theory. It may also help elucidate the timeline of concentration theory, which motivates our retrospective approach in analyzing iterative methods, as some relevant results were not available earlier. We begin by presenting what we consider to be common graduate-level knowledge in data science regarding concentration in scalar and sum cases. We then outline some lines of work extending these concepts to matrix and product cases.

2.1.1 Concentration of Measure: Background

A fundamental question in probability theory concerns how closely a functional over a set of independent random variables $\{\underline{x}_1, \ldots, \underline{x}_n\}$ approximates its expectation:

$$f(\underline{x}_1, \ldots, \underline{x}_n) \approx \mathbb{E}[f(\underline{x}_1, \ldots, \underline{x}_n)] \tag{2.1}$$

This question has been extensively studied in scalar form for various functional forms f under the name "concentration of measure phenomena" [25].

In this thesis, we employ tools to control functionals in the form of products and sums of independent random matrices. Controlling sums of independent random variables is known as Bernstein inequalities. For completeness, we recall what we mean by control in that case:

$$\mathbb{P}\left\{\left|\sum_i \underline{\boldsymbol{x}}_i\right| \geq t\right\} \leq \text{Something}(\text{var}(\sum_i \underline{\boldsymbol{x}}_i), t) \tag{2.2}$$

where we hope this "something" decays rapidly, typically exponentially (or sub-Gaussian [42]). To achieve such bounds for scalar random variables, the key step is applying the *Laplace* technique to the moment generating function. The moment generating function of the sum becomes a product of independent moment generating functions, which are then upper bounded using Markov's inequality:

$$\mathbb{E}[\exp(\lambda \sum_i \underline{\boldsymbol{x}}_i)] = \prod_i \mathbb{E}[\exp(\lambda \underline{\boldsymbol{x}}_i)] \quad \text{and} \quad \text{Markov's inequality} \tag{2.3}$$

Extending similar results to sums of random matrices has been of great interest for analyzing various data science problems such as covariance estimation, graphs, randomized linear algebra, analysis of kernel methods, and dimension reduction (see [42, 44]). The key challenge in extending equation 2.2 to matrices lies in their *non-commutativity*, as commutativity is required almost surely for equation 2.2 to hold. In a series of works [2, 35, 41], authors provided more technical results which can be summarized by the following idea:

$$\mathbb{E}[\text{Tr}(\exp\{\lambda \sum_i \underline{\boldsymbol{X}}_i\})] \leq \text{Tr}(\exp\{\sum_i \log \mathbb{E}[e^{\lambda \underline{\boldsymbol{X}}_i}]\}) \quad \text{and} \quad \text{Markov's inequality} \tag{2.4}$$

where the results incur a factor of d compared to the scalar case, with d representing the dimension of the matrix:

$$\mathbb{P}\left\{\left\|\sum_i \underline{\boldsymbol{X}}_i\right\| \geq t\right\} \leq d \cdot \text{Something}(\text{var}(\sum_i \underline{\boldsymbol{X}}_i), t) \tag{2.5}$$

The result we use concerning concentration for sums of independent random matrices (Theorems 2.1.1) are taken from the work of Tropp [41].

The case where the functional f is a product of random elements (matrices or scalars) also has applications in modern data science, such as control of dynamical systems and reinforcement learning theory [9, 10]. Classical work on controlling products of random matrices has primarily been conducted in the asymptotic regime: Bellman [3] studied the law of large numbers and central limit theorem for $n^{-1}\log(\underline{\boldsymbol{X}}_n \cdots \underline{\boldsymbol{X}}_1)_{ij}$. The field of *Free Probability*, mainly developed by Dan Voiculescu in the late 1980s [43, 39], also studies products of random matrices, but again in the regime where the dimension d approaches infinity.

For non-asymptotic results, Henriksen and Ward [20] provided findings for products of random matrices with minimal requirements on matrix structure. Their results are based on a clever decomposition of products into sums of independent random matrices. The techniques used for this decomposition are derived from combinatorics theory. They then apply Bernstein inequalities to bound each sum in the product decomposition. The results we use in this thesis (Theorems 2.1.2 and 2.1.3) are taken from the work of Huang et al. [22]. Huang et al. improve upon Henriksen's work by providing tighter results through the exploitation of geometric properties of the Banach space $(\mathbb{R}^{d\times d}, \|\cdot\|_p)$ known as *Uniform Smoothness*.

2.1.2 Concentration of Measure: Results

Theorem 2.1.1 Bernstein Like Inequality

Consider a finite sequence $\{\underline{\boldsymbol{W}}_k\}$ of independent random matrices with common dimension $d_1 \times d_2$. Assume that

$$\mathbb{E}\underline{\boldsymbol{W}}_i = 0 \quad \text{and} \quad \|\underline{\boldsymbol{W}}_i\| \leq W \tag{2.6}$$

and introduce the random matrix

$$\underline{\boldsymbol{Z}} = \underline{\boldsymbol{W}}_1 + \cdots + \underline{\boldsymbol{W}}_k \tag{2.7}$$

Let $\nu(\underline{\boldsymbol{Z}})$ be the matrix variance statistic of the sum:

$$\nu(\underline{\boldsymbol{Z}}) = \max\left\{ \| \sum_{i\in[k]} \mathbb{E}\underline{\boldsymbol{W}}_i \underline{\boldsymbol{W}}_i^T \|, \| \sum_{i\in[k]} \mathbb{E}\underline{\boldsymbol{W}}_i^T \underline{\boldsymbol{W}}_i \| \right\} \tag{2.8}$$

then,

$$\mathbb{E}\|\underline{\boldsymbol{Z}}\| \leq \sqrt{2\nu(\underline{\boldsymbol{Z}})\log(d_1+d_2)} + \frac{1}{3}W\log(d_1+d_2) \tag{2.9}$$

$$\sqrt{\mathbb{E}\|\underline{\boldsymbol{Z}}\|^2} \leq \sqrt{2e\nu(\underline{\boldsymbol{Z}})\log(d_1+d_2)} + 4eW\log(d_1+d_2) \tag{2.10}$$

Furthermore, $\forall t \geq 0$,

$$\mathbb{P}\left\{ \|\underline{\boldsymbol{Z}}\| \geq t \right\} \leq (d_1+d_2)\exp\left(\frac{-t^2/2}{\nu(\underline{\boldsymbol{Z}}) + Wt/3}\right) \tag{2.11}$$

Proof of Theorem 2.1.1.

The bound on $\mathbb{E}\|\underline{\boldsymbol{Z}}\|$ to Theorem 6.1.1 in [41] whereas the bound on $\sqrt{\mathbb{E}\|\underline{\boldsymbol{Z}}\|^2}$ corresponds to a matrix formulation of the Rosenthal-Pinelis inequality [36] (Thm: 4.1), [41] (eq: 6.1.6). This theorem also corresponds to the proposition 2 in [4] ☺ □

Theorem 2.1.2 Expectation Bound for Products of Random Matrices

Consider an independent sequence $\{\underline{\boldsymbol{Y}}_1, \ldots, \underline{\boldsymbol{Y}}_k\}$ of random matrices with dimension $d \times d$, and form the random product $\underline{\boldsymbol{Z}}_k = \underline{\boldsymbol{Y}}_1 \cdots \underline{\boldsymbol{Y}}_k$. Assume that

$$\|\mathbb{E}\underline{\boldsymbol{Y}}_i\| \leq m_i \quad \text{and} \quad \left(\mathbb{E}\|\underline{\boldsymbol{Y}}_i - \mathbb{E}\underline{\boldsymbol{Y}}_i\|^2\right)^{1/2} \leq \sigma_i m_i \quad \forall i \in [k]$$

Let $M := \prod_{i\in[k]} m_i$ and $\nu = \sum_{i\in[k]} \sigma_i^2$. Then

$$\mathbb{E}\|\underline{\boldsymbol{Z}}_k\| \leq \exp\left(\sqrt{2\nu\max\{2\nu, log(d)\}}\right) \cdot M \tag{2.12}$$

Proof of Theorem 2.1.2.

The theorem correspond to corollary 5.4 in [22] . ☺ □

Theorem 2.1.3 Tail Bound for the Spectral Norm of Product of Random Matrices

Consider an independent sequence $\{\underline{\boldsymbol{Y}}_1, \dots, \underline{\boldsymbol{Y}}_k\}$ of $d \times d$ random matrices, and form the product $\underline{\boldsymbol{Z}}_k = \underline{\boldsymbol{Y}}_k \cdots \underline{\boldsymbol{Y}}_1$. Assume that

$$\|\mathbb{E}\underline{\boldsymbol{Y}}_i\| \leq m_i \quad \text{and} \quad \|\underline{\boldsymbol{Y}}_i - \mathbb{E}\underline{\boldsymbol{Y}}_i\| \leq \sigma_i m_i \quad \text{almost surely} \quad \forall i \in [k]$$

Let $M := \prod_{i \in [k]} m_i$ and $\nu = \sum_{i \in [k]} \sigma_i^2$. Then

$$\mathbb{P}\{\|\underline{\boldsymbol{Z}}_k\| \geq tM\} \leq d \exp\left\{\frac{-\log^2(t)}{2\nu}\right\} \quad \text{when} \quad \log(t) \geq 2\nu \tag{2.13}$$

Proof of 2.1.3.

The Theorem corresponds to corollary 5.6 in [22] ☺ □

2.2 Optimisation: Theory

This thesis draws upon various concepts from optimization theory and numerical linear algebra. As terminology can vary across different fields, we will clarify these concepts and establish the nomenclature used throughout the remainder of this work. For formal definitions of μ-strong convexity and L-smoothness, as well as related theoretical results, please refer to Appendix A.0.1 and Appendix A.0.2.

2.2.1 Common Optimisation Problems

This section introduces a spectrum of optimization problems that will be instrumental in later chapters of this thesis. We begin by defining a broad class of optimization problems, characterized by minimal known structure on the objective function. Subsequently, we present more specific classes of objective functions, highlighting their distinctive properties and applications. We also provide an overview of the fields where these optimization problems are particularly relevant, demonstrating their wide-ranging applicability. Given the extensive literature on optimization, we have chosen to anchor our discussion primarily in the comprehensive survey by Garrigos et al. [15]. This recent work provides a unified framework for many optimization concepts, aligning closely with the problems we address in this thesis. We derive most of our definitions and terminology from this comprehensive review, which allows us to situate our contributions within the current state of the field.

Problem 2.2.1 Well Defined Unconstraint Optimisation

$$\min_{x \in \mathbb{R}^n} f(x)$$

for f continuous and assume the existence of a minimum value solution

$$\exists x^* \quad s.t \quad f(x^*) \leq f(x) \quad \forall x \in \mathbb{R}^n$$

Problem 2.2.2 Finite Sum Objective

Assuming a well defined problem 2.2.1, we wish to minimise a function $f : \mathbb{R}^d \to \mathbb{R}$ which can be written as

$$f(x) := \sum_{i \in [n]} f_i(x)$$

Remark 2.2.1 Finite sum objective as in 2.2.2 plays an important role in machine learning and statistics, particularly for ERM. Indeed, the goal of a learning setting is to find a measurable map $f : \mathcal{X} \to \mathcal{Y}$ given a set of observations $\mathbb{D}_n = \{x_i, y_i\}_{i \in [n]}$. Depending on the problem at hand, the function f is restricted to a function class $\mathcal{F}$. Taking a decision theory approach as in [24] the learning problem can be formulated as a risk minisation

$$\min_{f \in \mathcal{F}} R(f) := \mathbb{E}_{\mathbb{P}} l(f(\underline{x}), \underline{y}) \quad (\underline{x}, \underline{y}) \sim \mathbb{P} \tag{2.14}$$

where the loss $l : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ encodes a penalty which is user defined and problem dependent. The risk function in eq 2.14 is not computable as $\mathbb{P}$ is unknown. The ERM paradigm replaces the measure $\mathbb{P}$ by its empirical counter part $\hat{\mathbb{P}}_{\mathbb{D}}$ leading to an optimisation problem of the form 2.2.2, i.e

$$\min_{f \in \mathcal{F}} \hat{R}(f) := \frac{1}{n} \sum_{i \in [n]} l(f(x_i), y_i) \tag{2.15}$$

Often the function class $\mathcal{F}$ is assumed to be parametrised by some $\theta \in \Theta$ yielding

$$\min_{\theta \in \Theta} \hat{R}(\theta) = \frac{1}{n} \sum_{i \in [n]} l(f_\theta(x_i), y_i) \tag{2.16}$$

Definition 2.2.1 Interpolation

Consider the problem of the form 2.2.2. We say that interpolation holds if there exists a common $x^* \in \mathbb{R}^d$ such that $f_i(x^*) = \inf f_i \quad \forall i \in [n]$. Then, interpolation holds at x^*.

Remark 2.2.2 From the definition 2.2.1 it is easy to see that if interpolations holds i.e

$$f(x^*) = \sum_{i \in [n]} f_i(x^*) \leq \sum_{i \in [n]} \inf f_i \leq \sum_{i \in [n]} f_i(x) = f(x) \tag{2.17}$$

which implies

$$x^* \in \arg\min f(x) \tag{2.18}$$

Then, in a machine learning context (ERM) we see from eq 2.15 that interpolation means there exists a function $f^* \in \mathcal{F}$ that interpolates the data points $\mathbb{D}_n$

Problem 2.2.3 Least-Squares

For a given $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$ the least squares minisation problem is

$$x^* = \arg\min_{x \in \mathbb{R}^d} \frac{1}{2} \|Ax - b\|^2 \tag{2.19}$$

If $b \in R(A)$ we refer to it as **consistent** and we are in interpolation regime (def 2.2.1)
If $b \notin R(A)$ we refer to it as **inconsistent**
The least square problem is a finite sum problem 2.2.2 i.e

$$f(x) = \frac{1}{2}\|Ax - b\|^2 = \frac{1}{2} \sum_{i \in [n]} (\langle a_i, x \rangle - b_i)^2 \tag{2.20}$$

Lemma 2.2.1

The least squares objective in problem 2.2.3 is μ-stronly convex (def: A.0.2) and L-smooth (def: A.0.3) with parameters

$$\mu = \lambda_{min}(A^T A) \quad L = \lambda_{max}(A^T A)$$

Proof. 2.2.1 the function $f(x)$ is twice continously differentiable, hence we use the equivalence between L smoothness and lipschitz gradient:

$$\|\nabla f(x) - \nabla f(y)\| = \|A^T A(x - y)\| \leq \underbrace{\|A^T A\|}_{=:L} \|x - y\|$$

Using lemma A.0.2 $\exists g(x)$ s.t $g(x) = f(x) - \frac{\mu}{2}\|x\|^2$ is convex. By second order charachterisation of convexity this is equivalent to

$$0 \preceq \nabla^2 g(x) \Leftrightarrow \mu \mathbb{I}_{d \times d} \preceq \nabla^2 f(x)$$

By Courant-Fisher theorem:

$$\lambda_i(A^T A) \geq \mu \quad \forall i \in [d]$$
$$\mu := \lambda_{min}(A^T A) \implies \mu - \text{strong convexity}$$

☺ □

Problem 2.2.4 Finite-Sum: $\mu-$ Strongly Convex and $L-$ Smooth

Assume a finite sum problem as in 2.2.2 i.e

$$f(x) = \sum_{i \in [n]} f_i(x) \tag{2.21}$$

Moreover,

$$f_i \in \mathcal{F}_{\mu,L} := \{f_i : \mu_i - \text{Strongly Convex}, L_i - \text{Smooth}\} \tag{2.22}$$

Remark 2.2.3 Problem 2.2.4 plays an important role in machine learning. The least squares problem 2.2.3 in the ERM setting corresponds to a quadratic loss function, which is not suitable for classification tasks. Encoding a binary classification task as a type of regression problem yields, in the ERM setting, a loss function called "logistic loss" [13]. The logistic loss with $L2$ regularization is a μ-strongly convex and L-smooth function [6]. Other types of $L2$-regularized ERM problems have loss functions that belong to $\mathcal{F}_{\mu,L}$, such as the "hinge loss" for classification tasks modeled as support vector machines [6].

2.2.2 Gradient Descent

The origin of gradient descent lies in 19th-century mathematics, but its development and adoptation into the field of machine learning have made it a cornerstone of modern optimization techniques. Its popularity stems from its simplicity, effectiveness, scalability, and foundational role in training machine learning models, particularly in the context of deep learning. This combination of factors has cemented gradient descent as a go-to algorithm in both theoretical research and practical applications.

At the same time, we find that gradient descent has, in some ways, become a victim of its own success. The algorithm is so widely discussed and popularized that finding in-depth, high-quality information can be challenging—especially when searching through common online sources or on the internet. Many explanations have been oversimplified, often overlooking the algorithm's numerical analysis foundations. In response, we propose an approach that revisits gradient descent from a numerical analysis perspective—a viewpoint that we believe offers a deeper and more rigorous understanding, which is often neglected in the typical machine learning context. We build our approach inspired by the work in [38]. Our Approach: For a detailed derivation of gradient descent,

see Appendix A.0.3, *Understanding Gradient Descent*.

We now define the gradient descent iterative scheme:

Definition 2.2.2 Gradient Descent

The GD defines a sequence $\{x_k\}_{k \in \mathbb{N}}$ satisfying

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k)$$

where $\{\alpha_k\}_{k \in \mathbb{N}}$ is the set of so-called step sizes.

2.2.3 Momentum

The concept of momentum [def: 2.2.3] in optimisation algorithms, particularly in gradient-based methods, has its roots in the mid-20th century, where it was originally introduced to accelerate convergence rates in iterative algorithms. Over time, momentum has been adapted and refined within the field of machine learning, where it has become a fundamental tool in enhancing the performance of gradient descent and other optimisation techniques. Its widespread adoption is due to its ability to mitigate the oscillations often encountered in gradient descent, thereby providing faster and more stable convergence, particularly in the training of deep neural networks.

However, much like gradient descent, the success of momentum has led to an over-saturation of simplified explanations and resources, often diluting the rich, underlying theory. This oversimplification, particularly in easily accessible sources, can make it difficult to find comprehensive, high-quality information. Consequently, the deeper mathematical foundations of momentum, such as its origins in classical numerical analysis, are frequently overlooked. To address this, we present an approach that revisits momentum from a numerical analysis standpoint—a perspective that offers a more profound and rigorous understanding, often under-appreciated in the context of its application in machine learning. Our approach is inspired by the foundational work in [38, 45]. Our Approach: For a detailed derivation of momentum methods,

Refer to the manuscript titled *Understanding Momentum* in Appendix A.0.4.

We now define the general first order iterative scheme with momentum:

Definition 2.2.3 Momentum Iterative Scheme

Let $\{x_k\}_{k\in\mathbb{N}}$ be the sequence of iterates and $\{m_k\}_{k\in\mathbb{N}}$ be the sequence of momentum terms. The method is characterised by two sequences of user-defined parameters: learning rates $\{\alpha_k\}_{k\in\mathbb{N}}$ and momentum coefficients $\{\gamma_k\}_{k\in\mathbb{N}}$, where $\alpha_k, \gamma_k \in (0, \infty)$ for all $k \in \mathbb{N}$. For each iteration $k \in \mathbb{N}$, the update rules are given by:

$$m_{k+1} = \gamma_k m_k + (1-\gamma_k)\nabla f(x_k)$$
$$x_{k+1} = x_k - \alpha_k m_{k+1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function to be minimized, and $\nabla f(x_k)$ is its gradient at x_k.

Claim 2.2.1 Equivalence of General Scheme

An equivalent formulation of the general scheme is

$$x_{t+1} = x_t - \tilde{\alpha}_t \nabla f(x_t) + \tilde{\beta}_t(x_t - x_{t-1})$$

with

$$\tilde{\alpha}_t = \alpha_t(1-\gamma_t) \quad \tilde{\beta}_t = \frac{\alpha_t \gamma_t}{\alpha_{t-1}}$$

Proof of 2.2.1.
$\forall t \in \mathbb{N} > 0$ we have:

$$\begin{aligned}
m_t &= \frac{x_{t-1} - x_t}{\alpha_{t-1}} \\
\implies m_{t+1} &= \frac{\gamma_t}{\alpha_{t-1}}(x_{t-1} - x_t) + (1 - \gamma_t)\nabla f(x_t) \\
\Leftrightarrow \frac{x_t - x_{t+1}}{\alpha_t} &= \frac{\gamma_t}{\alpha_{t-1}}(x_{t-1} - x_t) + (1 - \gamma_t)\nabla f(x_t) \\
\Leftrightarrow x_{t+1} &= -\alpha_t(1 - \gamma_t)\nabla f(x_t) + \frac{\alpha_t}{\alpha_{t-1}}\gamma_t(x_t - x_{t-1})
\end{aligned}$$

☺ □

Definition 2.2.4 Heavy Ball Momentum

Let $\{x_k\}_{k\in\mathbb{N}}$ be the sequence of iterates. The HBM is characterized by two fixed user-defined parameters: a learning rate $\alpha > 0$ and a momentum coefficient $\beta \in [0, 1)$.
For each iteration $k \in \mathbb{N}$, the update rule is given by:

$$x_{k+1} = x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1})$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function to be minimized, and $\nabla f(x_k)$ is its gradient at x_k.

2.2.4 Theoretical Optimality of First-Order Methods on Quadratic Problems

Our findings on mini-SGD and mini-HBM on least squares problem 2.2.3, along with the results from Bollapragada et al. [4], are summarised in Table 1.1.1. These can be benchmarked against the known information-theoretic limitations for first-order iterative schemes on quadratic problems. A fundamental result in optimization theory, as presented in [34], establishes the information-theoretic lower bound for first-order methods. It states that under mild conditions, no first-order iterative method can converge faster than $\mathcal{O}(\sqrt{\kappa(Q)})$ iterations for ϵ-accuracy on quadratic objectives of the form $f(x) := x^TQx + b^Tx$. Notably, the HBM algorithm applied to the least squares problem 2.2.3 achieves this optimal convergence rate of $\mathcal{O}(\sqrt{\kappa(A^TA)})$, matching the information-theoretic lower bound.

2.3 Stochastic Optimisation

Recall that the problems we tackle in this thesis fit within the framework of finite sum problems (Equations 2.2.3 and 2.2.4), i.e.,

$$\min_{x\in\mathbb{R}^d} f(x) = \min_{x\in\mathbb{R}^d} \sum_{i\in[n]} f_i(x) \tag{2.23}$$

In the least squares problem (Equation 2.2.3), n represents the number of rows or data points in the ERM problem with loss function $\in F_{\mu,L}$ (see Remark 2.2.1). In large-scale settings (where n is large), the computational cost of evaluating the gradient $\nabla f(x)$ for first-order iterative methods becomes a significant computational bottleneck. To address this limitation, researchers drew inspiration from the field of stochastic optimization, developing stochastic algorithms that approximate the true gradient of f to reduce computational complexity. These methods, studied extensively in the 2010s, have played a crucial role in the success of machine learning over the past decade and are now widely used and well-known, with stochastic gradient descent being a prime example. Providing strong guarantees from an optimization perspective has further implications for the statistical problem at hand, as statisticians can rely on these optimization results. Interestingly, stochastic optimization, much like gradient descent, has become a victim of its own success in terms of the quality and quantity of information available. In this thesis, we draw particular inspiration from the work of Needell et al. [33] and Gower et al. [18] for our notation and approach.

Remark 2.3.1

It is worth noting that our notation and modeling approach differ from those used by Bollapragada et al. [4], whose work inspired our research. We find their notation for defining the stochastic problem in the context of finite-sum problems to be unclear or potentially problematic. Consequently, we propose our own derivation of notation, which we believe offers a more rigorous mathematical treatment—an aspect that is often lacking in the existing literature.

2.3.1 Finite-Sum Problem as Stochastic Optimisation Problem

The desire to reduce the computational cost of evaluating the gradient ∇f can be formulated as evaluating a stochastic approximation of this gradient, denoted $\nabla f(x, \underline{\boldsymbol{\xi}})$. To ensure consistency in the approximation, a classical approach is to require the stochastic gradient $\nabla f(x, \underline{\boldsymbol{\xi}})$ to be an unbiased estimator of the *true* gradient $\nabla f(x)$. Recall the definition of a stochastic optimization problem:

$$\min_{x\in\mathbb{R}^d} \mathbb{E} f(x, \underline{\boldsymbol{\xi}}) \quad , \quad \underline{\boldsymbol{\xi}} \sim \mathcal{D} \quad , \quad \operatorname{supp}(\xi) = \Omega \subseteq \mathbb{R}^d \tag{2.24}$$

To cast finite sum problems into stochastic optimization problems, we can write:

$$f(x) = \sum_{i\in[n]} f_i(x) = \sum_{i\in[n]} \underbrace{\mathbb{E}[\underline{\boldsymbol{\xi}}_i]}_{=1} \cdot f_i(x) = \mathbb{E}[\sum_{i\in[n]} f_i(x) \cdot \underline{\boldsymbol{\xi}}_i] =: \mathbb{E}[f(x, \underline{\boldsymbol{\xi}})] \tag{2.25}$$

where the random variables $\{\underline{\boldsymbol{\xi}}_i\}_{i\in[n]}$ fulfill the condition $\mathbb{E}[\underline{\boldsymbol{\xi}}_i] = 1$. Notice that unbiasedness of the gradient directly follows (under measurability assumptions):

$$\mathbb{E}[\nabla f(x, \underline{\boldsymbol{\xi}})] = \int_\Omega \nabla f(x, \underline{\boldsymbol{\xi}}) d\mathbb{P}_{\mathcal{D}} = \nabla \int_\Omega f(x, \underline{\boldsymbol{\xi}}) d\mathbb{P}_{\mathcal{D}} = \nabla \mathbb{E} f(x, \underline{\boldsymbol{\xi}}) = \nabla \sum_{i\in[n]} \underbrace{\mathbb{E}[\underline{\boldsymbol{\xi}}_i]}_{=1} \cdot f_i(x) = \nabla f(x) \tag{2.26}$$

In our work, we consider so-called *sampling with replacement*. We define the number of data points we wish to use in the evaluation of the gradient as B, which simultaneously defines the computational complexity of evaluating the approximated gradient. In machine learning literature, this is known as the **batch size**. To construct such an approximation of the gradient, we define the random variable $\underline{\boldsymbol{s}}$ for the indices of the dataset:

$$\mathbb{P}(\underline{\boldsymbol{s}} = i) = p_i \quad , \quad i \in [n] \quad , \quad \sum_i p_i = 1 \tag{2.27}$$

We then sample B i.i.d. copies of $\underline{\boldsymbol{s}}$, i.e., $\{\underline{\boldsymbol{s}}_1, \ldots, \underline{\boldsymbol{s}}_B\}$. Define:

$$\underline{\boldsymbol{\xi}}_i := \frac{1}{B \cdot p_i} \sum_{k\in[B]} \mathbb{I}\{\underline{\boldsymbol{s}}_k = i\} \qquad \mathbb{I}\{\cdot\} = \text{indicator function} \tag{2.28}$$

where we notice that $\mathbb{E}[\underline{\boldsymbol{\xi}}_i] = 1 \quad \forall i \in [n]$. Hence, the following stochastic gradient approximation is unbiased:

$$\nabla f(x, \underline{\boldsymbol{\xi}}) = \sum_{i\in[n]} \nabla f_i(x) \cdot \left(\frac{1}{B \cdot p_i} \sum_{k\in[B]} \mathbb{I}\{\underline{\boldsymbol{s}}_k = i\} \right) = \frac{1}{B} \sum_{i\in[B]} \frac{1}{p_{\underline{\boldsymbol{s}}_i}} \nabla f_{\underline{\boldsymbol{s}}_i}(x) \tag{2.29}$$

We can replace the gradient in the GD algorithm [def: 2.2.2] with the approximated gradient $\nabla f(x, \underline{\boldsymbol{\xi}})$ from equation 2.3.1. This yields the algorithm we study in this manuscript, namely the mini-SGD algorithm:

Definition 2.3.1 mini-SGD

The mini-SGD defines a stochastic sequence $\{\underline{\boldsymbol{x}}_k\}_{k\in\mathbb{N}}$ satisfying

$$\underline{\boldsymbol{x}}_{k+1} = \underline{\boldsymbol{x}}_k - \alpha_k \cdot \frac{1}{B} \cdot \sum_{j\in[B]} \frac{1}{p_{\underline{\boldsymbol{s}}_j}} \cdot \nabla f_{\underline{\boldsymbol{s}}_j}(\underline{\boldsymbol{x}}_k) \qquad \{\underline{\boldsymbol{s}}_j\}_{j\in[B]} \sim_{iid} \underline{\boldsymbol{s}} \tag{2.30}$$

where $\underline{\boldsymbol{s}}$ is a discrete random variable defined on the set of indices $[n] = \{1, 2, ..., n\}$, with probability mass function:

$$\mathbb{P}(\underline{\boldsymbol{s}} = j) = p_j \quad \text{for } j \in [n] \tag{2.31}$$

where p_j represents the probability of selecting index j. This setting corresponds to drawing indices with replacement from the set $[n]$. Additionally, let $\{\alpha_k\}_{k\in\mathbb{N}}$ denote a sequence of user-defined parameters.

Similarly, we can replace the gradient in the HBM algorithm [def: 2.2.4] with the stochastic approximation $\nabla f(x, \underline{\boldsymbol{\xi}})$ from equation 2.3.1. This modification yields the second algorithm we study in this manuscript, which is also described by Bollapragada et al. [4]. We refer to this algorithm as the mini-HBM algorithm:

Definition 2.3.2 mini-HBM

The mini-HBM defines a stochastic sequence $\{\underline{\boldsymbol{x}}_k\}_{k\in\mathbb{N}}$ satisfying

$$\underline{\boldsymbol{x}}_{k+1} = \underline{\boldsymbol{x}}_k - \alpha \frac{1}{B} \sum_{j\in[B]} \frac{1}{p_{\underline{\boldsymbol{s}}_j}} \nabla f_{\underline{\boldsymbol{s}}_j}(\underline{\boldsymbol{x}}_k) + \beta(\underline{\boldsymbol{x}}_k - \underline{\boldsymbol{x}}_{k-1}) \tag{2.32}$$

where $\underline{\boldsymbol{s}}$ is a discrete random variable as defined in mini-SGD definition (2.3.1). The set $\{\alpha, \beta\}$ is the set of user defined parameters.

2.4 Kaczmarz Methods

The mini-SGD [def: 2.3.1] and the mini-HBM [def: 2.3.2] have a direct connection to the so-called *average block Kaczmarz* method presented in [31] when solving the consistent least squares problem 2.2.3. The link is presented in the work by Needell et al. [33], where the authors establish the connection between SGD with weighted sampling and the Kaczmarz method. The assumption we make (Assumption 3.1.1) in our analysis casts into the randomized block average Kaczmarz. From the perspective of SGD with weighted sampling, the distribution over the index space or the rows of the system matrix A should be proportional to the squared norm of the selected rows. This yields our assumption:

$$\mathbb{P}\left(\underline{s} = j\right) = p_j := \frac{\|a_j\|^2}{\|A\|_F^2} \quad \Leftrightarrow \quad \eta = 1 \tag{2.33}$$

Hence, in this section, we provide a concise overview of the Kaczmarz methods and their applications, as well as a geometric interpretation of the Kaczmarz methods that are relevant to our application.

2.4.1 Historical Background and Applications of Kaczmarz Methods

Kaczmarz methods, named after Polish mathematician Stefan Kaczmarz, have a rich history dating back to the early 20th century. In 1937, Kaczmarz introduced an iterative algorithm for solving linear systems of equations [23]. This algorithm, now known as the Kaczmarz method or the algebraic reconstruction technique (ART), has found extensive applications in various fields. The first version of the Kaczmarz algorithm has the following form for solving a linear system $Ax = b$ with $A \in \mathbb{R}^{n\times d}$:

$$x_{k+1} = x_k + \alpha_i \frac{b_i - \langle a_i, x_k\rangle}{|a_i|^2} a_i^T \quad \text{with} \quad i = k \mod n \tag{2.34}$$

This version is commonly referred to as the *cyclic Kaczmarz* algorithm. In 2008, in their seminal work, Strohmer and Vershynin proposed the *randomized Kaczmarz* algorithm (RK) [40]. The main difference with respect to the standard Kaczmarz method lies in the selection process of the rows:

$$\underline{\boldsymbol{x}}_{k+1} = \underline{\boldsymbol{x}}_k + \alpha_i \frac{b_{\underline{i}} - \langle a_{\underline{i}}, \underline{\boldsymbol{x}}_k\rangle}{\|a_{\underline{i}}\|^2} a_{\underline{i}}^T \quad \text{with} \quad \mathbb{P}\left(\underline{\boldsymbol{i}} = i, i \in [n]\right) = \frac{\|a_i\|^2}{\|A\|_F^2} \tag{2.35}$$

The Kaczmarz iteration is a projection operator; hence, it has a nice geometric interpretation, i.e., it projects the iterates onto the affine space of the row equation. See Figure 2.1 for a simplified visualization (sketch) highlighting the difference between the cyclic and randomized Kaczmarz methods.

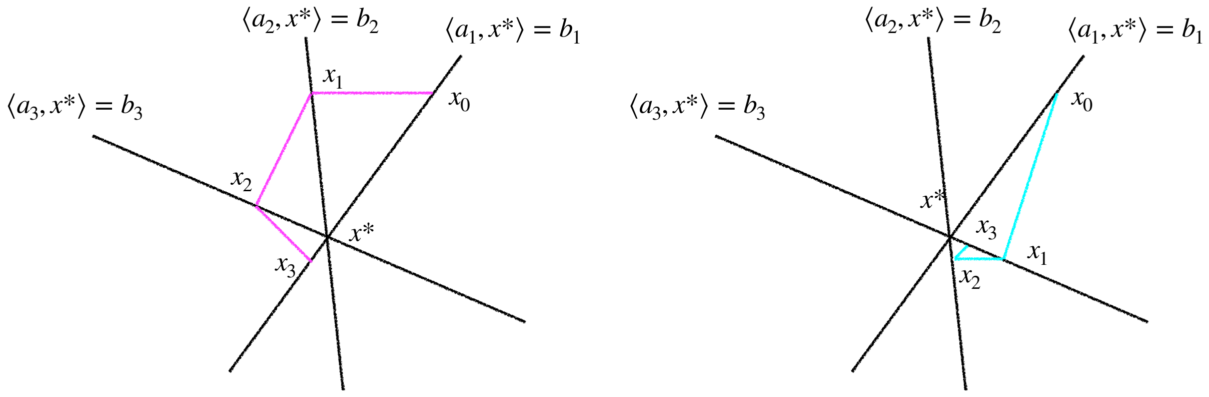

$\langle a_2, x^* \rangle = b_2$
$\langle a_1, x^* \rangle = b_1$
$\langle a_3, x^* \rangle = b_3$
x_1
x_0
x_2
x^*
x_3
$\langle a_2, x^* \rangle = b_2$
$\langle a_1, x^* \rangle = b_1$
$\langle a_3, x^* \rangle = b_3$
x_0
x^*
x_3
x_1
x_2

Figure 2.1: Geometric interpretation of the cyclic (left) and randomized (right) Kaczmarz method for solving a consistent linear system $Ax^* = b$ with $A \in \mathbb{R}^{n \times d}$ where $n = 3$

A huge body of literature has been built upon the cyclic and randomized Kaczmarz methods. The main takeaway is that for solving highly overdetermined linear systems, Kaczmarz methods can outperform reference methods such as the conjugate gradient iterative scheme. With the current evolution of large-scale problems, Kaczmarz methods have regained importance and are extensively studied in modern data science. We refer the reader to the following survey for an overview of the methods [14].

2.4.2 Average Block Kaczmarz Algorithm

Block Kaczmarz methods [1, 11, 12, 32, 46] are an extension of the classical Kaczmarz algorithm for solving large-scale linear systems of equations. While the original Kaczmarz method updates the solution estimate one row at a time, block Kaczmarz methods update the solution using multiple rows simultaneously. The way for choosing the block can be deterministic, as for the cyclic randomized Kaczmarz, or in a randomized fashion, called randomized block Kaczmarz. A common feature of block Kaczmarz algorithms is that for each iteration, the projection onto the space formed by the selected rows corresponds to solving a generalized least squares problem, i.e., computing the pseudo-inverse of the block matrix, which can be costly and difficult to parallelize.

$$x_{k+1} = x_k + A^{+}_{[\text{Block}]}(b_{[\text{Block}]} - A_{[\text{Block}]} x_k) \tag{2.36}$$

In cases where the selection of the rows follows a so-called *row paving* [32], the computation of the pseudo-inverse can be made efficient. In order to reduce the computational complexity and allow for parallelization while using more information than a single row of the system, the *average block Kaczmarz* [5, 33] algorithm can be used. For a geometric interpretation of the difference between the two methods see figure 2.2:

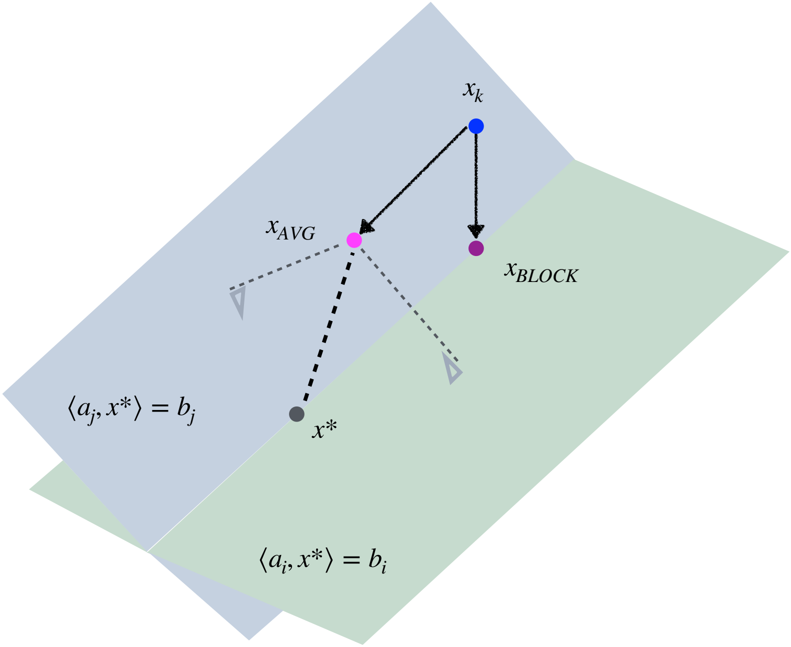

x_k
x_{AVG}
x_{BLOCK}
$\langle a_j, x^* \rangle = b_j$
x^*
$\langle a_i, x^* \rangle = b_i$

Figure 2.2: Sketch of the difference between *average* and *block* Kaczmarz. The linear system is consistent $Ax^* = b$. The new iterates are defined as x_{AVG} and x_{BLOCK} for the average and block Kaczmarz methods respectively.

Next, we provide the definition of average block Kaczmarz and its link to stochastic gradient descent with importance sampling, explicitly for *sampling with replacement*, as described in [31]:

$$\text{Importance Sampling [33]:} \quad \underline{\boldsymbol{x}}_{k+1} = \underline{\boldsymbol{x}}_k - \frac{1}{B}\sum_{i\in[B]} \frac{w_{\underline{\boldsymbol{s}}_i}}{L_{\underline{\boldsymbol{s}}_i}} \nabla f_{\underline{\boldsymbol{s}}_i}(\underline{\boldsymbol{x}}_k) \quad , \quad L_i = \|a_i\| \quad , \quad \nabla f_i(x) = (\langle a_i, x\rangle - b_i)a_i \tag{2.37}$$

$$\text{Avg. Block Randomized Kaczmarz [33, 31]:} \quad \underline{\boldsymbol{x}}_{k+1} = \underline{\boldsymbol{x}}_k - \frac{1}{B}\sum_{i\in[B]} w_{\underline{\boldsymbol{s}}_i} \cdot \frac{\langle a_{\underline{\boldsymbol{s}}_i}, \underline{\boldsymbol{x}}_k\rangle}{\|a_{\underline{\boldsymbol{s}}}\|} a_{\underline{\boldsymbol{s}}_i} \tag{2.38}$$

which cast into the mini-SGD framework with

$$\frac{1}{p_j} = \frac{w_i}{\|a_i\|^2} = \frac{\|A\|_F^2}{\|a_i\|^2} \quad \forall i \in [n] \tag{2.39}$$

Moreover, adding momentum to the average randomized Kaczmarz method makes it fall within the mini-HBM framework.

CHAPTER

3

NEW PROOF TECHNIQUES FOR OPTIMISATION VIA RM CONCENTRATION

3.1 Mini-Batch SGD Converges as GD on Quadratics

In this section we present the first contribution of the thesis where by taking a different approach than standard proof techniques in optimisation literature [15] we combine results for concentration of product of random matrices [22] and proof techniques used in [4] to provide a critical batch size needed for mini-SGD with fixed step size to converge with the same linear rate as GD with fixed step size. The optimisation problem corresponds to the **consistent** least squares problem 2.2.3 where the system matrix A is tall ($n > d$).

3.1.1 Convergence of GD

Here we propose a proof of convergence for the error norm of the iterates of gradient descent on consistent least square problem 2.2.3, i.e

$$\begin{aligned}
\|x_{k+1} - x^*\| &= \|x_k - \alpha \nabla f(x_k) - x^*\| && (3.1)\\
&= \|(\mathbb{I}_{d\times d} - \alpha A^T A)(x_k - x^*)\| && (3.2)\\
&= \|(\mathbb{I}_{d\times d} - \alpha A^T A)^k (x_0 - x^*)\| && (3.3)\\
&\leq \|(\mathbb{I}_{d\times d} - \alpha A^T A)^k\| \cdot \|x_0 - x^*\| && (3.4)\\
&\leq (\rho(\mathbb{I}_{d\times d} - \alpha A^T A) + \epsilon_k)^k \cdot \|x_0 - x^*\| && (3.5)
\end{aligned}$$

where the last step follow from Gelfand's formula [appendix: A.0.5] with $\{\epsilon_k\}_{k\in\mathbb{N}} \rightarrow_{k\rightarrow\infty} 0$. The spectral radius corresponds to

$$\begin{aligned}
\rho(\mathbb{I}_{d\times d} - \alpha A^T A) &= \max_{i\in[d]} |\lambda_i(\mathbb{I}_{d\times d} - \alpha A^T A)| && (3.6)\\
&= \max\left\{|1 - \alpha\lambda_{max}(A^T A)|, |1 - \alpha\lambda_{min}(A^T A)|\right\} && (3.7)
\end{aligned}$$

optimising for α yields

$$|1 - \alpha^*\lambda_{max}(A^T A)| = |1 - \alpha^*\lambda_{min}(A^T A)| \implies \alpha^* = \frac{2}{\lambda_{min}(A^T A) + \lambda_{max}(A^T A)} \quad (3.8)$$

Proof. eq: 3.8
First Case: $1 - \alpha^*\lambda_{max}(A^T A) = 1 - \alpha^*\lambda_{min}(A^T A)$ where if $\lambda_{min}(A^T A) \neq \lambda_{max}(A^T A) \implies \alpha^*$. Second Case:: $1 - \alpha * \lambda_{max}(A^T A) = \alpha * \lambda_{min}(A^T A) + 1 \implies \alpha^* = 2/(\lambda_{min}(A^T A) + \lambda_{max}(A^T A))$

☺ □

Plugging α^* into for the spectral radius yields

$$\rho(\mathbb{I}_{d\times d} - \alpha^* A^T A) = \frac{\kappa(A^T A) - 1}{\kappa(A^T A) + 1} \quad (3.9)$$

Hence we recover the convergence rate proposed in standard proofs for GD [15] for μ strongly convex and L smooth function when plugging results for strong convexity and smoothness on least square objective from lemma 2.2.1

3.1.2 Convergence of mini-SGD

Applying the mini-SGD [def: 2.3.1] recursive scheme with constant step size to the consistent least squares [problem 2.2.3] yields the following iteration:

$$\underline{\mathbf{x}}_{k+1} = \underline{\mathbf{x}}_k - \alpha\frac{1}{B}\sum_{j\in[B]}\frac{1}{p_{\underline{\mathbf{s}}_j}} a_{\underline{\mathbf{s}}_j} a_{\underline{\mathbf{s}}_j}^T (\underline{\mathbf{x}}_k - x^*) \quad (3.10)$$

where a_j corresponds to the $j-th$ row of the system matrix A draw with replacement and with probability p_j. We define the following random matrix

$$\underline{\boldsymbol{M}}_k = \frac{1}{B}\sum_{j\in[B]}\frac{1}{p_{\underline{\boldsymbol{s}}_j}}a_{\underline{\boldsymbol{s}}_j}a_{\underline{\boldsymbol{s}}_j}^T \tag{3.11}$$

Next using the same proof strategy as for GD we have

$$\begin{aligned}\|\underline{\boldsymbol{x}}_{k+1}-x^*\| &= \|(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_k)(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_{k-1})\cdots(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_1)(x_0-x^*)\| \qquad (3.12)\\ &\leq \|(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_k)(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_{k-1})\cdots(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_1)\|\cdot\|x_0-x^*\| \qquad (3.13)\end{aligned}$$

taking expectation on both sides we wish to bound the following quantity

$$\mathbb{E}\|(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_k)(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_{k-1})\cdots(\mathbb{I}_{d\times d}-\alpha\underline{\boldsymbol{M}}_1)\| \tag{3.14}$$

the expectation of the spectral norm of a product of random matrices.

3.1.3 Preparation for Main Proof

Lemma 3.1.1

Define α^* as in eq: 3.8 then

$$\|\mathbb{I}_{d\times d}-\alpha^*\mathbb{E}\underline{\boldsymbol{M}}_k\| \leq \frac{\kappa(A^TA)-1}{\kappa(A^TA)+1} \quad \forall k \tag{3.15}$$

Proof. 3.1.1 Notice that

$$\mathbb{E}\underline{\boldsymbol{M}}_k = \mathbb{E}\frac{1}{B}\sum_{j\in[B]}\frac{1}{p_{\underline{\boldsymbol{s}}_j}}a_{\underline{\boldsymbol{s}}_j}a_{\underline{\boldsymbol{s}}_j}^T = A^TA$$

hence

$$\|\mathbb{I}_{d\times d}-\alpha^*\mathbb{E}\underline{\boldsymbol{M}}_k\| = \|\mathbb{I}_{d\times d}-\alpha^*A^TA\|$$

taking the SVD decomposition of system matrix $A = U\Sigma V^T$ we have

$$\begin{aligned}\|\mathbb{I}_{d\times d}-\alpha^*A^TA\| &= \|\mathbb{I}_{d\times d}-\alpha^*V\Sigma^2V^T\|\\ &= \|V(\mathbb{I}_{d\times d}-\alpha^*\Sigma^2)V^T\|\\ &= \|\mathbb{I}_{d\times d}-\alpha^*\Sigma^2\| = 1-\alpha^*\sigma_{min}^2(A)\end{aligned}$$

Note that $\sigma_{min}^2(A) = \lambda_{min}(A^TA)$ and last equation holds true by substituting $\alpha^* = 2/(\lambda_{min}(A^TA)+$

$\lambda_{max}(A^TA))$. Simplifying further yields

$$\begin{aligned} 1 - \frac{2\lambda_{min}(A^TA)}{\lambda_{min}(A^TA) + \lambda_{max}(A^TA)} &= \frac{\lambda_{max}(A^TA) - \lambda_{min}(A^TA)}{\lambda_{min}(A^TA) + \lambda_{max}(A^TA)} \\ &= \frac{\kappa(A^TA) - 1}{\kappa(A^TA) + 1} \end{aligned}$$

☺ □

Assumption 3.1.1 1

We assume that for some $\eta \geq 1$, the sampling probabilities p_j from mini-SGD satisfy

$$\eta p_j \geq \frac{\|a_j\|^2}{\|A\|_F^2} \quad \forall j \in [n] \tag{3.16}$$

Lemma 3.1.2

Define

$$\underline{\boldsymbol{W}}_j := \frac{1}{B}(-\frac{1}{p_{\underline{\boldsymbol{s}}_j}} a_{\underline{\boldsymbol{s}}_j} a_{\underline{\boldsymbol{s}}_j}^T + A^TA) \tag{3.17}$$

and let

$$\underline{\boldsymbol{W}} = \sum_{j\in[B]} \underline{\boldsymbol{W}}_j \tag{3.18}$$

where $\{\underline{\boldsymbol{s}}_j\}_{j\in[B]} \sim_{iid} \underline{\boldsymbol{s}}$, and $\mathbb{P}(\underline{\boldsymbol{s}} = j)$ satisfy assumption 3.1.1. Then

$$\sqrt{\mathbb{E}\|\underline{\boldsymbol{W}}\|} \leq \delta \tag{3.19}$$

provided

$$B \geq 8e\eta \log(2d) \max\left\{\|A\|_F^2\|A\|^2\delta^{-2}, (4\|A\|_F^4\delta^{-2})^{1/2}\right\} \tag{3.20}$$

Proof. Lemma 3.1.2 For the proof we refer to [22] lemma 2. ☺ □

Lemma 3.1.3

Define $\underline{\boldsymbol{Y}}_k = (\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_k)$ then

$$\sqrt{\mathbb{E}\|\underline{\boldsymbol{Y}}_k - \mathbb{E}\underline{\boldsymbol{Y}}_k\|^2} \leq \delta \tag{3.21}$$

provided that the **batch size** B satisfies

$$B \geq 8e\eta \log(2d) \max\left\{\|A\|_F^2\|A\|^2\alpha^2\delta^{-2}, (4\|A\|_F^4\alpha^2\delta^{-2})^{1/2}\right\} \tag{3.22}$$

Proof. 3.1.3 Notice the following

$$\begin{aligned}\mathbb{E}\|\underline{\boldsymbol{Y}}_k - \mathbb{E}\underline{\boldsymbol{Y}}_k\|^2 &= \alpha^2\mathbb{E}\|\underline{\boldsymbol{M}}_k - \mathbb{E}\underline{\boldsymbol{M}}_k\|^2 \\ &= \alpha^2\mathbb{E}\|\sum_{j\in[B]}\frac{1}{B}\big(-\frac{1}{p_{\underline{s}_j}}a_{\underline{s}_j}a_{\underline{s}_j}^T + A^TA\big)\|\end{aligned}$$

hence the result follow by a direct application of Lemma 3.1.2 which used matrix Berstein's inequality to bound the sum

$$\sum_{j\in[B]}\frac{1}{B}\big(-\frac{1}{p_{\underline{s}_j}}a_{\underline{s}_j}a_{\underline{s}_j}^T + A^TA\big)$$

🙂 □

3.1.4 Main Theorem

Theorem 3.1.1

Consider mini-SGD applied to **consistent** least squares problem 2.2.3 whose sampling probabilities satisfy assumption 3.1.1. Fix parameter $\alpha = \alpha^*$ as in eq: 3.8. Define constant $L := \frac{\kappa-1}{\kappa+1}$ where κ is the 2-norm condition number of A^TA and $\overline{\kappa}$ the *smoothed* conditioned number of A^TA.
For any $k^* > 1$ choose

$$B \geq 32e\eta\overline{\kappa}d\log(2d)\max\left\{\frac{2\kappa}{(\kappa-1)^2}\frac{k^*}{\log k^*}, \frac{\sqrt{2}}{(\kappa-1)}\cdot\sqrt{\frac{k^*}{\log(k^*)}}\right\} \tag{3.23}$$

then for $Ax^* = b$, the mini-SGD iterate satisfy

$$\mathbb{E}\|\underline{\boldsymbol{x}}_k - x^*\| \leq L^k\max\{d, (k^*)^{k/k^*}\}\|x_0 - x^*\| \tag{3.24}$$

Proof. Theorem 3.1.1
Recall the following for mini-SGD iteration

$$\begin{aligned}\mathbb{E}\|\underline{\boldsymbol{x}}_{k+1} - x^*\| &\leq \mathbb{E}\|(\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_k)(\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_{k-1})\cdots(\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_1)\|\|x_0 - x^*\| \\ &=: \mathbb{E}\|\underline{\boldsymbol{Z}}_k\|\|x_0 - x^*\|\end{aligned}$$

for

$$\underline{\boldsymbol{M}}_k = \frac{1}{B}\sum_{j\in[B]}\frac{1}{p_{\underline{s}_j}}a_{\underline{s}_j}a_{\underline{s}_j}^T$$

where 3.1.1 holds for p_j. Then using lemmas 3.1.1 and 3.1.2 we can apply theorem 2.1.2

$$m_i := L \quad \sigma_i := \delta/L \implies M = L^k \quad \nu = \sum_{i\in[k]}\sigma_i^2 = k\delta^2/L^2$$

Choosing $\delta^2 := \frac{L^2 \log(k^*)}{2k^*}$ yields the bound on B in 3.1.2 i.e

$$B \geq 8e\eta \log(2d) \max\left\{C_1, \left(C_2\right)^{1/2}\right\}$$

where

$$C_1 = \frac{2\|A\|^2\|A\|_F^2 k^*(\alpha^*)^2}{L^2\log(k^*)} \quad \text{and} \quad C_2 = \frac{8\|A\|_F^4(\alpha^*)^2 k^*}{\log(k^*)L^2}$$

Plugging L and α^* yields for C_1

$$\begin{aligned}
C_1 &= 2\|A\|^2\|A\|_F^2 \frac{(\alpha^*)^2}{L^2}\frac{k^*}{\log k^*} \\
&= 8\overline{\kappa}d\lambda_{max}(A^TA)\lambda_{min}(A^TA)\frac{1}{(\lambda_{max}(A^TA)-\lambda_{min}(A^TA))^2}\frac{k^*}{\log k^*} \\
&= \frac{8d\overline{\kappa}\kappa}{(\kappa-1)^2}\frac{k^*}{\log k^*}
\end{aligned}$$

Plugging L and α^* yields for C_2

$$\begin{aligned}
C_2 &= 8\|A\|_F^2 \frac{4}{(\lambda_{max}-\lambda_{min})^2}\frac{k^*}{\log k^*} \\
&= \frac{32d^2\overline{\kappa}^2\lambda_{min}^2}{(\lambda_{max}-\lambda_{min})^2}\frac{k^*}{\log k^*} \\
\Leftrightarrow (C_2)^{1/2} &= \frac{4\sqrt{2}d\overline{\kappa}}{(\kappa-1)} \cdot \sqrt{\frac{k^*}{\log(k^*)}}
\end{aligned}$$

Plugging C_1 and C_2 in the bound for B yield

$$\begin{aligned}
B &\geq 8e\eta\log(2d)\max\left\{\frac{8d\overline{\kappa}\kappa}{(\kappa-1)^2}\frac{k^*}{\log k^*}, \frac{4\sqrt{2}d\overline{\kappa}}{(\kappa-1)} \cdot \sqrt{\frac{k^*}{\log(k^*)}}\right\} \\
&= 32e\eta\overline{\kappa}d\log(2d)\max\left\{\frac{2\kappa}{(\kappa-1)^2}\frac{k^*}{\log k^*}, \frac{\sqrt{2}}{(\kappa-1)} \cdot \sqrt{\frac{k^*}{\log(k^*)}}\right\}
\end{aligned}$$

Furthermore define:

$$\nu = \frac{k\log(k^*)}{2k^*}$$

Then

$$\mathbb{E}\|\underline{\mathbf{Z}}_k\| \leq L^k \exp\left(\sqrt{\frac{k\log(k^*)}{k^*}\max\{\frac{k\log(k^*)}{k^*}, \log(d)\}}\right)$$

notice the following for any $x \in \mathbb{R}_+$

$$\sqrt{x\max\{x, \log d\}} \leq \max\{x, \log d\}$$

hence

$$
\begin{aligned}
\mathbb{E}\|\underline{\boldsymbol{Z}}_k\| &\leq L^k \exp\left(\sqrt{\frac{k\log(k^*)}{k^*}\max\{\frac{k\log(k^*)}{k^*},\log(d)\}}\right)\\
&\leq L^k \exp\left(\max\{\frac{k\log(k^*)}{k^*},\log(d)\}\right)\\
&\leq L^k \max\{\exp(\frac{k\log(k^*)}{k^*}),d\} = L^k\max\{(k^*)^{k/k^*},d\}
\end{aligned}
$$

☺ □

Corollary 3.1.1 Under Assumptions of Theorem 3.1.1
Fix $c \in (0,2)$. There exists a parameter α such that for sufficiently large κ, the mini-SGD iterates applied to the consistent least squares problem 2.2.3 converge in expected norm at a linear rate

$$
\mathbb{E}\|\underline{\boldsymbol{x}}_k - x^*\| \leq \mathcal{O}\left(\left(1-\frac{c}{\kappa}\right)^k\right) \quad \text{given} \quad B \geq \mathcal{O}\Big(\eta d\log(d)\overline{\kappa}\Big) \tag{3.25}
$$

Proof of Corollary 3.1.1.
Taking the result from theorem 3.1.1 we assume

$$
\mathbb{E}\|\underline{\boldsymbol{x}}_k - x^*\| \leq Rate^k\|x_0 - x^*\|
$$

where $Rate = Lk^* / \log k^*$ then

$$
\begin{aligned}
Rate = L(k^*)^{1/k^*} =: L^{1-\delta} &\implies -\delta\log(L) = \log(k^*)/k^*\\
&\Leftrightarrow \delta = \frac{\log(k^*)}{k^*\log(1/L)}
\end{aligned}
$$

Suppose that for κ sufficiently large, we have

$$
\begin{aligned}
L = \frac{\kappa-1}{\kappa+1} &\approx 1-\frac{2}{\kappa} = \exp(-\kappa/2)+\mathcal{O}(-\kappa^{-2})\\
\implies \log(L) &= \log(\exp(-\kappa/2)+\mathcal{O}(-\kappa^{-2}))\\
&\approx \log(1-2/\kappa-\mathcal{O}(\kappa^2))
\end{aligned}
$$

Using Taylor polynomial around zero for

$$
\log(1-x) = -x - x^2/2 + \mathcal{O}(x^3)
$$

replace $x = 2/\kappa + \mathcal{O}(\kappa^{-2})$ yield

$$
\log(1+2/\kappa+\mathcal{O}(\kappa^2)) \approx -2/\kappa - \mathcal{O}(\kappa^{-2})
$$

Hence

$$\log(1/L) = -\log(L) = 2/\kappa + \mathcal{O}(\kappa^{-2})$$

Define $\frac{k^*}{\log k^*} =: \frac{\kappa}{c}$ for some constant $c > 0$ then

$$\begin{aligned} 1-\delta &= 1 - \frac{c}{\kappa}\frac{1}{\log(1/L)} \\ &= 1 - \frac{c}{2+\mathcal{O}(\kappa^{-1})} \end{aligned}$$

Take Taylor polynomial of

$$\frac{1}{x+2} \approx \frac{1}{2}(1 - \frac{x}{2} + \frac{x^2}{4})$$

replace $x := 2 + \mathcal{O}(\kappa^{-1})$ yield

$$\begin{aligned} \frac{c}{2+\mathcal{O}(\kappa^{-1})} &\approx \frac{c}{2} - \mathcal{O}(\kappa^{-1}) \\ \implies 1-\delta &= 1 - \frac{c}{2} + \mathcal{O}(\kappa^{-1}) \end{aligned}$$

Hence

$$\begin{aligned} L^{1-\delta} &= (1 - 2/\kappa + \mathcal{O}(\kappa^{-2}))^{\left(1-c/2+\mathcal{O}(\kappa^{-1})\right)} \\ &\approx \exp\Bigg(\Big(-2/\kappa + \mathcal{O}(\kappa^{-2})\Big)\Big((1-c/2) + \mathcal{O}(\kappa^{-1})\Big)\Bigg) \\ &= \exp\Bigg(-2/\tilde{c} + \mathcal{O}(\kappa^{-2})\Bigg) \\ &\approx 1 - \tilde{c}/\kappa + \mathcal{O}(\kappa^{-2}) \end{aligned}$$

where $\tilde{c} := 1 - c/2$.
Plugging $\frac{k*}{\log k^*} =: \frac{\kappa}{c}$ in the bound for the batch size in theorem 3.1.1 yield

$$\begin{aligned} &\text{Term1:} && \frac{\kappa}{(\kappa-1)}\frac{\kappa}{c} \approx_{(\kappa>>1)} \in \mathcal{O}(1) \\ &\text{Term2:} && \frac{\sqrt{2}}{\kappa-1}\sqrt{\frac{\kappa}{c}} \approx_{(\kappa>>1)} \in \mathcal{O}(1/\sqrt{\kappa}) \end{aligned}$$

where

$$\begin{aligned} B \geq \mathcal{O}\Bigg(\eta d \log(d)\overline{\kappa}\Bigg) \max\Bigg\{\text{Term 1}, \text{Term 2}\Bigg\} \\ \implies B \geq \mathcal{O}\Bigg(\eta d \log(d)\overline{\kappa}\Bigg) \end{aligned}$$

☺ □

Corollary 3.1.2 Under Assumptions of Theorem 3.1.1 and Corollary 3.1.1
The $\epsilon-$expected error of mini-SGD on consistent least squares problem 2.2.3 is

$$k - \text{iterations} \geq \mathcal{O}\Big(\kappa \log(1/\epsilon)\Big) \quad \text{given} \quad B \geq \mathcal{O}\Big(\eta d \log(d) \overline{\kappa}\Big) \tag{3.26}$$

Proof of Corollary 3.1.2.
From lemma A.0.1 we have for $\epsilon \in]0,1[$

$$k \geq \frac{1}{1-L}\mathcal{O}(\log(1/\epsilon)) \implies \mathbb{E}\|\underline{\boldsymbol{x}}_k - x^*\| \leq \epsilon$$

where from corollary 3.1.1

$$\frac{1}{1-L} = \kappa/c \in \mathcal{O}(\kappa) \quad \text{given} \quad B \geq \mathcal{O}\Big(\eta d \log(d) \overline{\kappa}\Big)$$

☺ □

3.2 High Probability Bound for mini-SGD on Quadratics

In this section, we provide stronger results than the bounds in expectation from last section, namely bounds with high probability for the mini-SGD algorithm applied to consistent least square problem 2.2.3. The assumption 3.1.1 still yield. These high probability bounds offer several advantages that make them "stronger" in a meaningful sense. First, they ensure greater reliability, as they guarantee that the algorithm's performance stays within specified limits with high confidence, rather than just on average. This is particularly relevant for practical applications, where we often need assurances that hold for specific runs of the algorithm, not just on average over many executions. Moreover, high probability bounds naturally correspond to confidence intervals, providing a clear statistical interpretation of the algorithm's behavior. This correspondence allows for a more intuitive understanding of the guarantees provided. Importantly, these bounds also capture crucial information about the tail behavior of the random variables involved. This insight into extreme cases is essential for a comprehensive understanding of the algorithm's performance, especially when considering worst-case scenarios.

Following the derivation in section *Convergence of mini-SGD* (3.1.2), we aim to control the following expression:

$$\|(\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_k)(\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_{k-1}) \cdots (\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_1)\| =: \|\underline{\boldsymbol{Z}}_k\| \tag{3.27}$$

By "control," we mean establishing an exponentially decaying tail bound, which we achieve using Theorem 2.1.3. It's important to note that for the tail bound in Theorem 2.1.2, the conditions differ from those in the expectation case. Specifically, we require:

$$\|\underline{\boldsymbol{M}}_i - \mathbb{E}\underline{\boldsymbol{M}}_i\| \leq \sigma_i \cdot m_i \qquad \text{almost surely} \quad \forall i \in [k] \tag{3.28}$$

To ensure this condition holds with high probability, we provide a high-probability bound version of

Lemma 3.1.2.

Lemma 3.2.1

Define

$$\underline{\boldsymbol{W}}_j = \frac{1}{B}(-\frac{1}{p_{\underline{\boldsymbol{s}}_j}} a_{\underline{\boldsymbol{s}}_j} a_{\underline{\boldsymbol{s}}_j} T + A^T A) \tag{3.29}$$

and let

$$\underline{\boldsymbol{W}} = \sum_{j \in [B]} \underline{\boldsymbol{W}}_j \tag{3.30}$$

where $\{\underline{\boldsymbol{s}}_j\}_{j \in [B]} \sim_{iid} \underline{\boldsymbol{s}}$, and $\mathbb{P}\big(\underline{\boldsymbol{s}} = j\big)$ satisfy assumption 3.1.1.
Then with probability at least $1 - \delta$ it holds for some $t > 0$

$$\|\underline{\boldsymbol{W}}\| \leq t \tag{3.31}$$

provided

$$B \geq 2\eta \cdot \|A\|_F^2 \cdot \left(\|A\|^2 \cdot t^{-2} + \frac{2t^{-1}}{3} \right) \cdot \left(\log(1/\delta) + \log(2d) \right) \tag{3.32}$$

Proof of Lemma 3.2.1.
According to theorem 2.1.1 we have

$$\mathbb{P}\left\{ \|\underline{\boldsymbol{W}}\| \geq t \right\} \leq 2d \exp\left(\frac{-t^2/2}{\nu(\underline{\boldsymbol{Z}}) + \frac{Wt}{3}} \right)$$

where

$$\nu(\underline{\boldsymbol{Z}}) = \max\left\{ \sum_j \mathbb{E}\underline{\boldsymbol{W}}_j \underline{\boldsymbol{W}}_j^T, \sum_j \mathbb{E}\underline{\boldsymbol{W}}_j^T \underline{\boldsymbol{W}}_j \right\}$$
$$\|\underline{\boldsymbol{W}}_j\| \leq W \quad \forall j \in [B]$$

We can upper bound the variance $\nu(\underline{\boldsymbol{Z}})$ as follow:

$$\underline{\boldsymbol{W}}_j \underline{\boldsymbol{W}}_j^T = \underline{\boldsymbol{W}}_j^T \underline{\boldsymbol{W}}_j = \frac{1}{B^2}\left(\frac{\|a_{\underline{\boldsymbol{s}}_j}\|^2}{p_{\underline{\boldsymbol{s}}_j^2}} a_{\underline{\boldsymbol{s}}_j} a_{\underline{\boldsymbol{s}}_j}^T - \frac{1}{p_{\underline{\boldsymbol{s}}_j}} a_{\underline{\boldsymbol{s}}_j} a_{\underline{\boldsymbol{s}}_j}^T A^T A - \frac{1}{p_{\underline{\boldsymbol{s}}_j}} A^T A a_{\underline{\boldsymbol{s}}_j} a_{\underline{\boldsymbol{s}}_j}^T + A^T A \right)$$

then by $\mathbb{E}(p_{\underline{\boldsymbol{s}}_j}^{-1} a_{\underline{\boldsymbol{s}}_j} a_{\underline{\boldsymbol{s}}_j}^T) = A^T A$ and assumption 3.1.1 $\|a_{\underline{\boldsymbol{s}}_j}\|^2 \leq \eta p_{\underline{\boldsymbol{s}}_j} \|A\|_F^2$ we have

$$\mathbb{E}\underline{\boldsymbol{W}}_j \underline{\boldsymbol{W}}_j^T = \frac{1}{B^2}\left(\sum_{i \in [n]} p_i^{-1} \|a_i\|^2 a_i a_i^T - (A^T A) \right) \preceq \frac{1}{B^2}\left(\eta \|A\|_F^2 A^T A - (A^T A)^2 \right)$$
$$\Leftrightarrow \|\mathbb{E}\underline{\boldsymbol{W}}_j \underline{\boldsymbol{W}}_j^T\| \leq \frac{1}{B^2} \|(\eta \|A\|_F^2 \mathbb{I}_{d \times d} - A^T A) A^T A\| \leq \frac{\eta \|A\|_F^2 \|A\|^2}{B^2}$$

hence by triangle inequality we have

$$\nu(\underline{\boldsymbol{W}}) \leq \|\sum_j \mathbb{E}\underline{\boldsymbol{W}}_j\underline{\boldsymbol{W}}_j^T\| \leq \sum_j \|\mathbb{E}\underline{\boldsymbol{W}}_j\underline{\boldsymbol{W}}_j^T\| \leq \frac{\eta\|A\|_F^2\|A\|^2}{B}$$

We can upper bound the quantity $\|\underline{\boldsymbol{W}}_j\|$ as follow

$$\|\underline{\boldsymbol{W}}_j\| \leq \frac{1}{B}\left(\frac{1}{p_{\underline{\boldsymbol{s}}_j}}\|a_{\underline{\boldsymbol{s}}_j}a_{\underline{\boldsymbol{s}}_j}^T\| + \|A\|^2\right) \leq_{\text{Assumption 3.1.1}} \frac{\eta\|A\|_F^2 + \|A\|^2}{B} \leq \frac{2\eta\|A\|_F^2}{B} =: W$$

Plug-in the upper bound on $\nu(\underline{\boldsymbol{W}})$ and the value W we have

$$\mathbb{P}\left\{\|\underline{\boldsymbol{W}}\| \geq t\right\} \leq 2d\exp\left(\frac{-3t^2B}{6\eta\|A\|_F^2\|A\|^2 + 4\eta\|A\|_F^2 t}\right)$$

we find a condition such that for $\delta \in]0,1[$ we have

$$\begin{aligned}
2d\exp\left(\frac{-3t^2B}{6\eta\|A\|_F^2\|A\|^2 + 4\eta\|A\|_F^2 t}\right) \leq \delta \quad &\Leftrightarrow \\
\frac{3t^2B}{6\eta\|A\|_F^2\|A\|^2 + 4\eta\|A\|_F^2 t} \geq -\log(\delta) + \log(2d) \quad &\Leftrightarrow \\
B \geq (2\eta\|A\|_F^2\|A\|^2 t^{-2} + \frac{2}{3}\eta\|A\|_F^2 t^{-1})(-\log(\delta) + \log(2d))&
\end{aligned}$$

☺ □

Theorem 3.2.1

Consider mini-SGD applied to **consistent** least squares problem 2.2.3 whose sampling probabilities satisfy assumption 3.1.1. Fix parameter $\alpha = \alpha^*$ as in eq: 3.8. Define constant $L := \frac{\kappa-1}{\kappa+1}$ where κ is the 2-norm condition number of A^TA and $\bar{\kappa}$ the *smoothed* conditioned number of A^TA.
Assume solution $Ax^* = b$ and initial radius $D_0 = \|x_0 - x^*\|$.
For any $k^* > 1$ choose

$$B \geq 8\eta\cdot\bar{\kappa}\cdot d\cdot\left(2\frac{\kappa}{(\kappa-1)^2}\frac{k^*}{\log(k^*)} + \frac{4}{3}\sqrt{\frac{2k^*}{\log(k^*)}\frac{1}{\kappa-1}}\right)\cdot\left(\log(1/\tilde{\delta}) + \log(2d)\right) \tag{3.33}$$

with $\tilde{\delta}$ corresponding to a $(1-k\tilde{\delta})$ probability of failure of the statement, then the error norm of the iterative algorithm can be controlled with probability at least $1-\delta$

$$\mathbb{P}\left\{\|\underline{\boldsymbol{x}}_k - x^*\| \leq L^k\exp\left(\sqrt{\frac{k\log(k^*)}{k^*}}\log(d/\delta)\right)D_0\right\} \geq 1-\delta \tag{3.34}$$

if

$$\log(1/\delta) \geq k\frac{k^*}{\log(k^*)} - \log(d) \tag{3.35}$$

Proof of Theorem 3.2.1.

Define the error process of mini-SGD

$$\|\underline{\boldsymbol{e}}_k\| := \|\underline{\boldsymbol{x}}_k - x^*\| \leq \|\underline{\boldsymbol{Z}}_k\| \underbrace{\|x_0 - x^*\|}_{=:D_0}$$

where

$$\underline{\boldsymbol{Z}}_k := (\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_k)(\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_{k-1}) \cdots (\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_1)$$
$$\underline{\boldsymbol{M}}_i = \frac{1}{B} \sum_{j\in[B]} \frac{1}{p_{\underline{\boldsymbol{s}}_j}} a_{\underline{\boldsymbol{s}}_j} a_{\underline{\boldsymbol{s}}_j}^T \qquad \forall i \in [k]$$

Note the event inclusion $E_1 := \{\|\underline{\boldsymbol{e}}_k\| \geq CD_0\} \implies E_2 := \{\|\underline{\boldsymbol{Z}}_k\| \geq C\} \implies E_1 \subseteq E_2$ yield

$$\mathbb{P}\Big\{E_1\Big\} \leq \mathbb{P}\Big\{E_2\Big\}$$

Using theorem 2.1.3 we can control the norm of the product of random matrices as

$$\mathbb{P}\Big\{\|\underline{\boldsymbol{Z}}_k\| \geq t_1 M\Big\} \leq d \exp\left(\frac{-\log(t_1)^2}{2\nu}\right) \quad \textit{when} \quad \log(t_1) \geq 2\nu$$

if the following conditions holds:

- Condition **A**:

$$\|\mathbb{I}_{d\times d} - \alpha \mathbb{E} \underline{\boldsymbol{M}}_i\| \leq m_i \quad \forall i \in [k] \quad M := \prod_{i\in[k]} m_i$$

- Condition **B**:

$$\| - \underline{\boldsymbol{M}}_i + \mathbb{E}\, \underline{\boldsymbol{M}}_i\| \leq \sigma_i m_i \quad \forall i \in [k] \quad \nu := \sum_{i\in[k]} \sigma_i^2$$

Condition **A** is fulfilled be lemma 3.1.1 given the step size $\alpha = \alpha^* = \frac{2}{\lambda_{min}(A^TA)+\lambda_{max}(A^TA)}$ with

$$m_i = \frac{\kappa(A^TA) - 1}{\kappa(A^TA) + 1} =: L \implies M = \left(\frac{\kappa(A^TA) - 1}{\kappa(A^TA) + 1}\right)^k = L^k$$

Condition **B** holds with probability $1 - k\tilde{\delta}$ uniformly over all $i = \{1, \dots, k\}$. Indeed by lemma 3.2.1 we have

$$\mathbb{P}\Big\{\| - \underline{\boldsymbol{M}}_i + \mathbb{E}\underline{\boldsymbol{M}}_i\| \leq t_2/\alpha^*\Big\} \geq 1 - \tilde{\delta}$$

given

$$B \geq 2\eta \cdot \|A\|_F^2 \cdot \left(\|A\|^2 \cdot t_2^{-2}(\alpha^*)^2 + \alpha^* \frac{2t_2^{-1}}{3}\right) \cdot \Big(\log(1/\delta) + \log(2d)\Big)$$

Define the event $E_i := \{\| - \underline{\boldsymbol{M}}_i + \mathbb{E}\|\underline{\boldsymbol{M}}_i\| \leq t_2\}$ we need the condition to hold uniformly over all

$i \in [k]$ hence by the following argument

$$\mathbb{P}\left\{ \cap_{i\in[k]} E_i \right\} = 1 - \mathbb{P}\left\{ \cup_{i\in[k]} E_i^c \right\} \geq 1 - k\tilde{\delta}$$

We define the following:

$$\sigma_i = t_2/L \implies \nu = kt_2^2/L^2$$

choose

$$t_2^2 := L^2 \log(k^*)/2k^* \implies \nu = \frac{k\log(k^*)}{2k^*}$$

which yield

$$\mathbb{P}\left\{ \|\underline{\boldsymbol{Z}}_k\| \geq t_1 M \right\} \leq d\exp\left(\frac{-\log(t_1)^2 k^*}{k\log(k^*)} \right) \quad \textit{when} \quad \log(t_1) \geq \frac{k\log(k^*)}{k^*}$$

if with probability at least $1 - k\tilde{\delta}$

$$\begin{aligned} B &\geq 2\eta \cdot \|A\|_F^2 \cdot \left(\|A\|^2 \cdot t_2^{-2}(\alpha^*)^2 + \alpha^* \frac{2t_2^{-1}}{3} \right) \cdot \left(\log(1/\tilde{\delta}) + \log(2d) \right) \\ &= 2\eta\bar{\kappa} d\lambda_{min}(A^T A)\left(\lambda_{max}(A^T A)\frac{(\alpha^*)^2}{L^2}\frac{2k^*}{\log(k^*)} + 2/3\sqrt{\frac{2k^*}{\log(k^*)}\frac{\alpha^*}{L}} \right) \cdot \left(\log(1/\tilde{\delta}) + \log(2d) \right) \end{aligned}$$

Notice that

$$\frac{\alpha^*}{L} = \frac{2}{\lambda_{max}(A^T A) - \lambda_{min}(A^T A)}$$

which yield for the minimum batch size

$$B \geq 8\eta \cdot \bar{\kappa} \cdot d \cdot \left(2\frac{\kappa}{(\kappa-1)^2}\frac{k^*}{\log(k^*)} + \frac{4}{3}\sqrt{\frac{2k^*}{\log(k^*)}\frac{1}{\kappa - 1}} \right) \cdot \left(\log(1/\tilde{\delta}) + \log(2d) \right)$$

We wish to control with probability at least δ i.e

$$\begin{aligned} \mathbb{P}\left\{ \|\underline{\boldsymbol{Z}}_k\| \geq t_1 M \right\} &\leq d\exp\left(\frac{-\log(t_1)^2 k^*}{k\log(k^*)} \right) = \delta \\ \Leftrightarrow t_1 &= \exp\left(\sqrt{\frac{k\log(k^*)}{k^*} \cdot \log(d/\delta)} \right) \implies \\ \mathbb{P}\left\{ \|\underline{\boldsymbol{Z}}_k\| \geq M\exp\left(\sqrt{\frac{k\log(k^*)}{k^*}\log(d/\delta)} \right) \right\} &\leq \delta \end{aligned}$$

Recall that we need the condition $\log(t_1) \geq 2\nu$ to hold, i.e

$$\log(1/\delta) \geq k\frac{k^*}{\log(k^*)} - \log(d)$$

☺ □

3.2.1 Interpretation of Theorem and Limitations

To enhance the interpretability of Theorem 3.2.1, we present a refined corollary that specifies the convergence rate and establishes a lower bound on the required batch size.

Corollary 3.2.1 Under Assumptions of Theorem 3.2.1
Fix

$$c \in \left(0, \frac{2}{\log(d/\delta)}\right) \tag{3.36}$$

Then for all κ sufficiently large, the error norm of mini-SGD on consistent least squares 2.2.3 converges with probability at least $1-\delta$ at a linear rate, i.e.,

$$\mathbb{P}\left\{\|\underline{\boldsymbol{x}}_k - x^*\| \leq \left(1 - \frac{c}{\kappa}\right)^k D_0\right\} \geq 1 - \delta \quad \text{for} \quad k \leq \frac{\kappa \log(d/\delta)}{c} \tag{3.37}$$

Provided that with probability at least $1 - k\tilde{\delta}$,

$$B \geq \mathcal{O}\left(\eta\bar{\kappa}d\log(d/\tilde{\delta})\right) \tag{3.38}$$

Proof of Corollary 3.2.1.
Define

$$\frac{k^*}{\log k^*} := \kappa/c \quad \text{with} \quad c > 0$$

Recall from proof of corollary 3.1.1

$$L \approx_{\kappa>>1} 1 - 2/\kappa + \mathcal{O}\left(\kappa^{-2}\right)$$

Note from 3.2.1 eq: 3.34

$$\begin{aligned}
\|\underline{\boldsymbol{x}}_k - x^*\| &\leq L^k \exp\left(\sqrt{k\frac{c}{\kappa}\log(d/\delta)}\right) D_0 \\
&\leq \left(L \exp\left(\frac{c}{\kappa}\log(d/\delta)\right)\right)^k D_0 \\
&=: \left(L^{1-p}\right)^k D_0
\end{aligned}$$

hence we have for $1-p$

$$1 - p = 1 - \frac{c\log(d/\delta)}{\kappa}\frac{1}{\log(1/L)}$$

where from the proof in corollary 3.1.1 we know

$$1 - p = 1 - \frac{c\log(d/\delta)}{\kappa}\frac{1}{\log(1/L)} \approx 1 - \frac{c\log(d/\delta)}{2} + \mathcal{O}\Big(\kappa^{-1}\Big)$$

then

$$\begin{aligned} L^{1-p} &\approx \Big(1 - 2/\kappa + \mathcal{O}\big(\kappa^{-2}\big)\Big)^{\left(1 - \frac{c\log(d/\delta)}{2} + \mathcal{O}\Big(\kappa^{-1}\Big)\right)} \\ &\approx \exp\left(- \frac{(2 - c\log(d/\delta))}{\kappa} + \mathcal{O}\big(\kappa^{-2}\big)\right) \\ &\approx 1 - \frac{(2 - c\log(d/\delta))}{\kappa} \end{aligned}$$

Hence for convergence we have the following condition on c

$$|1 - \frac{(2 - c\log(d/\delta))}{\kappa}| < 1 \xrightarrow{\kappa \gg 1} c \in \left(0, \frac{2}{\log(d/\delta)}\right)$$

implying the existence of such a c.
Recall the condition for theorem 3.2.1 to hold and plug in $k^*/\log(k^*) = \kappa/c$

$$\begin{aligned} \log(t_1) \geq 2\nu = k\frac{\log(k^*)}{k^*} &\quad \Leftrightarrow \\ \log(d/\delta)\frac{k^*}{\log(k^*)} \geq k &\quad \Leftrightarrow \\ \log(d/\delta)\frac{\kappa}{c} \geq k & \end{aligned}$$

For the batch size assuming with probability $1 - k\tilde{\delta}$ we have

$$\begin{aligned} B &\geq 8\eta \cdot \overline{\kappa} \cdot d \cdot \left(2\frac{\kappa}{(\kappa-1)^2}\frac{k^*}{\log(k^*)} + \frac{4}{3}\sqrt{\frac{2k^*}{\log(k^*)}\frac{1}{\kappa-1}}\right) \cdot \Big(\log(1/\tilde{\delta}) + \log(2d)\Big) \\ &= 8\eta \cdot \overline{\kappa} \cdot d \cdot \left(2\frac{\kappa}{(\kappa-1)^2}\frac{\kappa}{c} + \frac{4}{3}\sqrt{\frac{2\kappa}{c}\frac{1}{\kappa-1}}\right) \cdot \Big(\log(1/\tilde{\delta}) + \log(2d)\Big) \\ &\approx_{\kappa>>1} \in \mathcal{O}\left(\eta\overline{\kappa}d\log(d/\tilde{\delta})(1 + \frac{1}{\sqrt{\kappa}})\right) = \mathcal{O}\Big(\eta\overline{\kappa}d\log(d/\tilde{\delta}\Big) \end{aligned}$$

☺ □

Our analysis reveals a non-asymptotic limitation on the number of iterations for which the results hold:

$$k \leq \log(d/\delta)\frac{\kappa}{c} \tag{3.39}$$

This constraint warrants careful interpretation. As the probability of failure δ approaches zero, the upper bound on k tends to infinity. Furthermore, we observe an interesting trade-off between the iteration bound on k and the linear convergence rate, which is governed by c:

$$c \in \left(0, \frac{2}{\log(d/\delta)}\right) \quad \text{with corresponding linear rate } 1 - \frac{c}{\kappa} \tag{3.40}$$

This relationship presents a dilemma: minimizing c improves the convergence rate, while maximizing c extends the validity of the theorem to a larger number of iterations k. However, the practical relevance of these results lies in their application to confidence intervals, which typically involve a fixed δ. Our findings indicate that confidence intervals can only be provided for a specific range of iterations, which depends on both the problem dimension d and the condition number κ. This limitation stems from the tail bound result for the product of random matrices (Theorem 2.1.3). Consequently, while our results offer valuable insights, they also highlight the intricate balance between convergence rate, iteration count, and problem parameters in the non-asymptotic regime.
A second limitation, harder to characterize than the previous one, is the bound on the batch size:

$$B \geq \mathcal{O}\left(\eta\bar{\kappa}d\log(d/\tilde{\delta})\right) \tag{3.41}$$

The issue here is that for Theorem 3.2.1 to hold, we need a uniform bound over all $i \in [k]$ on:

$$|\underline{\boldsymbol{Y}}_i - \mathbb{E}\underline{\boldsymbol{Y}}_i| \leq \sigma_i m_i \quad \text{w.p. at least } 1 - \tilde{\delta} \quad \text{(Lemma 3.2.1)} \tag{3.42}$$

where $\underline{\boldsymbol{Y}}_i$ corresponds to

$$\underline{\boldsymbol{Y}}_i = \mathbb{I} - \alpha^* \underline{\boldsymbol{M}_i} \tag{3.43}$$

Hence, we have an inverse proportionality between $\tilde{\delta}$ and the iteration number k, meaning that there exists a non-asymptotic bound on the number of iterations k for the batch size to be meaningful, i.e., less than the number of rows n. This bound is of the form:

$$k \lesssim \exp\left(\frac{n}{\eta\bar{\kappa}d}\right)d^{-1} \Leftrightarrow B \lesssim n \tag{3.44}$$

Therefore, in the large-scale system regime (i.e., $n \gg d$), this limitation should not be constraining.

3.3 Mini-Batch SGD converges as GD for on "Almost" Quadratics

In this section, we aim to study what could be seen as an extension to 3.1.1 on functions that behave closely to quadratics in the sense that they are upper and lower bounded by quadratics, or technically speaking, are μ-strongly convex and L-smooth as in definitions A.0.2 and A.0.3. We assume the objective to be a finite sum problem with interpolation as in problem 2.2.4. Inspired by the setting in [19] used to prove heavy tail phenomenon of SGD, we assume the initial iterate x_0 of the iterative method to be in the domain of attraction of a minimum x^* and assume that the function is well approximated by a quadratic in this basin. This motivation is somewhat informal but can be translated as approximating the gradient $\nabla f_i(x)$ by its Taylor approximation around the minimum,

$$\nabla f_i(x) \approx \nabla f_i(x^*) + \nabla^2 f_i(x^*)(x - x^*) \tag{3.45}$$

Replacing the gradient approximation in the mini-SGD algorithm yields an affine stochastic approximation of the problem:

$$\underline{\boldsymbol{x}}_{k+1} \approx \underline{\boldsymbol{x}}_k - \alpha \frac{1}{B} \sum_{j\in[B]} \frac{1}{p_{\underline{\boldsymbol{s}}_j}} \left(\nabla f_{\underline{\boldsymbol{s}}_j}(x^*) + \nabla^2 f_{\underline{\boldsymbol{s}}_j}(x^*)(\underline{\boldsymbol{x}}_k - x^*)\right) \tag{3.46}$$

which we analyze in the interpolation regime (definition 2.2.1), i.e., $\nabla f_i(x^*) = 0 \ \forall i \in [n]$, yielding the linear stochastic error process:

$$\underline{\boldsymbol{x}}_{k+1} - x^* \approx \left(\mathbb{I}_{d\times d} - \alpha \frac{1}{B} \sum_{j\in[B]} \frac{1}{p_{\underline{\boldsymbol{s}}_j}} \nabla^2 f_{\underline{\boldsymbol{s}}_j}(x^*)\right)(\underline{\boldsymbol{x}}_k - x^*) \tag{3.47}$$

$$=: \left(\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_k\right)(\underline{\boldsymbol{x}}_k - x^*) \tag{3.48}$$

An interesting question arises: how does the control of this stochastic error process, using the same concentration techniques as in section 3.1, compare to standard techniques? We address this question in this section.

3.3.1 GD Convergence

We propose a proof of convergence for the GD algorithm with fixed step size α applied to the finite sum problem 2.2.4. The convergence rate will be the one we wish to achieve using "less" data i.e using mini-SGD.

Lemma 3.3.1

Assume $f(x)$ as in problem 2.2.4, $\mu := \sum_{i\in[n]} \mu_i$ and $L := \sum_{i\in[n]} L_i$. Given the step size $\alpha^* := \frac{2}{L+\mu}$ GD iterates converges linearly as

$$\|x_{k+1} - x^*\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|x_0 - x^*\| \tag{3.49}$$

where $\kappa := \frac{L}{\mu}$

Proof of lemma 3.3.1.
First notice that $f(x) \in \mathcal{F}_{\mu,L}$ with parameters $\mu = \sum_{i\in[n]} \mu_i$ and $L = \sum_{i\in[n]} L_i$. Indeed using smoothness definition A.0.3 we have

$$\begin{aligned}\|\nabla f(y) - \nabla f(x)\| &\leq \sum_{i\in[n]} \|\nabla f_i(y) - \nabla f_i(x)\| \\ &\leq \sum_{i\in[n]} L_i \|y - x\| \quad \forall x, y \in dom(f)\end{aligned}$$

Using strong convexity equivalence lemma A.0.2 we have

$$\nabla^2 g(x) := \nabla^2 f(x) - \mu \mathbb{I}_{d\times d} \succeq 0 \Leftrightarrow f \text{ is } \mu - \text{strongly convex}$$

Note that by definition of strong convexity

$$\begin{aligned} \nabla^2 f(x) = \sum_{i\in[n]} \nabla^2 f_i(x) \succeq \sum_{i\in[n]} \mu_i \mathbb{I}_{d\times d} \\ \Longrightarrow \\ g(x) \succeq 0 \text{ if } \mu = \sum_{i\in[n]} \mu_i \end{aligned}$$

Next we define the following operator

$$T : \mathbb{R}^d \to \mathbb{R}^d \quad T(x) := x - \alpha \nabla f(x)$$

In interpolation regime

$$\begin{aligned} T(x^*) = x^* \implies \|x_{k+1} - x^*\| &= \|T(x_k) - T(x^*)\| \\ &\leq B\|x^k - x^*\| \end{aligned}$$

where B is the Lipschitz constant of the operator T i.e

$$\begin{aligned} \|\nabla^2 T(x)\| \leq B \quad &\Leftrightarrow \\ \|\mathbb{I}_{d\times d} - \alpha \nabla^2 f(x)\| \leq B \quad &\Leftrightarrow \\ \max_{i\in[d]} |\lambda(\mathbb{I}_{d\times d} - \alpha\nabla^2 f(x))| \leq B \quad &\Leftrightarrow \text{(lemma A.0.4)} \\ \max\Big\{|1 - \alpha\mu|, |1 - \alpha L|\Big\} \leq B& \end{aligned}$$

optimising with respect to α yield

$$\alpha^* = \frac{2}{\mu + L}$$

hence by plugging α^* and choosing $B := \frac{L-\mu}{L+\mu}$ we arrive at the desired linear rate. ☺ □

3.3.2 Preparation for Main Proof

Define the following quantities

$$\underline{\boldsymbol{Y}}_i := \mathbb{I}_{d\times d} - \alpha^* \underline{\boldsymbol{M}}_i \tag{3.50}$$

$$\underline{\boldsymbol{M}}_i := \frac{1}{B} \sum_{j\in[B]} \frac{1}{p_{\underline{\mathbf{s}}_j}} \nabla^2 f_{\underline{\mathbf{s}}_j}(x^*) \tag{3.51}$$

Recall from proof of lemma 3.3.1 we showed that $f \in \mathcal{F}_{\mu,L}$ with $L = \sum_{i\in[n]} L_i$ and $\mu = \sum_{i\in[n]} \mu_i$

Lemma 3.3.2

Given $\alpha^* = \frac{2}{L+\mu}$ we have

$$\|\mathbb{E}\underline{\boldsymbol{Y}}_i\| \leq \frac{L-\mu}{L+\mu} \quad \forall i \in [n] \tag{3.52}$$

Proof of lemma 3.3.2.

To simplify the notation define $H_i := \nabla^2 f_i(x^*)$. Then

$$\begin{aligned}
\|\mathbb{E}\underline{\boldsymbol{Y}}_i\| &= \|\mathbb{I}_{d\times d} - \alpha^* \frac{1}{B}\mathbb{E}\sum_{j\in\underline{\boldsymbol{S}}_i} \frac{1}{p_{\underline{\boldsymbol{s}}_j}} H_{\underline{\boldsymbol{s}}_j}\| \\
&= \|\mathbb{I}_{d\times d} - \alpha^* \underbrace{\sum_{i\in[n]} H_i}_{=:H} \| \\
&\leq \max_{i\in[n]} |1 - \alpha^* \lambda_i(H)|
\end{aligned}$$

where $H = \nabla^2 f(x^*)$. Plugging α^* and using lemma A.0.4 we have

$$\begin{aligned}
\max_{i\in[n]} |1 - \alpha^*\lambda_i(H)| &= \max\left\{|1 - \frac{2\lambda_{min}(H)}{\mu+L}|, |1 - \frac{2\lambda_{max}(H)}{\mu+L}|\right\} \\
&\leq \frac{L-\mu}{L+\mu}
\end{aligned}$$

☺ □

Assumption 3.3.1

Assume for some $\eta > 1$ that the following condition holds for the probabilities p_i in mini-SGD (2.3.1)

$$\frac{\|H_i\|}{p_i} \leq \eta\|H\| \quad \forall i \in [n] \tag{3.53}$$

where

$$H := \nabla^2 f(x^*) \qquad H_i = \nabla^2 f_i(x^*) \tag{3.54}$$

Lemma 3.3.3

Under assumption 3.3.1 we have

$$\left(\mathbb{E}\|\underline{\boldsymbol{Y}}_i - \mathbb{E}\underline{\boldsymbol{Y}}_i\|\right)^{1/2} \leq \delta \tag{3.55}$$

provided that the batch size B is large enough

$$B \geq 8e\log(2d)\max\left\{L(\eta L-\mu)\delta^{-2}\alpha^2, \left(4L^2\delta^{-2}\alpha^2\eta^2\right)^{\frac{1}{2}}\right\} \tag{3.56}$$

Proof of lemma 3.3.3 .

Define $\nabla^2 f(x^*) =: H$ and $\nabla^2 f_i(x^*) =: H_i$ then

$$\begin{aligned}
\mathbb{E}\|\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_i - \mathbb{E}\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_i\|^2 &= \alpha^2\mathbb{E}\| - \underline{\boldsymbol{M}}_i + H\|^2 \\
&= \alpha^2\mathbb{E}\| \sum_{j\in[B]} \underbrace{\frac{1}{B}\Big(- \frac{1}{p_{\underline{\mathbf{s}}_j}} H_{\underline{\mathbf{s}}_j} + H\Big)}_{=:\underline{\boldsymbol{W}}_j} \|^2 \\
&= \alpha^2\mathbb{E}\| \sum_{j\in[B]} \underline{\boldsymbol{W}}_j\|^2 =: \alpha^2\mathbb{E}\|\underline{\boldsymbol{W}}\|^2
\end{aligned}$$

Thus, the problem is reduced to controlling a sum of random matrices. To achieve this, we employ Bernstein-type results from Theorem 2.1.1. The following conditions are necessary:

1. $\|\underline{\boldsymbol{W}}_i\| \leq W$ and $\mathbb{E}\|\underline{\boldsymbol{W}}_i\| = 0 \quad \forall i \in [d]$
2. $\nu(\underline{\boldsymbol{Z}}) \leq \tilde{C}$

then we have

$$\sqrt{\mathbb{E}\|\underline{\boldsymbol{W}}\|^2} \leq \sqrt{2e\tilde{C}\log(2d)} + 4eW\log(2d)$$

For 1. we have for all realisations $\underline{\boldsymbol{W}}_j = W_j$:

$$\begin{aligned}
\|W_j\| &= \|B^{-1}\Big(- p_j^{-1}H_j + H\Big)\| \\
&\leq \frac{1}{B}(p_j^{-1}\|H_j\| + \|H\|) \\
&\leq \frac{1}{B}(\eta+1)\|H\| \leq \frac{2L\eta}{B} \quad \forall j \in [n]
\end{aligned}$$

and $\mathbb{E}\|\underline{\boldsymbol{W}}_j\| = 0$ trivially for the fact the randomness in mini-SGD leads to an unbiased estimator of the gradient.

For 2. we have

$$\underline{\boldsymbol{W}}_j\underline{\boldsymbol{W}}_j^T = \underline{\boldsymbol{W}}_j^T\underline{\boldsymbol{W}}_j = \frac{1}{B^2}\Big(p_{\underline{\mathbf{s}}_j}^{-2}H_{\underline{\mathbf{s}}_j}^2 - p_{\underline{\mathbf{s}}_j}^{-1}H_{\underline{\mathbf{s}}_j}H - p_{\underline{\mathbf{s}}_j}^{-1}HH_{\underline{\mathbf{s}}_j} + H^2\Big)$$

taking expectation leads to

$$\mathbb{E}\underline{\boldsymbol{W}}_j\underline{\boldsymbol{W}}_j^T = \frac{1}{B^2}\Big(\sum_{i\in[n]} p_i^{-1}H_i^2 - H^2\Big)$$

Notice

$$H_i^2 \preceq \|H_i\|H_i \quad \forall i \in [n] \quad H^2 \succeq \lambda_{min}(H)H \succeq \mu H$$

then

$$\begin{aligned}\frac{1}{B^2}\Big(\sum_{i\in[n]} p_i^{-1}H_i^2 - H^2\Big) &\preceq \frac{1}{B^2}\Big(\sum_{i\in[n]} p_i^{-1}\|H_i\|H_i - \mu H\Big)\\ &\preceq \frac{1}{B^2}\Big(\eta\|H\|\sum_{i\in[n]} H_i - \mu H\Big) \quad \text{Assumption: 3.3.1}\\ &\preceq \frac{1}{B^2}\Big(\eta L - \mu\Big)H\end{aligned}$$

Hence

$$\nu(\underline{\boldsymbol{W}}) = \|\sum_{j\in[B]}\mathbb{E}\underline{\boldsymbol{W}}_j\underline{\boldsymbol{W}}_j\| \leq \sum_{j\in[B]}\|\mathbb{E}\underline{\boldsymbol{W}}_j\underline{\boldsymbol{W}}_j\| \leq \frac{L(\eta L-\mu)}{B}$$

Define $\tilde{C} := \frac{L(\eta L-\mu)}{B}$ and $W := \frac{2L\eta}{B}$, then

$$\begin{aligned}\sqrt{\mathbb{E}\|\underline{\boldsymbol{W}}\|^2} &\leq \sqrt{2e\log(2d)L(\eta L-\mu)B^{-1}} + 8e\log(2d)L\eta B^{-1}\\ &\leq \delta\\ &\Leftrightarrow\\ \sqrt{2e\log(2d)L(\eta L-\mu)B^{-1}} \leq \delta/2 \quad &\text{and} \quad 8e\log(2d)L\eta B^{-1} \leq \delta/2\end{aligned}$$

We have:

$$\begin{aligned}\sqrt{2e\log(2d)L(\eta L-\mu)B^{-1}} \leq \delta/2 \quad &\Leftrightarrow \quad B \geq 8e\log(2d)L(L\eta-\mu)\delta^{-2}\\ 8e\log(2d)L\eta B^{-1} \leq \delta/2 \quad &\Leftrightarrow \quad B \geq 16e\log(2d)L\eta\delta^{-1}\end{aligned}$$

Putting together and taking $\delta = \delta/\alpha$ we arrive at the right condition on batch size B i.e

$$B \geq 8e\log(2d)\max\left\{L(\eta L-\mu)\delta^{-2}\alpha^2, (4L^2\delta^{-2}\alpha^2\eta^2)^{\frac{1}{2}}\right\}$$

☺ □

3.3.3 Convergence Bound for mini-SGD

Theorem 3.3.1

Consider mini-SGD applied to $\mu-$ strongly convex and $L-$ Smooth finite sum problem 2.2.4 whose sampling probabilities satisfy assumption 3.3.1. Fix parameter $\alpha^* = \frac{2}{L+\mu}$. Define the constant $\tilde{L} = \frac{L-\mu}{L+\mu}$ with $\mu = \sum_{i\in[n]} \mu_i$ and $L = \sum_{i\in[n]} L_i$.

For $k^* > 1$ choose

$$B \geq 16e\log(2d)\max\left\{\frac{L(\eta L-\mu)}{(L-\mu)^2}\frac{k^*}{\log(k^*)}, \frac{\sqrt{2}L\eta}{L-\mu}\left(\frac{k^*}{\log(k^*)}\right)^{1/2}\right\} \tag{3.57}$$

then if interpolation holds for x^* (2.2.1), the mini-SGD iterate satisfy

$$\mathbb{E}\,\|\underline{\boldsymbol{x}}_k - x^*\| \leq \tilde{L}^k \max\left\{(k^*)^{k/k^*}, d\right\}\|x_0 - x^*\| \tag{3.58}$$

Proof of Theorem 3.3.1.
Using Taylor approximation around the interpolation point x^* we analyse

$$\begin{aligned}
\underline{\boldsymbol{x}}_{k+1} - x^* &= \left(\mathbb{I}_{d\times d} - \alpha\frac{1}{B}\sum_{j\in[B]}\frac{1}{p_{\underline{\boldsymbol{s}}_j}}\nabla^2 f_{\underline{\boldsymbol{s}}_j}(x^*)\right)(\underline{\boldsymbol{x}}_k - x^*)\\
&=: \left(\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_k\right)(\underline{\boldsymbol{x}}_k - x^*)\\
&= \underbrace{\left(\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_k\right)\cdots\left(\mathbb{I}_{d\times d} - \alpha\underline{\boldsymbol{M}}_1\right)}_{=:\underline{\boldsymbol{Z}}_k}(x_0 - x^*)\\
&\Leftrightarrow\\
\mathbb{E}\|\underline{\boldsymbol{x}}_{k+1} - x^*\| &\leq \mathbb{E}\|\underline{\boldsymbol{Z}}_k\|\|x_0 - x^*\|
\end{aligned}$$

Define $m_i := \frac{L-\mu}{L+\mu} =: \tilde{L}$ and $\sigma_i := \delta/\tilde{L}$. By defining $\underline{\boldsymbol{Z}}_k$ by defining $\underline{\boldsymbol{Y}}_i := \mathbb{I}_{d\times d} - \alpha^*\underline{\boldsymbol{M}}_i \quad \forall i \in [n]$, use lemma 3.3.2 and lemma 3.3.3 to apply theorem 2.1.2 in order to control the expectation bound of the random product of matrices.

$$\nu = \sum_{i\in[k]}\sigma_i^2 = k\frac{\delta^2}{\tilde{L}^2} \qquad M = \prod_{i\in[k]} m_i = \tilde{L}^k$$

$$\begin{aligned}
\mathbb{E}\|\underline{\boldsymbol{Z}}_k\| &\leq \tilde{L}^k \exp\left\{\sqrt{\frac{2k\delta^2}{\tilde{L}^2}\max\{\frac{2k\delta^2}{\tilde{L}^2}, d\}}\right\}\\
&\leq \tilde{L}^k\max\{\exp\{\frac{2k\delta^2}{\tilde{L}^2}\}, d\}
\end{aligned}$$

where last equation holds conditioned on the fact that lemma 3.3.3 holds i.e for batch size large enough

$$B \geq 8e\log(2d)\max\left\{L(\eta L-\mu)\delta^{-2}\alpha^2, (4L^2\delta^{-2}\alpha^2\eta^2)^{\frac{1}{2}}\right\}$$

Set $\delta^2 := \tilde{L}^2 \log(k^*)/(2k^*)$ yield

$$\mathbb{E}\|\underline{\boldsymbol{Z}}_k\| \leq \tilde{L}^k \max\left\{(k^*)^{k/k^*}, d\right\}$$

given

$$B \geq 8e\log(2d)\max\left\{\underbrace{L(\eta L-\mu)\delta^{-2}\alpha^2}_{\text{Term 1}}, \left(\underbrace{4L^2\delta^{-2}\alpha^2\eta^2}_{\text{Term 2}}\right)^{\frac{1}{2}}\right\}$$

Simplifying Term 1 with $\alpha = \frac{1}{L+\mu}$ yield

$$L(\eta L-\mu)\delta^{-2}\alpha^2 = \frac{2L(\eta L-\mu)}{(L-\mu)^2}\frac{k^*}{\log(k^*)}$$

and for Term 2

$$4L^2\delta^{-2}\alpha^2\eta^2 = \frac{8L^2\eta^2}{(L-\mu)^2}\frac{k^*}{\log(k^*)}$$

yielding the minimum batch size

$$B \geq 16e\log(2d)\max\left\{\frac{L(\eta L-\mu)}{(L-\mu)^2}\frac{k^*}{\log(k^*)}, \frac{\sqrt{2}L\eta}{L-\mu}\left(\frac{k^*}{\log(k^*)}\right)^{1/2}\right\}$$

☺ □

Corollary 3.3.1 Fix $c \in (0,2)$. There exists parameter α such that for $\kappa := \frac{L}{\mu}$ sufficiently large, the mini-SGD iterates applied to finite sum problem 2.2.4 with interpolation at x^* in a region sufficiently close to quadratics converge in expected norm at least at linear rate

$$\mathbb{E}\|\underline{\boldsymbol{x}}_k - x^*\| \leq \mathcal{O}\left(\left(1-\frac{c}{\kappa}\right)^k\right) \quad \text{given} \quad B \geq \mathcal{O}\Big(\eta\log(d)\kappa\Big) \tag{3.59}$$

Proof of Corollary 3.3.1.
Taking the result from theorem 3.3.1 we have

$$\mathbb{E}\|\underline{\boldsymbol{x}}_k - x^*\| \leq Rate^k\|x_0 - x^*\|$$

where $Rate = Lk^*/\log k^*$. Following the identical steps as in the proof of corollary 3.1.1, i.e define $\frac{k^*}{\log(k^*)} := \frac{\kappa}{c}$ for some constant $c > 0$ we have

$$Rate^k \approx 1 - \tilde{c}/\kappa + \mathcal{O}(\kappa^{-2})$$

for $\kappa := \frac{L}{\mu}$.

Plug in the value κ/c in the batch size lower bound yield

$$B \geq 16e\log(2d)\max\left\{\underbrace{\frac{L(\eta L-\mu)}{(L-\mu)^2}\cdot\kappa/c}_{=:\text{Term 1}}, \underbrace{\frac{\sqrt{2}L\eta}{L-\mu}\cdot\sqrt{\kappa/c}}_{=:\text{Term 2}}\right\}$$

where for Term 1 :

$$\frac{L(\eta L-\mu)}{(L-\mu)^2}\cdot\kappa/c = \kappa\frac{\eta\kappa^2-\kappa}{c(\kappa-1)^2} \approx_{\kappa>>1} \mathcal{O}(\eta\kappa)$$

and for Term 2:

$$\frac{\sqrt{2}L\eta}{L-\mu}\cdot\sqrt{\kappa/c} = \frac{\sqrt{2}\kappa\eta}{\kappa-1}\sqrt{\kappa/c} \approx_{\kappa>>1} \mathcal{O}(\eta\sqrt{\kappa})$$

Hence

$$B \geq \mathcal{O}\left\{log(d)\eta\kappa\right\}$$

☺ □

3.4 High Probability Bound for mini-HBM on Quadratics

The cornerstone of our convergence analysis for the mini-HBM algorithm on consistent least squares problem [Problem 2.2.3] is the main theorem presented by [4]. For the sake of completeness, we first provide a standard proof analysis of heavy ball momentum on quadratic functions. We then supplement this with key lemmas from [4], culminating in the presentation of the main theorem. Our motivation for detailing these crucial lemmas extends beyond this chapter; they form the foundation for our subsequent work in sub-chapter [3.4.3], where we develop a high-probability bound extension to the main theorem, building upon the core proof strategy.

3.4.1 Standard Analysis of Heavy Ball Momentum on Quadratics

Assume the consistent least squares problem 2.2.3 with solution $Ax^* = b$. Then the update rule from the HBM algorithm (definition 2.2.4) satisfies

$$x_{k+1} = x_k - \alpha(A^TAx_k - A^TAx^*) + \beta(x_k - x_{k-1}) \tag{3.60}$$

$$\Leftrightarrow$$

$$x_{k+1} - x^* = \big((1+\beta)\mathbb{I}_{d\times d} - \alpha A^TA\big)(x_k - x^*) - \beta(x_{k-1} - x^*) \tag{3.61}$$

We can write the error transition in the following linear map

$$\begin{bmatrix} x_{k+1} - x^* \\ x_k - x^* \end{bmatrix} = \underbrace{\begin{bmatrix} (1+\beta)\mathbb{I}_{d\times d} - \alpha A^\top A & -\beta\mathbb{I}_{d\times d} \\ \mathbb{I}_{d\times d} & 0 \end{bmatrix}}_{T=T(\alpha,\beta)} \begin{bmatrix} x_k - x^* \\ x_{k-1} - x^* \end{bmatrix} \tag{3.62}$$

$$= T^k \begin{bmatrix} x_1 - x^* \\ x_0 - x^* \end{bmatrix} \tag{3.63}$$

Assuming $x_1 = x_0$ we have

$$\|x_k - x^*\| \leq \sqrt{2} \cdot \|T^k\| \cdot \|x_0 - x^*\| \tag{3.64}$$

Note that T is not symmetric implying $\|T^k\| \neq \|T\|^k$. To upper bound the norm of T^k it is common to use Gelfrand's formula [Appendix A.0.5] i.e for any matrix T

$$\rho(T) = \lim_{k\to\infty} \|T^k\|^{\frac{1}{k}} \tag{3.65}$$

which implies that for any ϵ it $\exists C_\epsilon$ such that

$$\|T^k\| \leq C_\epsilon(\rho(T) + \epsilon)^k \tag{3.66}$$

Applying to eq: 3.64

$$\|x_k - x_0\| \leq \sqrt{2}C_\epsilon\|x_0 - x^*\|\Big(\rho\big(T(\alpha,\beta)\big) + \epsilon\Big)^k \tag{3.67}$$

Hence the rate depends on the spectral radius of the error transition matrix T which can be upper bound by $\rho(T) = \max_i |z_i|$ where $z_i \in \lambda(T)$. To find the eigenvalues $\{z_i\}$ note that T is orthogonally similar to a block diagonal i.e

$$A^T A := U\Lambda U^T \quad \Lambda := diag\{\lambda_1, \dots, \lambda_d\} \quad \lambda_i \in \mathbb{C} \quad \forall i \in [d]$$

define the permutation matrix

$$\Pi_{i,j} = \begin{cases} i \quad \text{odd} \quad , \quad j = (i+1)/2 \\ i \quad \text{even} \quad , \quad j = n + i/2 \\ 0 \qquad\quad \text{otherwise.} \end{cases} \tag{3.68}$$

then

$$\Pi \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix}^T T \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} \Pi^T = \begin{bmatrix} T_1 & & & \\ & T_2 & & \\ & & \ddots & \\ & & & T_d \end{bmatrix} \quad \text{where} \quad T_i = \begin{bmatrix} 1+\beta-\alpha_i & -\beta \\ 1 & 0 \end{bmatrix} \tag{3.69}$$

where we read the eigenvalues of T

$$z_j^{\pm} := \frac{1}{2}\left(1+\beta-\alpha\lambda_j \pm \sqrt{(1+\beta-\alpha\lambda_i)^2-4\beta}\right) \tag{3.70}$$

which are maximised when

$$(1+\beta-\alpha\lambda_i)^2-4\beta < 0 \tag{3.71}$$

hence

$$\rho(T) \leq \max_{j\in[n]} |z_j^{\pm}| = \sqrt{\beta}$$

Now we wish to minimise w.r.t to β yielding

$$\min \sqrt{\beta} \quad \text{s.t} \quad (1+\beta-\alpha\lambda_i)^2-4\beta < 0 \tag{3.72}$$

Note the equivalence on the constraint

$$(1+\beta-\alpha\lambda_i)^2-4\beta < 0 \Leftrightarrow \tag{3.73}$$

$$\frac{(1-\sqrt{\beta})^2}{\lambda_{min}} < \alpha < \frac{(1+\sqrt{\beta})^2}{\lambda_{max}} \tag{3.74}$$

hence the minimum over the constraint region is attained by

$$\frac{(1-\sqrt{\beta})^2}{\lambda_{min}} = \alpha = \frac{(1+\sqrt{\beta})^2}{\lambda_{max}} \tag{3.75}$$

from which the best parameter in the sense minising the upper bound on $\rho(T)$ are

$$\sqrt{\alpha^*} = \frac{2}{\sqrt{\lambda_{max}}+\sqrt{\lambda_{min}}} \qquad \sqrt{\beta^*} = \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} \tag{3.76}$$

with $\kappa = \kappa(A^TA)$ condition number with respect to spectral norm.
Hence we have proved the asymptotic bound for the HBM algorithm with rate $\sqrt{\beta}$

In order to provide a **non-asymptotic** result, we need to adapt the analysis. The issue with parameters $\{\alpha^*, \beta^*\}$ is that the matrix T is defective hence diagonalisation does not exists. To circumvent this fact, we follow the proposition of [4] to replace the parameters $\{\alpha^*, \beta^*\}$ by a perturbed version parametrised by $\gamma \in (0, \lambda_{min})$

$$L = \lambda_{max} + \gamma \quad l = \lambda_{min} - \gamma \tag{3.77}$$

$$\sqrt{\alpha^*} := \frac{2}{\sqrt{L}+\sqrt{l}} \qquad \sqrt{\beta^*} = \frac{\sqrt{L/l}-1}{\sqrt{L/l}+1} \tag{3.78}$$

As the matrix $T(\alpha^*, \beta^*)$ has full rank, we define its diagonalisation

$$T = U_T C D (U_T C)^{-1} \tag{3.79}$$

with

$$\Pi \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix} =: U_T \quad T_j = C_j D_j C_j^{-1} \quad j \in [d] \quad C := diag\{C_1, \dots, C_d\} \tag{3.80}$$

Then

$$\begin{aligned} \|T^k\| &\leq \|U_T C\| \|(U_T C)^{-1}\| \|D^k\| \qquad (3.81) \\ &=: M(\alpha, \beta) \|D\|^k \qquad (3.82) \end{aligned}$$

where we defined the condition number $M(\alpha, \beta)$ of the eigenvector matrix UC . In [4], the authors show that under the current setting

$$\|D\| = \sqrt{\beta^*} \tag{3.83}$$

hence the non asymptotic error bound for HBM on consistent ls yield

$$\|x_k - x^*\| \leq \sqrt{2} \cdot \|x_0 - x^*\| \cdot M(\alpha^*, \beta^*) \cdot \left(\frac{\sqrt{L/l} - 1}{\sqrt{L/l} + 1} \right)^k \tag{3.84}$$

We can bound the condition number $M(\alpha^*, \beta^*)$ by [lemma 3.4.1]

Lemma 3.4.1

For any $\gamma \in \left(0, \lambda_{min}(A^T A)\right)$, the condition number of the eigenvector matrix $U_T C$ evaluated at (α^*, β^*) satisfies

$$M(\alpha^*, \beta^*) \leq \frac{4}{\alpha^* \cdot \sqrt{\gamma \cdot (\gamma + \lambda_{max} - \lambda_{min})}} \tag{3.85}$$

with $\lambda_{min} := \lambda_{min}(A^T A)$ and $\lambda_{max} = \lambda_{max}(A^T A)$

Proof of lemma 3.4.1.

The proof is provided in lemma 1 in the work [4] ☺ □

3.4.2 Expectation Bound for mini-HBM on Consistent Least Squares

Here we state the main theorem of [4] which shows that the same linear rate of convergence can be achieved by mini-HBM as the deterministic version HBM on the consistent least squares problem 2.2.3. The authors require assumption 3.1.1 to hold.

Theorem 3.4.1

Consider mini-HBM with parameters $\{\alpha^*, \beta^*\}$ [eq: 3.78] on consistent least squares [problem: 2.2.3], whose sampling probabilities satisfies assumption 3.1.1. Assume the solution $Ax^* = b$.
Then, for $\kappa(A^TA)$ sufficiently large, the average iterate error norm converges with linear rates

$$\mathbb{E}\|\underline{\boldsymbol{x}}_k - x^*\| \in \mathcal{O}\left(1 - \frac{1}{\sqrt{\kappa}}\right) \tag{3.86}$$

given

$$B \geq \mathcal{O}\left(\eta d \log(d) \overline{\kappa} \sqrt{\kappa}\right) \tag{3.87}$$

Proof of theorem 3.4.1.

The proof can be found in [4] [Theorem 4] together with [corollary 1] ☺ □

3.4.3 High Probability Bound for mini-HBM on Consistent Least Squares

In this section, we begin by presenting a proof sketch to provide clarity to the reader. We then state the main theorem that provides a high-probability bound on the norm of the error rate and follow it with a more rigorous proof. Finally, we present a corollary derived from the theorem.
Assume the consistent least squares problem 2.2.3 with solution $Ax^* = b$ and the stochastic optimisation algorithm mini-HBM [def: 2.3.2] with sampling probabilities satisfying assumption 3.1.1. Define the random error at iteration k by

$$\underline{\boldsymbol{e}}_k := \underline{\boldsymbol{x}}_k - x^* \tag{3.88}$$

the stochastic gradient as

$$\nabla f(\underline{\boldsymbol{x}}_k, \underline{\boldsymbol{\xi}}) = \underline{\boldsymbol{M}}_k \cdot \underline{\boldsymbol{e}}_k \quad \underline{\boldsymbol{M}}_k := \frac{1}{B} \sum_{i \in [B]} \frac{1}{p_{\underline{\boldsymbol{s}}_i}} a_{\underline{\boldsymbol{s}}_i} a_{\underline{\boldsymbol{s}}_i}^T \tag{3.89}$$

Then following the same procedure as in the deterministic case [section 3.4.1], the transition error map satisfies

$$\begin{bmatrix} \underline{\boldsymbol{e}}_{k+1} \\ \underline{\boldsymbol{e}}_k \end{bmatrix} = \underbrace{\begin{bmatrix} (1+\beta)\mathbb{I}_{d\times d} - \alpha \underline{\boldsymbol{M}}_k & -\beta \mathbb{I}_{d\times d} \\ \mathbb{I}_{d\times d} & 0 \end{bmatrix}}_{=:\underline{\boldsymbol{Y}}_k \in \mathbb{R}^{2d\times 2d}} \begin{bmatrix} \underline{\boldsymbol{e}}_k \\ \underline{\boldsymbol{e}}_{k-1} \end{bmatrix} \tag{3.90}$$

Assuming $x_0 = x_1$ we have

$$\|\underline{\boldsymbol{e}}_k\| \leq \sqrt{2} \cdot \|e_0\| \cdot \|\underline{\boldsymbol{Y}}_k \cdots \underline{\boldsymbol{Y}}_1\| \tag{3.91}$$

By controlling the tail of the spectral norm of the product of random matrices $\underline{\boldsymbol{Y}}_k \cdots \underline{\boldsymbol{Y}}_1$ [Theorem 2.1.3] we aim at a result of the form

$$\mathbb{P}\left\{\|\underline{\boldsymbol{e}}_k\| \leq CRate^k\right\} \geq 1 - \delta \qquad C > 0, \delta \in (0,1) \tag{3.92}$$

3.4.4 Main Result

Theorem 3.4.2

Consider mini-HBM applied to consistent least squares problem 2.2.3 whose sampling probabilities satisfy assumption 3.1.1. Assume the solution $Ax^* = b$. Fix parameters $\{\alpha = \alpha^*, \beta = \beta^*\}$ as in [eq: 3.78]. Define the condition number of the eigenvector matrix of T as in [eq: 3.82] by $M(\alpha^*, \beta^*)$.
For any $k^* > 1$ assume

$$B \geq 4 \cdot \eta \cdot \|A\|_F^2 \cdot \Big(\text{Term } 1 + \text{Term } 2\Big) \cdot \Big(\log(1/\tilde{\delta}) + \log(2d)\Big) \tag{3.93}$$

with probability at least $1 - k\tilde{\delta}$

where

$$\text{Term1} := \|A\|^2 \cdot M(\alpha^*, \beta^*)^2 \cdot (\alpha^*)^2 \cdot \frac{k^*}{\log(k^*) \cdot \beta^*} \tag{3.94}$$

$$\text{Term2} := \left(4/9 \cdot M(\alpha^*, \beta^*)^2 \cdot (\alpha^*)^2 \cdot \frac{2k^*}{\log(k^*) \cdot \beta^*}\right)^{1/2} \tag{3.95}$$

Then with probability at least $1 - \delta$

$$\|\underline{\boldsymbol{x}}_k - x^*\| \leq \sqrt{2} \cdot M(\alpha^*, \beta^*) \cdot \|x_0 - x^*\| \cdot (\sqrt{\beta^*})^k \cdot \exp\left(\sqrt{\frac{k \cdot \log(k^*)}{k^*}} \cdot \log(2d/\delta)\right) \tag{3.96}$$

for all $k \in \mathbb{N}$ satisfying

$$\log(d/\delta)\frac{\log(k^*)}{k^*} \geq k \tag{3.97}$$

Proof of Theorem 3.4.2.
Define

$$\|\underline{\boldsymbol{Y}}_k \cdots \underline{\boldsymbol{Y}}_1\| =: \|(U_T C)\underline{\boldsymbol{X}}_k \cdots \underline{\boldsymbol{X}}_1 (U_T C)^{-1}\|$$

where $U_T C$ is the eigenvector matrix of the deterministic transition error matrix T [eq: 3.79]. Then the random iterates error norm yield

$$\begin{aligned} \|\underline{\boldsymbol{e}}_k\| &\leq \|U_T C\| \|(U_T C)^{-1}\| \sqrt{2} \|e_0\| \|\underline{\boldsymbol{X}}_k \cdots \underline{\boldsymbol{X}}_1\| \\ &= M(\alpha^*, \beta^*) \sqrt{2} \|e_0\| \| \underbrace{\underline{\boldsymbol{X}}_k \cdots \underline{\boldsymbol{X}}_1}_{=:\underline{\boldsymbol{Z}}_k} \| \end{aligned}$$

By theorem 3.2.1 we can control the tail of the spectral norm of $\underline{\boldsymbol{Z}}_k$ if the following condition holds:

Condition A :

$$\|\mathbb{E}\underline{\boldsymbol{X}}_i\| \leq m_i \quad \forall i \in [k]$$

where we notice (eq: 3.79)

$$\|\mathbb{E}\underline{\boldsymbol{X}}_i\| = \|(U_T C)^{-1}\mathbb{E}\underline{\boldsymbol{Y}}_i(U_T C)\| = \|(U_T C)^{-1}T(U_T C)\| = \|D\|$$

Then by eq: 3.83

$$\|D\| = \sqrt{\beta^*}$$

Define

$$m_i := \sqrt{\beta^*} \quad \forall i \in [k] \implies \text{Condition A True}$$

Condition B :

$$\|\underline{\boldsymbol{X}}_i - \mathbb{E}\underline{\boldsymbol{X}}_i\| \leq \sigma_i m_i \quad \text{a.s} \quad \forall i \in [k]$$

Notice the following:

$$\|\underline{\boldsymbol{X}}_i - \mathbb{E}\underline{\boldsymbol{X}}_i\| \leq M(\alpha^*, \beta^*)\|\underline{\boldsymbol{Y}}_i - \mathbb{E}\underline{\boldsymbol{Y}}_i\| = M(\alpha^*, \beta^*)\|\underline{\boldsymbol{Y}}_i - T\|$$

notice

$$\underline{\boldsymbol{Y}}_i - T = \sum_{j \in \underline{\boldsymbol{S}}_i} \frac{\alpha^*}{B} \begin{bmatrix} -p_j^{-1}a_j a_j^T + A^T A & 0 \\ 0 & 0 \end{bmatrix} =: \alpha^* \sum_{j \in S_i} \begin{bmatrix} \underline{\boldsymbol{W}}_j & 0 \\ 0 & 0 \end{bmatrix}$$

implying

$$\|\underline{\boldsymbol{Y}}_i - T\| = \alpha^* \|\underline{\boldsymbol{W}}\| \quad \text{with} \quad \underline{\boldsymbol{W}} := \sum_{j \in S_i} \underline{\boldsymbol{W}}_j$$

$$\implies$$

$$\|\underline{\boldsymbol{X}}_i - \mathbb{E}\underline{\boldsymbol{X}}_i\| \leq \alpha^* M(\alpha^*, \beta^*)\|\underline{\boldsymbol{W}}\|$$

Applying Lemma 3.2.1 with $t = t_1/(\alpha^* M(\alpha^*, \beta^*))$ yield with probability at least $1 - \tilde{\delta}$

$$\|\underline{\boldsymbol{X}}_i - \mathbb{E}\underline{\boldsymbol{X}}_i\| \leq t_1$$

given

$$B \geq 2\eta\|A\|_F^2 \left(\|A\|^2 M(\alpha^*, \beta^*)^2 (\alpha^*)^2 t_1^{-2} + \frac{2}{3} M(\alpha^*, \beta^*)(\alpha^*) t_1^{-1} \right) \cdot \log\left(\frac{2d}{\tilde{\delta}}\right)$$

Last result holds uniformly over all k with probability at least $1 - k\tilde{\delta}$.
Hence define

$$\sigma_i := t_1/\sqrt{\beta^*} \implies \text{Condition B True w.p at least } 1 - k\tilde{\delta}$$
$$\text{given} \quad B \geq 2\eta\|A\|_F^2\left(\|A\|^2 M(\alpha^*,\beta^*)^2(\alpha^*)^2 t_1^{-2} + \frac{2}{3}M(\alpha^*,\beta^*)(\alpha^*)t_1^{-1}\right) \cdot \log\left(\frac{2d}{\tilde{\delta}}\right)$$

From Condition A and B we have

$$M = \left(\sqrt{\beta^*}\right)^k \quad \nu = \sum_{i\in[k]} \sigma_i^2 = k\frac{t_1^2}{\beta^*}$$

Choose

$$t_1^2 := \beta^* \frac{\log(k^*)}{2k^*} \implies 2\nu = k\frac{\log(k^*)}{k^*}$$

Apply Theorem 3.2.1 to $\|\underline{\boldsymbol{Z}}_k\|$

$$\mathbb{P}\left\{\|\underline{\boldsymbol{Z}}_k\| \geq t_2 M\right\} \leq 2d\exp\left(\frac{-\log(t_2)^2 k^*}{k\log(k^*)}\right) \quad \text{given} \quad \log(t_2) \geq \frac{k\log(k^*)}{k^*}$$

Setting the right handside to δ yield

$$t_2 = \exp\left(\sqrt{\log(2d/\delta)\frac{\log(k^*)k}{k^*}}\right)$$

Recall from eq: 3.91 and denote $\|e_0\| =: D_0$ for clarity:

$$\begin{aligned}\|\underline{\boldsymbol{e}}_k\| &\leq \sqrt{2}\|e_0\|\|\underline{\boldsymbol{Y}}_k\cdots\underline{\boldsymbol{Y}}_1\| \\ &= \underbrace{\sqrt{2}D_0 M(\alpha^*,\beta^*)}_{=:C}\|\underline{\boldsymbol{X}}_k\cdots\underline{\boldsymbol{X}}_1\| \\ &\Leftrightarrow \\ \|\underline{\boldsymbol{e}}_k\| &\leq C\|\underline{\boldsymbol{X}}_k\cdots\underline{\boldsymbol{X}}_1\|\end{aligned}$$

Then the following events are inclusive

$$E_1 := \left\{\|\underline{\boldsymbol{e}}_k\| \geq Ct_2\right\} \implies E_2 := \left\{\|\underline{\boldsymbol{X}}_k\cdots\underline{\boldsymbol{X}}_1\| \geq t_2\right\}$$

implying

$$\mathbb{P}\left\{E_1\right\} \leq \mathbb{P}\left\{E_2\right\} \leq \delta$$

which finally yield

$$\mathbb{P}\left\{\|\underline{\boldsymbol{e}}_k\| \geq C\cdot\exp\left(\sqrt{\log(2d/\delta)\frac{\log(k^*)k}{k^*}}\right)\right\} \leq \delta$$

To obtain the desired bound for the batch size it remains to plug $t_1^2 = \beta^*\frac{\log(k^*)}{2k^*}$ in

$$B \geq 2\eta\|A\|_F^2\left(\|A\|^2 M(\alpha^*,\beta^*)^2(\alpha^*)^2 t_1^{-2} + \frac{2}{3}M(\alpha^*,\beta^*)(\alpha^*)t_1^{-1}\right)\cdot\log\left(\frac{2d}{\tilde{\delta}}\right)$$

☺ □

3.4.5 Interpretation of Theorem and Limitations

In this section, we refine Theorem 3.4.2 into a more interpretable form, focusing on the batch size condition. We demonstrate that under specific conditions, we can control the error rate with high probability at an information-theoretically optimal rate. Additionally, we conclude the section by discussing some non-asymptotic limitations of our results.

Corollary 3.4.1 Under Assumptions of Theorem 3.2.1
Fix constant $c_1 \in (0,1)$ and c_2 such that the following holds

$$c_2 \in \left(\frac{2\sqrt{1-c_1}}{\log(d/\delta)}\ ,\ \frac{2\sqrt{\kappa}-2\sqrt{1-c_1}}{\log(d/\delta)}\right) \tag{3.98}$$

Define $C := 2\sqrt{1-c_1} - c_2\log(d/\delta)$.
Then for all $\kappa(A)$ sufficiently large, the error norm of mini-HBM on consistent least squares 2.2.3 converges with probability at least $1-\delta$ at a linear rate, i.e,

$$\mathbb{P}\left\{\|\underline{\boldsymbol{x}_k} - x^*\| \leq \sqrt{2}\cdot M(\alpha^*,\beta^*)\cdot\|x_0 - x^*\|\cdot\left(1-\frac{C}{\sqrt{\kappa}}\right)^k\right\} \geq 1-\delta \tag{3.99}$$

for

$$k \leq \log(2d/\delta)\frac{\sqrt{\kappa}}{c_2}$$

Provided that with probability at least $1-k\tilde{\delta}$,

$$B \geq \mathcal{O}\left(\eta d\log(d/\tilde{\delta})\overline{\kappa}\sqrt{\kappa}\right) \tag{3.100}$$

Proof of Corollary 3.4.1.

Assume $\gamma := c_1\lambda_{min}(A^TA)$ for $c_1 \in (0,1)$ which implies

$$\frac{L}{l} = \frac{\kappa}{1-c_1} + \frac{c_1}{1-c_1}$$

hence

$$\sqrt{\beta^*} = \frac{\sqrt{L/l}-1}{\sqrt{L/l}+1} = 1 - \frac{2\sqrt{1-c_1}}{\sqrt{\kappa+c_1}+\sqrt{1-c_1}} \approx_{\kappa>>1} 1 - \frac{2\sqrt{1-c_1}}{\sqrt{\kappa}}$$

From Theorem 3.4.2 we notice

$$\begin{aligned}(\sqrt{\beta^*})^k \exp\left(\sqrt{\frac{k\cdot\log(k^*)}{k^*}}\cdot\log(d/\delta)\right) &\leq (\sqrt{\beta^*})^k \exp\left(\frac{k\cdot\log(k^*)}{k^*}\cdot\log(d/\delta)\right)\\ &= \Bigg(\underbrace{\sqrt{\beta^*}\exp\left(\frac{c_2}{\sqrt{\kappa}}\cdot\log(d/\delta)\right)}_{=:RATE}\Bigg)^k\end{aligned}$$

where we defined

$$\frac{k^*}{\log(k^*)} := \frac{\sqrt{\kappa}}{c_2} \quad \text{for some} \quad c_2 > 0$$

Define p sucht that

$$\begin{aligned}\left(\sqrt{\beta^*}\right)^{1-p} &= \sqrt{\beta^*}\exp\left(\frac{c_2}{\sqrt{\kappa}}\cdot\log(d/\delta)\right)\\ &\Leftrightarrow\\ 1-p &= 1 - \frac{c_2\log(d/\delta)}{\sqrt{\kappa}}\cdot\frac{1}{\log(1/\sqrt{\beta^*})}\end{aligned}$$

Note:

$$\log(1/\sqrt{\beta^*}) = -\log(\sqrt{\beta}) \approx -\log(1-\frac{2\sqrt{1-c_1}}{\sqrt{\kappa}}) \approx \frac{2\sqrt{1-c_1}}{\sqrt{\kappa}}$$

hence

$$1-p = 1 - \frac{c_2\log(d/\delta)}{2\sqrt{1-c_1}} + \mathcal{O}\left(\kappa^{-1/2}\right)$$

$$\Leftrightarrow$$

$$\begin{aligned}(\sqrt{\beta^*})^{1-p} &= \left(1-\frac{2\sqrt{1-c_1}}{\sqrt{\kappa}}\right)^{1-\frac{c_2\log(d/\delta)}{2\sqrt{1-c_1}}+\mathcal{O}\left(\kappa^{-1/2}\right)}\\ &\approx \exp\left(-\frac{2\sqrt{1-c_1}-c_2\log(d/\delta)}{\sqrt{\kappa}}\right)\\ &= \left(1-\frac{2\sqrt{1-c_1}-c_2\log(d/\delta)}{\sqrt{\kappa}}\right)\\ &:= \left(1-\frac{C}{\sqrt{\kappa}}\right)\end{aligned}$$

for some $C = 2\sqrt{1-c_1} - c_2\log(d/\delta)$. Hence we have convergence if and only if

$$|1 - \frac{C}{\sqrt{\kappa}}| < 1 \quad \Leftrightarrow \quad c_2 \in \left(\quad \frac{2\sqrt{1-c_1}}{\log(d/\delta)} \quad , \quad \frac{2\sqrt{\kappa} - 2\sqrt{1-c_1}}{\log(d/\delta)} \quad \right)$$

Recall that $c_1 \in (0,1)$, we proved the existence of a linear rate of convergence of order $1 - \mathcal{O}(\sqrt{\kappa})$.

The condition (artefact form 2.1.3) on t_2 yield in this setting

$$\begin{aligned} \log(t_2) \geq 2\nu \quad &\Leftrightarrow \\ \sqrt{\frac{k\log(k^*)}{k^*}\log(2d/\delta)} \geq k\frac{\log(k^*)}{k^*} \quad &\Leftrightarrow \\ k \leq \frac{k^*}{\log(k^*)}\log(2d/\delta) = \log(2d/\delta)\frac{\sqrt{\kappa}}{c_2} \end{aligned}$$

Next to simplify the notation we drop the star on M, α, β and we plug in the constant choice

$$\frac{k^*}{\log(k^*)} := \frac{\sqrt{\kappa}}{c_2}$$

into the batch size bound

$$\begin{aligned} B &\geq 2\eta\|A\|_F^2 \left(\|A\|^2 M^2\alpha^2 \frac{k^*}{\log(k^*)}\frac{1}{\beta} + \left(\frac{8}{9}M^2\alpha^2\frac{k^*}{\log(k^*)}\frac{1}{\beta}\right)^{1/2} \right) \cdot \log(d/\tilde{\delta}) \\ &= 2\left(\underbrace{\eta d\bar{\kappa}\kappa\lambda_{min}^2 M^2\alpha^2\frac{\sqrt{\kappa}}{\beta c_2}}_{=:\text{Term 1}} + \underbrace{\left(\frac{8}{9}M^2\alpha^2\lambda_{min}^2\frac{\sqrt{\kappa}}{\beta c_2}\right)^{1/2}}_{=:\text{Term 2}} \right) \cdot \log(d/\tilde{\delta}) \end{aligned}$$

Note the term $(M\alpha\lambda_{min})^2$ appearing in both Term1 and Term 2. To bound this, we use Lemma 3.4.1

$$(M\alpha\lambda_{min})^2 \leq \frac{4\lambda_{min}^2}{\gamma(\gamma + \lambda_{max} - \lambda_{min})} \leq \frac{\lambda_{min}}{\gamma}\frac{4\lambda_{min}}{\lambda_{max} - \lambda_{min}} \in \mathcal{O}\left(\kappa^{-1}\right) \implies \kappa(M\alpha\lambda_{min}) \in \mathcal{O}\left(1\right)$$

Hence we have for Term 1 as $1/c_2 \in \mathcal{O}\left(\log(d)\right)$

$$\text{Term 1} \in \mathcal{O}\left(\eta d\log(d)\bar{\kappa}\sqrt{\kappa}\right)$$

and for Term 2

$$\text{Term 2} \in \mathcal{O}\left(\eta d\log(d)\bar{\kappa}\kappa^{1/4}\right) \in \mathcal{O}\left(\eta d\log(d)\bar{\kappa}\sqrt{\kappa}\right)$$

Finally

$$B \in \mathcal{O}\left(\eta d\log(d/\tilde{\delta})\bar{\kappa}\sqrt{\kappa}\right)$$

☺ □

Non-Asymptotic Limitations of Corollary 3.4.1

As in Section 3.2.1, we encounter a non-asymptotic limitation on the number of iterations for which Corollary 3.4.1 holds:

$$k \leq \log(2d/\delta)\frac{\sqrt{\kappa}}{c_2} \tag{3.101}$$

The constant c_2, which behaves as $\mathcal{O}(1/\log(d/\delta))$, introduces a critical trade-off. Minimizing the constant $C = 2\sqrt{1-c_1} - c_2\log(d/\delta)$ for faster convergence requires maximizing c_2. However, to extend the result's validity to a larger number of iterations k, we need to minimize c_2. Similar limitations arise in the bound on the batch size, stemming from the proof technique [Theorem 3.2.1]. For Theorem 3.4.2 to hold, we require a uniform bound over all $i \in [k]$:

$$|\underline{\boldsymbol{X}}_i - \mathbb{E}\underline{\boldsymbol{X}}_i| \leq \sigma_i m_i \quad \text{w.p. at least } 1-\tilde{\delta} \quad \text{(see proof of [Theorem 3.4.2])} \tag{3.102}$$

This must hold with probability at least $1 - k\tilde{\delta}$, creating an inverse relationship between iterations k and probability $\tilde{\delta}$. Consequently, there exists a non-asymptotic bound on k for the batch size to remain meaningful (i.e., less than the number of rows n):

$$k \lesssim \exp\left(\frac{n}{\eta\bar{\kappa}\sqrt{\kappa}d}\right)d^{-1} \Leftrightarrow B \lesssim n \tag{3.103}$$

In the large-scale system regime (i.e., $n \gg d$), this limitation should not be significantly constraining.

CHAPTER 4 FUTURE DIRECTIONS

4.1 Inconsistent Systems

The results we have proven all apply to consistent systems. Following the work of Bollapragada et al. [4], it is possible to extend all our findings to inconsistent systems as well. This extension would broaden the applicability of our results to a wider range of practical scenarios where exact solutions may not exist.

4.2 Towards Unification

The core of our proof technique lies in applying concentration results for products of matrices to stochastic optimization theory, particularly for linear update operators. This approach, while novel, has been largely unexplored due to the historical development of these fields. Concentration results for matrix products have primarily emerged in recent years, whereas the study of stochastic iterative methods with linear updates has a longer history. This new perspective, analyzing through the lens of concentration, deviates from classical proof techniques and opens up significant avenues for exploration. A particularly promising direction would be to extend this approach to the unified framework of stochastic methods for solving linear systems, as presented by Gower and Richtárik [16]. Their work provides a general framework for batch versions of iterative algorithms, including:

- Block Kaczmarz
- Randomized Newton
- Randomized Coordinate Descent with Batching

In their framework, the stochastic matrix for iterate update is presented as:

$$\underline{\boldsymbol{Z}} := A^T \underline{\boldsymbol{S}}(\underline{\boldsymbol{S}}^T A B^{-1} A^T \underline{\boldsymbol{S}})^+ \underline{\boldsymbol{S}} A \tag{4.1}$$

Taking an analytic viewpoint of solving linear systems yields the random fixed point iteration:

$$\underline{\boldsymbol{x}}_{k+1} - x^* = (I - B^{-1}\underline{\boldsymbol{Z}}_k)(\underline{\boldsymbol{x}}_k - x^*) \tag{4.2}$$

where it is assumed that $Ax^* = b$. Analyzing the following product in a manner similar to the work presented in this thesis could potentially reveal unknown results:

$$\underline{\boldsymbol{M}}_n := (I - B^{-1}\underline{\boldsymbol{Z}}_n) \cdots (I - B^{-1}\underline{\boldsymbol{Z}}_0) \tag{4.3}$$

This approach, applied within a unification framework, could yield new insights not just for one algorithm, but for several, further advancing the theory of mini-batching. Such an extension would complement and expand upon the results presented in this thesis, providing a more comprehensive understanding of stochastic optimization methods in various contexts.

ACRONYMS

This document is incomplete. The external file associated with the glossary 'acronym' (which should be called `main.acr`) hasn't been created.

Check the contents of the file `main.acn`. If it's empty, that means you haven't indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can't be generated. If the file isn't empty, the document build process hasn't been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`. For example:

  ```
  \usepackage[automake]{glossaries-extra}
  ```

- Run the external (Lua) application:

  ```
  makeglossaries-lite.lua "main"
  ```

- Run the external (Perl) application:

  ```
  makeglossaries "main"
  ```

Then rerun LaTeX on this document.

This message will be removed once the problem has been fixed.

BIBLIOGRAPHY

[1] Ron Aharoni and Yair Censor. "Block-iterative projection methods for parallel computation of solutions to convex feasibility problems." In: *Linear Algebra and Its Applications* 120 (1989), pp. 165–175.

[2] Rudolf Ahlswede and Andreas Winter. "Strong converse for identification via quantum channels." In: *IEEE Transactions on Information Theory* 48.3 (2002), pp. 569–579.

[3] Richard Bellman. "Limit theorems for non-commutative operations. I." In: (1954).

[4] Raghu Bollapragada, Tyler Chen, and Rachel Ward. "On the fast convergence of minibatch heavy ball momentum." In: *arXiv preprint arXiv:2206.07553* (2022).

[5] Léon Bottou. "Online algorithms and stochastic approximations." In: *Online learning in neural networks* (1998).

[6] Léon Bottou, Frank E Curtis, and Jorge Nocedal. "Optimization methods for large-scale machine learning." In: *SIAM review* 60.2 (2018), pp. 223–311.

[7] Bugra Can, Mert Gurbuzbalaban, and Lingjiong Zhu. "Accelerated linear convergence of stochastic momentum methods in wasserstein distances." In: *International Conference on Machine Learning*. PMLR. 2019, pp. 891–901.

[8] Yair Censor, Dan Gordon, and Rachel Gordon. "Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems." In: *Parallel computing* 27.6 (2001), pp. 777–808.

[9] Alain Durmus et al. "Finite-Time High-Probability Bounds for Polyak–Ruppert Averaged Iterates of Linear Stochastic Approximation." In: *Mathematics of Operations Research* (2024).

[10] Alain Durmus et al. "Tight high probability bounds for linear stochastic approximation with fixed stepsize." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 30063–30074.

[11] Paulus Petrus Bernardus Eggermont, Gabor T Herman, and Arnold Lent. "Iterative algorithms for large partitioned linear systems, with applications to image reconstruction." In: *Linear algebra and its applications* 40 (1981), pp. 37–67.

[12] Tommy Elfving. "Block-iterative methods for consistent and inconsistent linear equations." In: *Numerische Mathematik* 35 (1980), pp. 1–12.

[13] Ludwig Fahrmeir et al. *Regression models*. Springer, 2013.

[14] Inês A Ferreira, Juan A Acebrón, and José Monteiro. "Survey of a Class of Iterative Row-Action Methods: The Kaczmarz Method." In: *arXiv preprint arXiv:2401.02842* (2024).

[15] Guillaume Garrigos and Robert M Gower. "Handbook of convergence theorems for (stochastic) gradient methods." In: *arXiv preprint arXiv:2301.11235* (2023).

[16] Robert M Gower and Peter Richtárik. "Randomized iterative methods for linear systems." In: *SIAM Journal on Matrix Analysis and Applications* 36.4 (2015), pp. 1660–1690.

[17] Robert M Gower et al. "Variance-reduced methods for machine learning." In: *Proceedings of the IEEE* 108.11 (2020), pp. 1968–1983.

[18] Robert Mansel Gower et al. "SGD: General analysis and improved rates." In: *International conference on machine learning*. PMLR. 2019, pp. 5200–5209.

[19] Mert Gurbuzbalaban, Umut Simsekli, and Lingjiong Zhu. "The heavy-tail phenomenon in SGD." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 3964–3975.

[20] Amelia Henriksen and Rachel Ward. "Concentration inequalities for random matrix products." In: *Linear Algebra and its Applications* 594 (2020), pp. 81–94.

[21] Magnus Rudolph Hestenes, Eduard Stiefel, et al. *Methods of conjugate gradients for solving linear systems*. Vol. 49. 1. NBS Washington, DC, 1952.

[22] De Huang et al. "Matrix concentration for products." In: *Foundations of Computational Mathematics* 22.6 (2022), pp. 1767–1799.

[23] Stefan Karczmarz. "Angenaherte auflosung von systemen linearer glei-chungen." In: *Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat.* (1937), pp. 355–357.

[24] Robert W Keener. *Theoretical statistics: Topics for a core course*. Springer Science & Business Media, 2010.

[25] Michel Ledoux. *The concentration of measure phenomenon*. 89. American Mathematical Soc., 2001.

[26] Kiwon Lee et al. "Trajectory of mini-batch momentum: batch size saturation and convergence in high dimensions." In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 36944–36957.

[27] Jörg Liesen and Zdenek Strakos. *Krylov subspace methods: principles and analysis*. Numerical Mathematics and Scie, 2013.

[28] Ji Liu and Stephen Wright. "An accelerated randomized Kaczmarz algorithm." In: *Mathematics of Computation* 85.297 (2016), pp. 153–178.

[29] Nicolas Loizou and Peter Richtárik. "Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods." In: *Computational Optimization and Applications* 77.3 (2020), pp. 653–710.

[30] Siyuan Ma, Raef Bassily, and Mikhail Belkin. "The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning." In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3325–3334.

[31] Jacob D Moorman et al. "Randomized Kaczmarz with averaging." In: *BIT Numerical Mathematics* 61 (2021), pp. 337–359.

[32] Deanna Needell and Joel A Tropp. "Paved with good intentions: analysis of a randomized block Kaczmarz method." In: *Linear Algebra and its Applications* 441 (2014), pp. 199–221.

[33] Deanna Needell, Rachel Ward, and Nati Srebro. "Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm." In: *Advances in neural information processing systems* 27 (2014).

[34] Yurii Nesterov et al. *Lectures on convex optimization*. Vol. 137. Springer, 2018.

[35] Roberto Imbuzeiro Oliveira. "Concentration of the adjacency matrix and of the Laplacian in random graphs with independent edges." In: *arXiv preprint arXiv:0911.0600* (2009).

[36] Iosif Pinelis. "Optimum bounds for the distributions of martingales in Banach spaces." In: *arXiv preprint arXiv:1208.2200* (2012).

[37] Boris T Polyak. "Some methods of speeding up the convergence of iteration methods." In: *Ussr computational mathematics and mathematical physics* 4.5 (1964), pp. 1–17.

[38] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*. Vol. 37. Springer Science & Business Media, 2006.

[39] Roland Speicher. "Free probability theory." In: *arXiv preprint arXiv:0911.0087* (2009).

[40] Thomas Strohmer and Roman Vershynin. "A randomized Kaczmarz algorithm with exponential convergence." In: *Journal of Fourier Analysis and Applications* 15.2 (2009), pp. 262–278.

[41] Joel A Tropp et al. "An introduction to matrix concentration inequalities." In: *Foundations and Trends® in Machine Learning* 8.1-2 (2015), pp. 1–230.

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

[43] Dan Voiculescu. "Symmetries of some reduced free product C-algebras, Operator Algebras and Their Connection with Topology and Ergodic Theory (Lecture Notes in Mathematics 1132)." In: *COMPOSITION OF SUBFACTORS* 383 (1985).

[44] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press, 2019.

[45] Stephen J Wright and Benjamin Recht. *Optimization for data analysis*. Cambridge University Press, 2022.

[46] Yangyang Xu and Wotao Yin. "Block stochastic gradient iteration for convex and nonconvex optimization." In: *SIAM Journal on Optimization* 25.3 (2015), pp. 1686–1716.

[47] Lin Zhu, Yuan Lei, and Jiaxin Xie. "A greedy randomized average block projection method for linear feasibility problems." In: *arXiv preprint arXiv:2211.10331* (2022).

APPENDIX

A

APPENDIX

A.0.1 Facts on Convergence

Definition A.0.1 Speed of Convergence

A sequence $\{x_k\}_{k\in\mathbb{N}} \in \mathbb{R}^n$ converges **linearly** to $x^* \in \mathbb{R}^n$ if

$$\exists 0 < \tilde{L} < 1 : \|x_{k+1} - x^*\| \leq \tilde{L}\|x_k - x^*\| \quad \forall k \in \mathbb{N}_0$$

the **rate of convergence** is defined as the least upper bound for $\tilde{L}$

$$L := \sup_{k\in\mathbb{N}_0} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|}$$

Lemma A.0.1

Consider a sequence $\{a_k\}_{k\in\mathbb{N}} \in \mathbb{R}_+$ satisfying

$$a_k \leq L^k \alpha_0 \tag{A.1}$$

for $L \in [0,1[$ and $\alpha_0 \in \mathbb{R}_+$. Then for a given $\epsilon \in]0,1[$ we have

$$k \geq \frac{1}{1-L} log(\frac{\alpha_0}{\epsilon}) \implies \alpha_k \leq \epsilon \tag{A.2}$$

Proof. A.0.1

$$\alpha_k \leq \epsilon \Leftrightarrow log\frac{\alpha_0}{\alpha_k} \geq log\frac{\alpha_0}{\epsilon}$$

from the eq: A.1 we have

$$log\frac{\alpha_0}{\alpha_k} \geq klog\frac{1}{L} \geq k(1-L) \geq \frac{1}{1-L}log\frac{\alpha_0}{\epsilon}(1-L) = log\frac{\alpha_0}{\epsilon} \Leftrightarrow \alpha_k \leq \epsilon$$

☺ □

A.0.2 Theory: μ-Strong Convexity and Smoothness

We introduce several definitions and lemmas from optimisation theory that may be unfamiliar to readers from other fields. While we assume the reader has a basic understanding of fundamental concepts like convexity, which are common across various disciplines, we aim to provide a foundation for the more specialised ideas used in this work.

Definition A.0.2 μ-Strong Convexity

Let $f : dom(f) \subseteq \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and $\mu > 0$ be convex and differentiable over $X \subseteq dom(f)$. We say that f is $\mu-$ strongly convex over X if

$$f(y) \geq f(x) + \nabla f(x)^T(y-x) + \frac{\mu}{2}\|x-y\|^2 \quad , \quad \forall x, y \in X \tag{A.3}$$

If $X = dom(f)$ then f is simply called strongly convex.

Lemma A.0.2

Suppose that $dom(f)$ is open and convex, and that $f : dom(f) \to \mathbb{R}$ is differentiable. Let $\mu \in \mathbb{R}^+$. Then the following two statements are equivalent.

1. f is $\mu-$ strongly convex
2. g defined by $g(x) = f(x) - \frac{\mu}{2}x^Tx$ is convex over $dom(g) := dom(f)$

Proof of lemma A.0.2.

Section 2 in [15] ☺ □

Definition A.0.3 ***L*-Smoothness**

Let $f : dom(f) \to \mathbb{R}$ be differentiable and convex over $X \subseteq dom(f)$. Assume $L \in \mathbb{R}+$. Then function f is called $L-$smooth over X if

$$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}\|x-y\|^2 \quad , \quad \forall x, y \in X. \tag{A.4}$$

If $X = dom(f)$ then f is simply called $L-$ smooth.

Lemma A.0.3

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable. The following two statements are equivalent

1. f is $L-$ smooth
2. $\|\nabla f(y) - \nabla f(x)\| \leq L\|y-x\| \quad \forall x, y \in X \subseteq dom(f)$

Proof of Lemma A.0.3.

Section 2 [15] ☺ □

Lemma A.0.4

Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable $\mu-$strongly convex and $L-$smooth function. Then the following holds

$$\mu \leq \lambda(\nabla^2 f(x)) \leq L \quad \forall x \tag{A.5}$$

Proof of Lemma A.0.4.

Combined lemma 2.15 and 2.26 in [15] ☺ □

A.0.3 Understand Gradient Descent

We wish to define a 1-point iterative method for problem 2.2.1 with continous iterative function, where the sequence of iterates produced by the method is well defined and converges in the limit to the minimum value solution x^* , i.e

$$\begin{aligned}
&x_{k+1} = \phi(x_k) \quad \phi : \mathbb{R}^n \to \mathbb{R}^n \\
&x^* = \lim_{k\to\infty} x_k \\
&\phi \in C(\mathbb{R}^n) \implies \text{CONSISTENCY} \quad x^* = \phi(x^*)
\end{aligned}$$

Focusing on the following class of iterative functions:

$$x_{k+1} = x_k + t_k d_k \quad t_k \in \mathbb{R}, d_k \in \mathbb{R}^n$$

we defined the **decent direction d** as follow:

$$f(x+td) < f(x) \quad t \in \text{ some } U_t \subseteq \mathbb{R}$$

Hence we wish to find a a descent direction, hoping to convergence to the minimum of f.
The following claim is a key motivation for gradient descent direction.

Claim A.0.1 Steepest Direction: $d = -\frac{\nabla f(x)}{\|\nabla f(x)\|}$

For f continously differentiable, any direction $d \in \{d \in \mathbb{R}^n : \|d\|_2 = 1\}$ for some $t \in U_t$ we have

$$d^T \nabla f(x) < 0 \implies f(x+td) < f(x)$$

and

$$d = -\frac{\nabla f(x)}{\|\nabla f(x)\|} \implies \text{Steepest Direction}$$

Proof of claim A.0.1.
Let define $g(x) := \langle d, \nabla f(x) \rangle < 0$ by continuity of g s.t

$$\exists \bar{t} \quad \text{s.t} \quad g(x+td) < 0 \quad \forall t \in (0, \bar{t}] =: U_t$$

Given $x_{k+1} = x_k + td$, we define $\psi(\theta) := f(x_k + \theta td)$ then by **mean-value Thm** on $\psi(\theta)$ we have

$$\begin{aligned} &\exists \gamma \in (0,1) \quad s.t \quad \psi(1) = \psi(0) + \psi'(\gamma) \\ \Leftrightarrow \quad & f(x_k + td) = f(x_k) + t\langle d, \nabla f(x_k + \gamma td) \rangle \\ \Leftrightarrow \quad & f(x_k + td) - f(x_k) = t\langle d, \nabla f(x_k + \gamma td) \rangle \end{aligned}$$

Hence for $t\gamma \in U_t$ we have $\langle d, \nabla f(x_k + \gamma td) \rangle < 0 \implies f(x_{k+1}) < f(x_k)$
Using cauchy-schwartz

$$\begin{aligned} d_{opt} &= \inf_{d \in B_2(\mathbb{R}^n)} \langle d, \nabla f(x) \rangle \\ &= -\frac{\nabla f(x)}{\|\nabla f(x)\|_2} \end{aligned}$$

☺ □

Hence motivated by the claim we define the following 1-fixed point iterative method

$$x_{k+1} = x_k - t_k \nabla f(x_k)$$

A.0.4 Understand Momentum

We motivate momentum like algorithm though the lense of the continous version of the algorithms as in [45]. To illustrate what is meant by "continous version", assuming the gradient decent iteration, as

$$x_{k+1} = x_k - \gamma \nabla f(x_k)$$

we consider x_k to be sampled at each multiple of γ from a function $X : \mathbb{R}^+ \to \mathbb{R}^n$ i.e

$$x_k = X(k\gamma)$$

then for $t = k\gamma$ we have

$$\begin{aligned} X(t+\gamma) &= x_{k+1} = x_k - \gamma \nabla f(x_k) = X(t) - \gamma \nabla f(X(t)) \\ &\Leftrightarrow \frac{1}{\gamma}(X(t+\gamma) - X(t)) = -\nabla f(X(t)) \\ \lim_{\gamma \to 0} \implies &\text{ODE: } \dot{X}(t) = -\nabla f(X(t)) \end{aligned}$$

Note that $\nabla f(X(t)) = 0$ is a fixed point or a stationary point of the ODE, which is also a minimizer of a convex smooth function. Reversing the process, gradient descent can be seen as a finite difference approximation scheme of the ODE system.

Notice that the minimum of $f(x)$ fullfiling $\nabla f(x) = 0$ is also a stationary point of the following second order differential equation which interpretated from physics corresponds

$$\mu \frac{\partial^2 X(t)}{\partial^2 t} = -\nabla f(X(t)) - b\frac{\partial X(t)}{\partial t} \tag{A.6}$$

This equation describes the dynamics of a particle in terms of its momentum, the forces acting on it due to a potential, and friction:

- $\mu \frac{\partial^2 X(t)}{\partial t^2}$: This term represents the particle's change in momentum over time, where μ is the mass of the particle, and $\frac{\partial^2 X(t)}{\partial t^2}$ is its acceleration.
- $-\nabla f(X(t))$: This is the force acting on the particle due to a potential f. The negative gradient points towards the direction of the greatest decrease in potential energy, guiding the particle towards lower potential states.
- $-b\frac{\partial X(t)}{\partial t}$: This term models the friction or damping force, which is proportional and opposite to the velocity, thereby slowing the particle down.

Together, these terms model the movement of a particle as a balance between momentum, potential-driven forces, and friction, providing insights into how it will traverse space over time. This will motivate the name of **momentum** iterative procedures.

Applying a finite difference approximation yield

$$\frac{\mu}{(\Delta t)^2}\left[X(t+\Delta t) - 2X(t) + X(t-\Delta t)\right] \approx -\nabla f(X(t)) - b\frac{X(t+\Delta t) - X(t)}{\Delta t} \tag{A.7}$$

rearranging we have

$$X(t+\Delta t) = X(t) - \alpha\nabla f(X(t)) + \beta(X(t) - X(t+\Delta t)) \tag{A.8}$$

using the same idea of sampling from $X(t)$ we define the heavy ball momentum algorithm

$$x_{k+1} = x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1}) \tag{A.9}$$

A.0.5 Gelfand's Formula

Theorem A.0.1 Gelfand's Theorem

For any matrix norm $\|\cdot\|$, we have

$$\rho(A) = \lim_{k\to\infty} \|A^k\|^{\frac{1}{k}} \tag{A.10}$$