Randomized Iterative Methods for Linear Systems: Momentum, Inexactness and Gossip

Nicolas Loizou

Doctor of Philosophy
University of Edinburgh
2019

Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

(*Nicolas Loizou*)

To my parents,
Christaki and Agathi.

Abstract

In the era of big data, one of the key challenges is the development of novel optimization algorithms that can accommodate vast amounts of data while at the same time satisfying constraints and limitations of the problem under study. The need to solve optimization problems is ubiquitous in essentially all quantitative areas of human endeavor, including industry and science. In the last decade there has been a surge in the demand from practitioners, in fields such as machine learning, computer vision, artificial intelligence, signal processing and data science, for new methods able to cope with these new large scale problems.

In this thesis we are focusing on the design, complexity analysis and efficient implementations of such algorithms. In particular, we are interested in the development of randomized first order iterative methods for solving large scale linear systems, stochastic quadratic optimization problems and the distributed average consensus problem.

In Chapter 2, we study several classes of stochastic optimization algorithms enriched with *heavy ball momentum*. Among the methods studied are: stochastic gradient descent, stochastic Newton, stochastic proximal point and stochastic dual subspace ascent. This is the first time momentum variants of several of these methods are studied. We choose to perform our analysis in a setting in which all of the above methods are equivalent: convex quadratic problems. We prove global non-asymptotic linear convergence rates for all methods and various measures of success, including primal function values, primal iterates, and dual function values. We also show that the primal iterates converge at an accelerated linear rate in a somewhat weaker sense. This is the first time a linear rate is shown for the stochastic heavy ball method (i.e., stochastic gradient descent method with momentum). Under somewhat weaker conditions, we establish a sublinear convergence rate for Cesàro averages of primal iterates. Moreover, we propose a novel concept, which we call *stochastic momentum*, aimed at decreasing the cost of performing the momentum step. We prove linear convergence of several stochastic methods with stochastic momentum, and show that in some sparse data regimes and for sufficiently small momentum parameters, these methods enjoy better overall complexity than methods with deterministic momentum. Finally, we perform extensive numerical testing on artificial and real datasets.

In Chapter 3, we present a convergence rate analysis of *inexact* variants of stochastic gradient descent, stochastic Newton, stochastic proximal point and stochastic subspace ascent. A common feature of these methods is that in their update rule a certain sub-problem needs to be solved exactly. We relax this requirement by allowing for the sub-problem to be solved inexactly. In particular, we propose and analyze inexact randomized iterative methods for solving three closely related problems: a convex stochastic quadratic optimization problem, a best approximation problem and its dual – a concave quadratic maximization problem. We provide iteration complexity results under several assumptions on the inexactness error. Inexact variants of many popular and some more exotic methods, including randomized block Kaczmarz, randomized Gaussian Kaczmarz and randomized block coordinate descent, can be cast as special cases. Finally, we present numerical experiments which demonstrate the benefits of allowing inexactness.

When the data describing a given optimization problem is big enough, it becomes impossible to store it on a single machine. In such situations, it is usually preferable to distribute the data among the nodes of a cluster or a supercomputer. In one such setting the nodes cooperate to minimize the sum (or average) of private functions (convex or non-convex) stored at the nodes. Among the most popular protocols for solving this problem in a decentralized fashion (communication is allowed only between neighbors) are randomized gossip algorithms.

In Chapter 4 we propose a new approach for the design and analysis of randomized gossip

algorithms which can be used to solve the distributed average consensus problem, a fundamental problem in distributed computing, where each node of a network initially holds a number or vector, and the aim is to calculate the average of these objects by communicating only with its neighbors (connected nodes). The new approach consists in establishing new connections to recent literature on randomized iterative methods for solving large-scale linear systems. Our general framework recovers a comprehensive array of well-known gossip protocols as special cases and allow for the development of block and arbitrary sampling variants of all of these methods. In addition, we present novel and provably accelerated randomized gossip protocols where in each step all nodes of the network update their values using their own information but only a subset of them exchange messages. The accelerated protocols are the first randomized gossip algorithms that converge to consensus with a provably accelerated linear rate. The theoretical results are validated via computational testing on typical wireless sensor network topologies.

Finally, in Chapter 5, we move towards a different direction and present the first randomized gossip algorithms for solving the average consensus problem while at the same time protecting the private values stored at the nodes as these may be sensitive. In particular, we develop and analyze three privacy preserving variants of the randomized pairwise gossip algorithm ("randomly pick an edge of the network and then replace the values stored at vertices of this edge by their average") first proposed by Boyd et al. [16] for solving the average consensus problem. The randomized methods we propose are all dual in nature. That is, they are designed to solve the dual of the best approximation optimization formulation of the average consensus problem. We call our three privacy preservation techniques "Binary Oracle", "ϵ-Gap Oracle" and "Controlled Noise Insertion". We give iteration complexity bounds for the proposed privacy preserving randomized gossip protocols and perform extensive numerical experiments.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Prof. Peter Richtárik, for his guidance through every facet of the research world. Thank you for being such a great advisor, mentor and teacher, and for being an inspiring role model that I live up to both in my academic and personal life. Thanks for your continuous feedback, your constant encouragement, your great suggestions for writing and presentation, for offering career advice and for all the nice moments. I couldn't ask for a better advisor, mentor and a friend during my PhD studies.

I would also like to thank my second supervisor and mentor Dr. Lukasz Szpruch for various discussions about research and differences between related fields. I am also grateful to my examination committee, Prof. Coralia Cartis and Prof. Miguel F. Anjos for their suggestions for improvement and for their time.

For stimulating discussions and fun we had together during last four years, I thank other past and current members of our research group: Robert Mansel Gower, Jakub Konečný, Dominik Csiba, Filip Hanzely, Konstantin Mishchenko, Samuel Horváth, Aritra Dutta, El Houcine Bergou, Xun Qian, Adil Salim, Dmitry Kovalev, Elnur Gasanov, Alibek Sailanbayev. I would also like to thank my office mates Matt Booth, Juliet Cooke and Rodrigo Garcia Nava and the rest of my friends from Operational Research and Optimization group, Chrystalla Pavlou, Spyros Pougkakiotis, Minerva Martin del Campo, Xavier Cabezas Garcia, Ivet Galabova, Saranthorn Phusingha, Marion Lemery and Wenyi Qin for all the nice moments.

I am indebted to the University of Edinburgh and to the Principal's Career Development PhD Scholarship for funding my PhD studies. I would like to thank the school of Mathematics of the University of Edinburgh, for providing me with a wonderful work environment and for funding several trips during my PhD studies. I am also extremely grateful to Gill Law, Iain Dornan and Tatiana Chepelina who were always very helpful in overcoming every formal or administrative problems I encountered. I am most grateful to the Dr. Laura Wisewell Travel scholarship for funding my conference travels expenses in 2016 and 2017 and to Prof. Peter Richtárik for funding my conference travels expenses in 2018 and 2019. The chance to participate in international conferences, with all the experts in my area in one place, was invaluable. I would also like to thank the A.G. Leventis Foundation for the financial support during both my MSc and PhD studies.

I appreciate the advice, discussions and fun I had with many amazing people during my internship at Facebook AI Research in Montreal. More specifically, I would like to thank Dr. Mike Rabbat, Dr. Nicolas Ballas and Mahnoud Assran for the excellent collaboration and Prof. Joelle Pineau, Prof. Pascal Vincent, Dr. Adriana Romero, Dr. Michal Drozdzal for the great research environment. A special thanks should go to Dr. Mike Rabbat for the immerse trust and support I received during my internship. Mike has been an inspiration to me, a great enthusiastic collaborator, and warm person in general.

For willingness to help and provide recommendation, reference, or connection at various stages of my study, I would like to thank Dr. Martin Takáč, Dr. Martin Jaggi, Dr. Panos Parpas, Dr. Wolfram Wiesemann, Dr. Lin Xiao, Dr. Kimon Fountoulakis and Dr. Anastasios Kyrillidis. In addition, I am extremely thankful to Prof. Apostolos Burnetas and Prof. Apostolos Giannopoulos for introducing me to the areas of operational research and convex analysis, respectively.

This journey would not have been possible without the support of my parents, Christaki and Agathi. Thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams. I dedicate this thesis to you.

Finally, I would like to thank the two people that were the most important to me during

my time in Edinburgh:

My brother George, thank you for all the funny moments and the joyful experiences. You have been a constant support throughout my PhD studies and throughout my life. You have made my time in Edinburgh one of a kind. :)

My girlfriend, Katerina, thank you for your endless patience and for your continuous encouragement over the last four years. Thank you for reminding me what is important in life, for traveling around the world with me, for supporting all of my decisions. Thank you for being so amazing!!!

Contents

Chapter 1

Introduction

In this thesis we study the design and analysis of new efficient randomized iterative methods for solving large scale linear systems, stochastic quadratic optimization problems, the best approximation problem and quadratic optimization problems. A large part of the thesis is also devoted to the development of efficient methods for obtaining average consensus on large scale networks. As we will explain later in more detail, some of our proposed algorithms for solving the average consensus problem are carefully constructed special cases of methods for solving linear systems. All methods presented in the thesis (except two algorithms in the last chapter that converge with a sublinear rate) converge with global linear convergence rates, which means that they achieve an approximate solution of the problem fast.

In this introductory chapter we present the setting shared throughout the thesis and explain the relationships between the four problems mentioned above. We describe some baseline methods for solving these problems and present their convergence rates. Finally, we give a summary of the main contributions of each chapter.

Organization of thesis. The thesis is divided into two main parts. In the first part (Chapters 2 and 3) we present and analyze novel *momentum* (Chapter 2) and *inexact* (Chapter 3) variants of several randomized iterative methods for solving three closely related problems:

(i) stochastic convex quadratic minimization,

(ii) best approximation, and

(iii) (bounded) concave quadratic maximization.

In the second part (Chapters 4 and 5), we focus on the design and analysis of novel randomized gossip algorithms for solving the average consensus problem. This is a fundamental problem in distributed computing with the following goal: each node of a network initially holds a number or a vector, and the aim is for every node to calculate the average of these objects in a decentralized fashion (communicating with neighbors only). The proposed decentralized algorithms are inspired by recent advances in the area of randomized numerical linear algebra and optimization. In particular, in Chapter 4 we propose a new framework for the design and analysis of efficient randomized gossip protocols. We show how randomized iterative methods for solving linear systems can be interpreted as gossip algorithms when applied to special systems encoding the underlying network. Using the already developed framework of Chapter 4, we move towards a different direction and in Chapter 5 we present the first randomized gossip algorithms for solving the average consensus problem while at the same time protecting the information about the initial private values stored at the nodes.

Excluding some introductory results presented in this section, each chapter of the thesis is self-contained, including the objective, contributions, definitions and notation. However, to the extend that this was possible and meaningful, a unified notation has been adopted throughout the thesis.

1.1 Thesis' Philosophy: Place in the Literature

Here we start by providing a bird's-eye view of the main concepts and a simple first explanation of the context and of the problems under study. We present how this thesis is related to different areas of research and provide important connections that allow readers with different backgrounds to easily navigate through the main contributions of this work.

1.1.1 Bridge across several communities

This thesis is related to three different areas of research: linear algebra, stochastic optimization and machine learning.

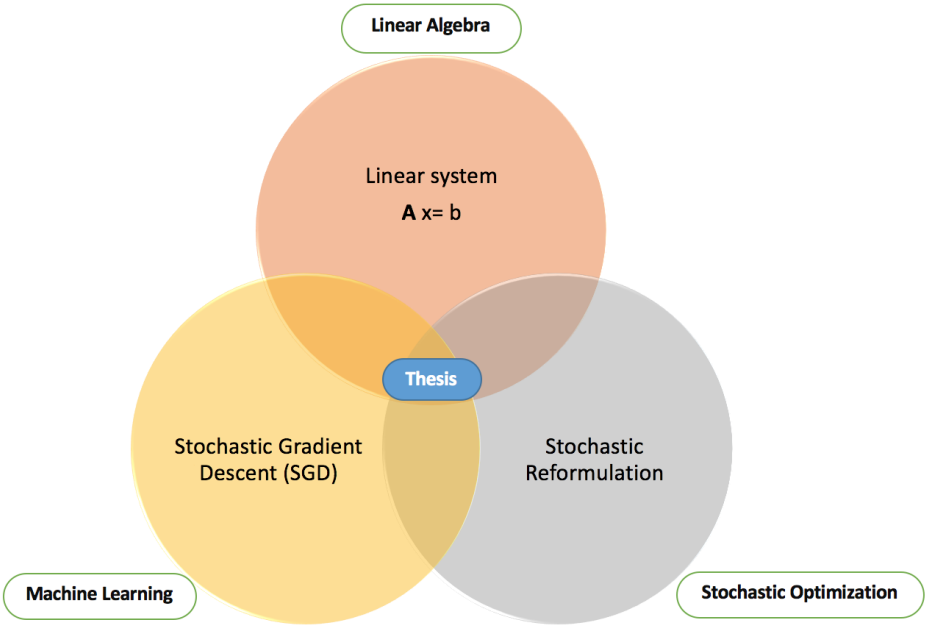


Figure 1.1: Thesis' place in the literature. Linear Algebra (through methods for solving linear systems), Stochastic Optimization (through stochastic reformulations) and Machine Learning (through stochastic gradient descent) are the three main areas of research related to this thesis.

Linear Algebra

Linear systems form the backbone of most numerical codes used in industry and academia. Solving large linear systems is a central problem in numerical linear algebra and plays an important role in computer science, mathematical computing, optimization, signal processing, engineering and many other fields.

In this thesis we are concerned with the problem of solving a *consistent* linear system. In particular, given a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and a vector $b \in \mathbb{R}^m$, we are interested to solve the problem:

$$\mathbf{A}x = b. \tag{1.1}$$

Main Assumption: Consistency. Throughout the thesis we assume that the linear system (1.1) has a solution $x^* \in \mathbb{R}^n$ (not necessarily unique) that satisfies $\mathbf{A}x^* = b$. That is, the linear system is consistent, i.e., $\mathcal{L} := \{x : \mathbf{A}x = b\} \neq \emptyset$. We make no extra assumption on the form, positive definiteness, rank or any other property of matrix $\mathbf{A}$. Thus, all methods proposed in this thesis converge under virtually no additional assumptions on the system beyond consistency. However, our methods are particularly well suited for the case of large over-determined linear systems. That is, to the case when the number of linear equations (rows) of the matrix is much larger than number of columns (variables) ($m \gg n$).

Stochastic Optimization

This thesis is also related to the stochastic optimization literature, through a recently proposed stochastic optimization reformulation of linear systems [168].

It is well known that the linear system (1.1) can be expressed as an optimization problem as follows [133]:

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}x-b\|^2 = \frac{1}{2}\sum_{i=1}^{m}\left(\mathbf{A}_{i:}^\top x - b_i\right)^2, \tag{1.2}$$

where $\mathbf{A}_{i:}$ denotes the i^{th} row of matrix $\mathbf{A}$. Note that if we denote with $\mathcal{F}$ the solution set of problem (1.2), then $\mathcal{F}=\mathcal{L}$, where $\mathcal{L}$ is the solution set of the consistent linear system (1.1).

The above approach of reformulating a linear system to an optimization problem is without doubt one of the most popular. However as we will later explain in more detail, it is not the only one. For example, one may instead consider the more general formulation

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{A}x-b\|_{\mathbf{V}}^2 = \frac{1}{2}\left(\mathbf{A}x-b\right)^\top \mathbf{V}\left(\mathbf{A}x-b\right), \tag{1.3}$$

where $\mathbf{V}\in\mathbb{R}^{m\times m}$ is a symmetric and positive definite matrix. In [168], Richtárik and Takáč proposed a stochastic optimization reformulation of linear systems similar to (1.3). In particular, they consider (1.3) with $\mathbf{V}=\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\mathbf{H}]$ and allow $\mathbf{V}$ to be positive semi-definite. The expectation is over random matrices $\mathbf{S}$ ($\mathbf{H}$ is matrix expression involving random matrix $\mathbf{S}$) that depend on an arbitrary user-defined distribution $\mathcal{D}$ and the matrix $\mathbf{A}$ of the linear system (1.1). Under a certain assumption on $\mathcal{D}$, for which the term *exactness* was coined in [168], the solution set of the stochastic optimization reformulation is identical to the solution set of the linear system. In [168], the authors provide necessary and sufficient conditions for exactness. Later in Sections 1.2 and 1.3 we describe exactness and comment on its importance in more detail.

In this thesis we design and analyze randomized iterative methods (stochastic optimization algorithms) for solving the stochastic convex quadratic minimization reformulation proposed in [168].

Machine Learning

Stochastic optimization problems are at the heart of many machine learning and statistical techniques used in data science. Machine learning practitioners, as part of their data analysis, are often interested in minimizing function f which in full generality takes the form:

$$\min_{x\in\mathbb{R}^n} \big[f(x)=\mathbb{E}_{i\sim\mathcal{D}} f_i(x)\big], \tag{1.4}$$

where $\mathbb{E}_{i\sim\mathcal{D}}$ denotes the expectation over an arbitrary distribution $\mathcal{D}$.

If we further assume that the distribution $\mathcal{D}$ is uniform over m functions f_i, the stochastic optimization problem is simplified to the finite-sum structure problem:

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x). \tag{1.5}$$

Problem (1.5) is referred to as Empirical Risk Minimization (ERM), and is one of the key optimization problems arising in large variety of models, ranging from simple linear regressions to deep learning. For example, note that problem (1.2) is also a special case of ERM (1.5) when functions $f_i:\mathbb{R}^n\to\mathbb{R}$ are chosen to be $f_i(x):=\frac{m}{2}\left(\mathbf{A}_{i:}^\top x-b_i\right)^2$.

A trivial benchmark for solving problem (1.5) in the case of differentiable function is Gradient Descent (GD). That is,

$$x^{k+1}=x^k-\omega^k\nabla f(x^k)=x^k-\omega^k\frac{1}{m}\sum_{i=1}^{m}\nabla f_i(x),$$

where ω^k is the stepsize parameter (learning rate). However in modern machine learning applications the number m of component functions f_i can be very large ($m\gg n$). As a result,

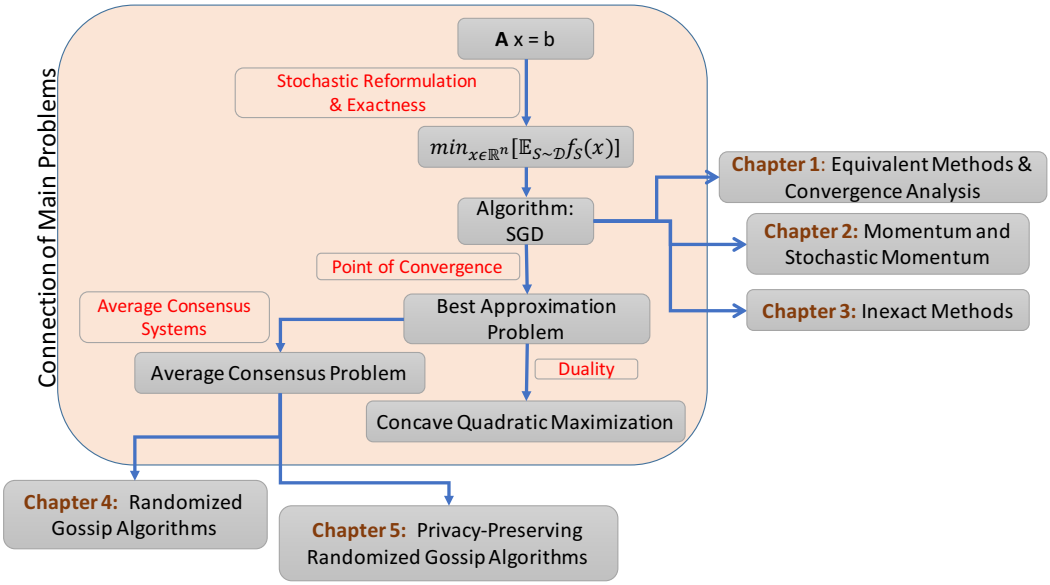


Figure 1.2: Roadmap of the Thesis. Relationships between the four main problems under study: (i) linear system, (ii) stochastic convex quadratic minimization, (iii) best approximation, (iv) (bounded) concave quadratic maximization. Connection to the average consensus problem and the main chapters of the thesis.

computing the full gradient in each iteration is prohibitively expensive and GD becomes impractical for most state-of-the-art applications.

To avoid such issues, machine learning practitioners use Stochastic Gradient Descent (SGD), a randomized variant of GD, first proposed by Robbins and Monro [169] in 1951 and which has enjoyed a lot of success ever since. For solving (1.5), SGD first uniformly at random samples function f_i (where $i \in \{1, \dots, m\}$) and then performs the iteration:

$$x^{k+1} = x^k - \omega^k \nabla f_i(x^k),$$

where ω^k is the stepsize parameter (learning rate). SGD has become the workhorse for training supervised machine learning problems which have the generic form (1.5) and many papers devoted to the understanding of its convergence behavior in different applications and under different assumptions on the functions f_i [137, 136, 81, 177, 170, 141, 68, 194].

This thesis is closely related to machine learning literature and the papers devoted to the analysis of SGD and its variants. In particular, besides other methods, we focus on analyzing SGD and two of its most popular variants: SGD with momentum (Chapter 2) and Inexact SGD (Chapter 3) for solving the stochastic optimization reformulation of linear systems proposed in [168].

1.1.2 Roadmap

In this subsection, by following the flowchart of Figure 1.2 we present the hierarchy of the main problems under study, explain the relationships between them and provide a brief summary of the chapters of the thesis. More details will be provided in the remaining sections of the Introduction.

In this thesis, we are studying the problem of solving large-dimensional consistent **linear systems** of the form $\mathbf{A}x = b$. In particular, we are adopting the stochastic optimization reformulation of linear systems first proposed in [168]. As we have already briefly mentioned, under a certain assumption (*exactness*) on the randomness of the stochastic reformulation, the solution set of the **stochastic convex quadratic minimization problem** is equal to the solution set of the original linear system. Hence, solving the stochastic optimization problem $\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}f_{\mathbf{S}}(x)$ is equivalent to solving the original linear system.

For solving the stochastic convex quadratic minimization problem one can use *stochastic gradient descent* (SGD), a popular stochastic optimization algorithm, particularly useful in machine learning applications (large scale setting). In Section 1.3 we explain how other stochastic optimization methods, like *stochastic Newton* (SN) method and *stochastic proximal point* (SPP) method, are identical to SGD for solving this particular problem and provide a simple analysis for their linear (exponential) convergence.

As it turns out, SGD and its equivalent methods converge to one particular minimizer of the stochastic optimization problem: the projection of their starting point x^0 onto the solution set of the linear system (1.1). This leads to the **best approximation problem**, which is the problem of projecting a given vector onto the solution space of the linear system. The best approximation problem is popular in numerical linear algebra and is normally solved using sketching techniques. We show how the sketch-and-project method proposed in [73] for solving the best approximation problem has also identical updates to SGD.

The dual of the best approximation problem is a **bounded unconstrained concave quadratic maximization problem**. In this thesis, we are also interested in the development and convergence analysis of efficient, dual in nature, algorithms for directly solving the dual of the best approximation problem. The baseline method for solving the dual of the best approximation problem is *stochastic dual subspace accent* (SDSA) [74]. As we will explain later, the random iterates of SGD, SN and SPP arise as affine images of the random iterates produced by SDSA.

In Chapters 2 and 3 we study novel *momentum* and *inexact* variants of several randomized iterative methods for solving the above problems. Among the methods studied are: stochastic gradient descent, stochastic Newton, stochastic proximal point and stochastic dual subspace ascent.

As we can also see in Figure 1.2, a large part of the thesis will be devoted to the development of efficient methods for solving the **average consensus (AC) problem**. In particular, we will explain how the AC problem can be expressed as a best approximation problem once we choose special linear systems encoding the underlying network (average consensus systems). In Chapter 4 we show how classical randomized iterative methods for solving the best approximation problem can be interpreted as gossip algorithms and explain in detail their decentralized nature. In Chapter 5, we present the first privacy-preserving randomized gossip algorithms that solve the AC problem while at the same time protect the private values stored at the nodes as these may be sensitive.

1.2 Stochastic Optimization Reformulation of Linear Systems

The starting point of this thesis is a general framework for studying consistent linear systems via carefully designed *stochastic reformulations* recently proposed by Richtárik and Takáč [168]. In particular, given the consistent linear system (1.1), the authors provide four reformulations in the form of a stochastic optimization problem, stochastic linear system, stochastic fixed point problem and a stochastic feasibility problem. These reformulations are equivalent in the sense that their solution sets are identical. That is, the set of minimizers of the stochastic optimization problem is equal to the set of solutions of the stochastic linear system and so on. Under a certain assumption on the randomness defining these reformulations, for which the term *exactness* was coined in [168], the solution sets of these reformulations are equal to the solution set of the linear system.

For the sake of a simplified narrative, in this thesis we choose to focus mostly on one of the above reformulations: the stochastic convex quadratic optimization problem, which can be expressed as follows:

$$\min_{x\in\mathbb{R}^n} f(x) := \mathbb{E}_{\mathbf{S}\sim\mathcal{D}} f_{\mathbf{S}}(x). \tag{1.6}$$

Here the expectation is over random matrices $\mathbf{S}$ drawn from an arbitrary, user defined, distribution $\mathcal{D}$ and $f_{\mathbf{S}}$ is a stochastic convex quadratic function of a least-squares type, defined

as

$$f_{\mathbf{S}}(x) := \frac{1}{2}\|\mathbf{A}x - b\|^2_{\mathbf{H}} = \frac{1}{2}(\mathbf{A}x - b)^\top \mathbf{H}(\mathbf{A}x - b). \tag{1.7}$$

Function $f_{\mathbf{S}}$ depends on the matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and vector $b \in \mathbb{R}^m$ of the linear system (1.1) and on a random symmetric positive semidefinite matrix

$$\mathbf{H} := \mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top. \tag{1.8}$$

The $n \times n$ positive definite matrix $\mathbf{B}$, in the expression of matrix $\mathbf{H}$, defines the geometry of the space and throughout the thesis, gives rise to an inner product

$$\langle x, z\rangle_{\mathbf{B}} := \langle \mathbf{B}x, z\rangle \tag{1.9}$$

and the induced norm $\|x\|_{\mathbf{B}} := (x^\top \mathbf{B}x)^{1/2}$ on $\mathbb{R}^n$. By $\dagger$ we denote the Moore-Penrose pseudoinverse.

On Moore-Pernose Pseudoinverse: The Moore-Pernose pseudoinverse matrix (or simply pseudoinverse) $\mathbf{M}^\dagger$ of a matrix $\mathbf{M}$ was first intoduced by Moore [125] and Penrose [154, 153] in their pioneering work.

A computationally simple and accurate way to compute the matrix $\mathbf{M}^\dagger$ is by using the singular value decomposition [67, 38]. That is, if $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ is the singular value decomposition of matrix $\mathbf{M}$, then $\mathbf{M}^\dagger = \mathbf{U}\mathbf{\Sigma}^\dagger\mathbf{V}^\top$, where the diagonal matrix $\mathbf{\Sigma}^\dagger$ is computed by taking the reciprocal of each non-zero element on the diagonal of matrix $\mathbf{\Sigma}$, leaving the zeros in place. That is $\mathbf{\Sigma}^\dagger_{ii} = \frac{1}{\mathbf{\Sigma}_{ii}}$ for all $\mathbf{\Sigma}_{ii} > 0$. Note that by its definition Moore-Penrose pseudoinverse is uniquely defined for all matrices (not necessarily square) whose entries are real or complex numbers.

It is worth to highlight that in this thesis, we applied the Moore-Pernose pseudoinverse only on symmetric positive semidefinite matrices. In particular, if $\mathbf{M} \succeq 0$ is a symmetric $m \times m$ matrix then its pseudoinverse will appear as a part of the expression $\mathbf{M}^\dagger b$, where $b \in \mathbb{R}^m$. Using properties of pseudoinverse this is equivalent to the least-norm solution of the least-squares problem $\min_x \|\mathbf{M}x - b\|^2$ [67, 74]. Hence, if the system $\mathbf{M}x = b$ has a solution the following holds:

$$\mathbf{M}^\dagger b = \operatorname{argmin}\|x\|^2 \quad \text{subject to} \quad \mathbf{M}x = b. \tag{1.10}$$

Let us know present some basic properties of the pseudoinverse:

- If matrix $\mathbf{M}$ is invertible, its pseudoinverse is its inverse. That is, $\mathbf{M}^\dagger = \mathbf{M}^{-1}$.
- The pseudoinverse of the pseudoinverse is the original matrix. That is , $(\mathbf{M}^\dagger)^\dagger = \mathbf{M}$
- If $\mathbf{M}$ is symmetric positive semidefinite matrix , then $\mathbf{M}^\dagger$ is also symmetric positive semidefinite matrix.
- There are several identities that can be used to cancel certain subexpressions or expand expressions involving pseudoinverses: $\mathbf{M}\mathbf{M}^\dagger\mathbf{M} = \mathbf{M}$, $\mathbf{M}^\dagger\mathbf{M}\mathbf{M}^\dagger = \mathbf{M}^\dagger$, $(\mathbf{M}^\top\mathbf{M})^\dagger\mathbf{M}^\top = \mathbf{M}^\dagger$, $(\mathbf{M}^\top)^\dagger = (\mathbf{M}^\dagger)^\top$ and $(\mathbf{M}\mathbf{M}^\top)^\dagger = (\mathbf{M}^\dagger)^\top\mathbf{M}^\dagger$. For more useful identities see [67].

As we have already mentioned, problem (1.6) is constructed in such a way that the set of minimizers of f is identical to the set of solutions of the given (consistent) linear system (1.1). In this sense, (1.6) can be seen as the reformulation of the linear system (1.1) into a stochastic optimization problem. As argued in [168], such reformulations provide an explicit connection between the fields of linear algebra and stochastic optimization, and allow the transfer of knowledge, techniques, and algorithms from one field to another. For instance, the randomized Kaczmarz method of Strohmer and Vershynin [182] for solving (1.1) is equivalent to the stochastic gradient descent method applied to (1.6), with $\mathcal{D}$ corresponding to a discrete distribution over unit coordinate vectors in $\mathbb{R}^m$ [133]. However, the flexibility of being able to choose $\mathcal{D}$ arbitrarily allows for numerous generalizations of the randomized Kaczmarz method [168]. Likewise, provably faster variants of the randomized Kaczmarz method (for instance, by utilizing importance sampling) can be designed using the connection.

Since their introduction in [168], stochastic reformulations of otherwise deterministic problems have found surprising applications in various areas, and are hence an important object of study in its own right. For instance, using a different stochastic reformulation Gower et al. [68] performed a tight convergence analysis of stochastic gradient descent in a more general convex setting, while [70] utilized "controlled" stochastic reformulations to develop a new approach to variance reduction for finite-sum problems appearing in machine learning. Further, this approach led to the development of the first accelerated quasi-Newton matrix update rules in the literature [72] and to the design of efficient randomized projection methods for convex feasibility problems [130]; all solving open problems in the literature.

Closed form expressions. We shall often refer to matrix expressions involving the random matrix $\mathbf{S}$ and the matrices $\mathbf{B}$ and $\mathbf{A}$. In order to keep these expressions brief throughout the thesis, it will be useful to define the $n \times n$ matrix:

$$\mathbf{Z} := \mathbf{A}^\top \mathbf{H} \mathbf{A} \overset{(1.8)}{=} \mathbf{A}^\top \mathbf{S} (\mathbf{S}^\top \mathbf{A} \mathbf{B}^{-1} \mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top \mathbf{A}. \tag{1.11}$$

Using matrix $\mathbf{Z}$ we can easily express important quantities related to the problems under study. For example, the stochastic functions $f_{\mathbf{S}}$ defined in (1.7) can be also expressed as

$$f_{\mathbf{S}}(x) \overset{(1.7)}{=} \frac{1}{2}(\mathbf{A}x - b)^\top \mathbf{H}(\mathbf{A}x - b) = \frac{1}{2}(x - x^*)^\top \mathbf{Z}(x - x^*), \tag{1.12}$$

where $x^* \in \mathcal{L}$. In addition, the gradient and the Hessian of $f_{\mathbf{S}}$ with respect to the $\mathbf{B}$ inner product (1.9) are equal to

$$\nabla f_{\mathbf{S}}(x) \overset{(1.7)}{=} \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}(\mathbf{A}x - b) = \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}\mathbf{A}(x - x^*) \overset{(1.11)}{=} \mathbf{B}^{-1}\mathbf{Z}(x - x^*), \tag{1.13}$$

where $x^* \in \mathcal{L}$ and $\nabla^2 f_{\mathbf{S}}(x) = \mathbf{B}^{-1}\mathbf{Z}$ [168].

Using the above expressions, the gradient and the Hessian of the objective function f of problem (1.6) are given by

$$\nabla f(x) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\nabla f_{\mathbf{S}}(x)] = \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{Z}\right](x - x^*),$$

and

$$\nabla^2 f = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\nabla^2 f_{\mathbf{S}}(x)] = \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{Z}\right],$$

respectively.

Projections. Let $\Upsilon \subseteq \mathbb{R}^n$ be a closed convex set. Throughout the thesis, with $\Pi_{\Upsilon,\mathbf{B}}$ we denote the projection operator onto Υ, in the $\mathbf{B}$-norm. That is, $\Pi_{\Upsilon,\mathbf{B}}(x) := \arg\min_{x' \in \Upsilon} \|x' - x\|_{\mathbf{B}}$. In particular, we are interested in the projection onto $\mathcal{L}$. An explicit formula for the projection onto $\mathcal{L}$ is given by

$$\Pi_{\mathcal{L},\mathbf{B}}(x) := \arg\min_{x' \in \mathcal{L}} \|x' - x\|_{\mathbf{B}} = x - \mathbf{B}^{-1}\mathbf{A}^\top(\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top)^\dagger(\mathbf{A}x - b). \tag{1.14}$$

A formula for the projection onto the *sketched system* $\mathcal{L}_{\mathbf{S}} := \{x : \mathbf{S}^\top \mathbf{A}x = \mathbf{S}^\top b\}$ is obtained by simply replacing matrix $\mathbf{A}$ and vector b in (1.14) with the matrix $\mathbf{S}^\top\mathbf{A}$ and vector $\mathbf{S}^\top b$, respectively. In this case we write $\Pi_{\mathcal{L}_{\mathbf{S}},\mathbf{B}}(x)$.

On complexity results. The complexity of the linearly convergent methods presented in this thesis is described by the spectrum of the following key matrix:

$$\mathbf{W} := \mathbf{B}^{-1/2}\mathbb{E}\left[\mathbf{Z}\right]\mathbf{B}^{-1/2}. \tag{1.15}$$

Matrix $\mathbf{W}$ has the same spectrum as the Hessian matrix $\nabla^2 f = \mathbf{B}^{-1}\mathbb{E}\left[\mathbf{Z}\right]$ and at the same time is symmetric and positive semidefinite (with respect to the standard inner product). Note that $\nabla^2 f$ is a not symmetric matrix (although it is self-adjoint with respect to the $\mathbf{B}$-inner product).

Let $\mathbf{W} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top = \sum_{i=1}^n \lambda_i u_i u_i^\top$ be the eigenvalue decomposition of $\mathbf{W}$, where $\mathbf{U} = [u_1, \ldots, u_n]$ is an orthonormal matrix composed of eigenvectors, and $\mathbf{\Lambda} = \mathbf{Diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is the diagonal matrix of eigenvalues with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$. In this thesis, by $\lambda_{\min}^+$ we will denote the smallest nonzero eigenvalue, and by $\lambda_{\max} = \lambda_1$ the largest eigenvalue of matrix $\mathbf{W}$. It was shown in [168] that $0 \leq \lambda_i \leq 1$ for all $i \in [n]$.

Main Assumption: Exactness. Note that in view of (1.12), $f_{\mathbf{S}}(x) = 0$ whenever $x \in \mathcal{L}$. However, $f_{\mathbf{S}}$ can be zero also for points x outside of $\mathcal{L}$. Clearly, f is nonnegative, and $f(x) = 0$ for $x \in \mathcal{L}$. However, without further assumptions, the set of minimizers of f can be larger than $\mathcal{L}$. The exactness assumption mentioned above ensures that this does not happen. For necessary and sufficient conditions for exactness, we refer the reader to [168]. One of a number of equivalent characterizations of exactness is the condition:

$$\text{Null}(\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\mathbf{Z}]) = \text{Null}(\mathbf{A}). \tag{1.16}$$

For this thesis it suffices to remark that a sufficient condition for exactness is to require $\mathbb{E}[\mathbf{H}]$ to be positive definite. This is easy to see by observing that $f(x) = \mathbb{E}[f_{\mathbf{S}}(x)] = \frac{1}{2}\|\mathbf{A}x - b\|^2_{\mathbb{E}[\mathbf{H}]}$. In other words, if $\mathcal{X} = \text{argmin} f(x)$ is the solution set of the stochastic optimization problem (1.6) and $\mathcal{L}$ the solution set of the linear system (1.1), then the notion of exactness is captured by:

$$\mathcal{X} = \mathcal{L}$$

1.3 Stochastic Gradient Descent (SGD) and Equivalent Iterative Methods

Problem (1.6) has several peculiar characteristics which are of key importance to this thesis. For instance, the Hessian of $f_{\mathbf{S}}$ is a (random) projection matrix, which can be used to show that $f_{\mathbf{S}}(x) = \frac{1}{2}\|\nabla f_{\mathbf{S}}(x)\|^2_{\mathbf{B}} = \frac{1}{2}\nabla f_{\mathbf{S}}(x)^\top \mathbf{B} \nabla f_{\mathbf{S}}(x)$ (see equation (1.38) in Lemma 2). Moreover, as we have already mentioned the Hessian of f has all eigenvalues bounded by 1, and so on. These characteristics can be used to show that several otherwise *distinct* stochastic algorithms for solving the stochastic optimization problem (1.6) are *identical*.

In particular, the following optimization methods for solving (1.6) are identical

- Stochastic Gradient Descent (SGD):

$$x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k), \tag{1.17}$$

- Stochastic Newton Method (SN)[1]:

$$x^{k+1} = x^k - \omega (\nabla^2 f_{\mathbf{S}_k}(x^k))^{\dagger_{\mathbf{B}}} \nabla f_{\mathbf{S}_k}(x^k), \tag{1.18}$$

- Stochastic Proximal Point Method (SPP)[2]:

$$x^{k+1} = \arg\min_{x \in \mathbb{R}^n} \left\{ f_{\mathbf{S}_k}(x) + \frac{1-\omega}{2\omega}\|x - x^k\|^2_{\mathbf{B}} \right\}. \tag{1.19}$$

In all methods $\omega > 0$, is a fixed stepsize and $\mathbf{S}_k$ is sampled afresh in each iteration from distribution $\mathcal{D}$.

Note that the equivalence of these methods for solving problem (1.6) is useful for the purposes of the thesis as it allows us to study their variants *with momentum* (Chapter 2) and their *inexact* variants (Chapter 3) by studying a single algorithm only.

[1] In this method we take the $\mathbf{B}$-pseudoinverse of the Hessian of $f_{\mathbf{S}_k}$ instead of the classical inverse, as the inverse does not exist. When $\mathbf{B} = \mathbf{I}$, the $\mathbf{B}$ pseudoinverse specializes to the standard Moore-Penrose pseudoinverse.

[2] In this case, the equivalence only works for $0 < \omega \leq 1$.

Using the closed form expression (1.13) of the gradient of functions $f_{\mathbf{S}}$, the update rules of the equivalent algorithms (1.17),(1.18) and (1.19) can be also written as:

$$x^{k+1} = x^k - \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x^k - b) \tag{1.20}$$

Following [168], we name the algorithmic update of equation (1.20), *basic method* and we use this in several parts of this thesis to simultaneously refer to the above equivalent update rules.

By choosing appropriately the two main parameters of the basic method, the matrix $\mathbf{B}$ and distribution $\mathcal{D}$ of the random matrices $\mathbf{S}$, we can recover a comprehensive array of well known algorithms for solving linear systems as special cases, such as the randomized Kaczmarz method, randomized Gauss Seidel (randomized coordinate descent) and their block variants. In addition, it is worth to notice that the basic method allows for a much wider selection of these two parameters, which means that it is possible to obtain a number of new specific and possibly more exotic algorithms as special cases. Hence, by having a convergence analysis for the general method (1.20) we can easily obtain the convergence rates of all these special cases by choosing carefully the combinations of the two main parameters.

Example 1. As a special case of the general framework, let us choose $\mathbf{B} = \mathbf{I}$ and $\mathbf{S}_k = e_i$, where $i \in [m]$ is chosen in each iteration independently, with probability $p_i > 0$. Here with $e_i \in \mathbb{R}^m$ we denote the i^{th} unit coordinate vector in $\mathbb{R}^m$. In this setup the update rule (1.20) simplifies to:

$$x^{k+1} = x^k - \omega \frac{\mathbf{A}_{i:}x^k - b_i}{\|\mathbf{A}_{i:}\|^2}\mathbf{A}_{i:}^\top. \tag{1.21}$$

where $\mathbf{A}_{i:}$ indicates the i^{th} row of matrix $\mathbf{A}$. This is a relaxed variant (stepsize not necessarily $\omega = 1$) of the randomized Kaczmarz method [182].

On Exactness. An important assumption that is required for the convergence analysis of the randomized iterative methods presented in this thesis is *exactness*. The exactness property is of key importance for the setting under study, and should be seen as an assumption on the distribution $\mathcal{D}$ and not on matrix $\mathbf{A}$.

Clearly, an assumption on the distribution $\mathcal{D}$ of the random matrices $\mathbf{S}$ should be required for the convergence of (1.20). For an instance, if $\mathcal{D}$ in the randomized Kaczmarz method (1.21), is such that, $\mathbf{S} = e_1$ with probability 1, where $e_1 \in \mathbb{R}^m$ be the 1^{st} unit coordinate vector in $\mathbb{R}^m$, then the algorithm will select the same row of matrix $\mathbf{A}$ in each step. For this choice of distribution it is clear that the algorithm will not converge to a solution of the linear system. The exactness assumption guarantees that this will not happen.

For necessary and sufficient conditions for exactness, we refer the reader to [168]. For this thesis, it suffices to remark that the exactness condition is very weak, allowing $\mathcal{D}$ to be virtually any reasonable distribution of random matrices. For instance, as we have already mentioned, a sufficient condition for exactness is for the matrix $\mathbb{E}[\mathbf{H}]$ to be positive definite [74].

A much stronger condition than exactness is $\mathbb{E}[\mathbf{Z}] \succ 0$ which has been used for the convergence analysis of (1.20) in [73]. In this case, the matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ of the linear system requires to have full column rank and as a result the consistent linear system has a unique solution.

1.4 Best Approximation and its Dual Problem

Best approximation problem. It was shown in [168] that SGD, SN and SPP converge to a very particular minimizer of f: the projection in the $\mathbf{B}$-norm, of the starting point x^0 onto the solution set of the linear system (1.1). That is, $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. This naturally leads to the *best approximation problem*, which is the problem of projecting a given vector onto the solution space of the linear system (1.1):

$$\min_{x\in\mathbb{R}^n} P(x) := \frac{1}{2}\|x - x^0\|_{\mathbf{B}}^2 \quad \text{subject to} \quad \mathbf{A}x = b. \tag{1.22}$$

Note that, unlike the linear system (1.1), which is allowed to have multiple solutions, the best approximation problem has always (from its construction) a unique solution.

For solving problem (1.22), the *Sketch and Project Method (SPM)*:

$$\begin{aligned} x^{k+1} \quad &:= \operatorname{argmin}_{x\in\mathbb{R}^n} \|x - x^k\|^2_{\mathbf{B}} \\ \text{subject to} \quad &\mathbf{S}_k^\top \mathbf{A}x = \mathbf{S}_k^\top b\,. \end{aligned} \tag{1.23}$$

was analyzed in [73, 74]. The name "sketch-and-project" method is justified by the iteration structure which consists of two steps: (i) draw a random matrix $\mathbf{S}_k$ from distribution $\mathcal{D}$ and formulate the *sketched* system $\mathcal{L}_{\mathbf{S}_k}$, (ii) *project* the last iterate x^k onto $\mathcal{L}_{\mathbf{S}_k}$. Analysis in [73] was done under the assumption that $\mathbf{A}$ has full column rank. This assumption was lifted in [74], and a *duality* theory for the method developed.

Using the closed form expression of projection (1.14), the iterative process of (1.23) can be equivalently written as [73]:

$$x^{k+1} = \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) \overset{(1.14)}{=} x^k - \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger(\mathbf{S}_k^\top\mathbf{A}x^k - \mathbf{S}_k^\top b), \tag{1.24}$$

and for the more general case of $\omega \neq 1$ the update takes the form:

$$x^{k+1} = \omega\Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) + (1-\omega)x^k. \tag{1.25}$$

By combining the definition of projection (1.25) and the update rule of equation (1.14) it can be easily observed that the sketch and project method is identical to the basic method (1.20). As a result, it is also identical to the previously mentioned algorithms, SGD (1.17), SN (1.18) and SPP (1.19). Thus, these methods can be also interpreted as randomized projection algorithms.

On Sketching. In numerical linear algebra, sketching is one of the most popular techniques used for the evaluation of an approximate solution of large dimensional linear systems $\mathbf{A}x = b$ where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$ [197].

Let $\mathbf{S} \in \mathbb{R}^{m\times q}$ be a random matrix with the same number of rows as $\mathbf{A}$ but far fewer columns ($n \gg q$). The goal of sketching is to design the distribution of random matrix $\mathbf{S}$ such that the solutions set of the much smaller (and potential much easier to solve) sketched system $\mathbf{S}^\top\mathbf{A}x = \mathbf{S}^\top b$ to be close to the solution set of the original large dimensional system, with high probability. That is, $\|\mathbf{S}^\top\mathbf{A}x - \mathbf{S}^\top b\|^2 \leq (1+\epsilon)\|\mathbf{A}x - b\|^2$ with high probability. Determining the random matrix $\mathbf{S}$ that satisfy the above constraint can be challenging and often depends on the properties and form of matrix $\mathbf{A}$. For recent advances in the area of sketching we suggest [197, 66, 37, 122, 44, 155].

In our setting, as we have already described above, sketching is part of our iterative process. In each iteration sketching is used in combination with a projection step in order to evaluate an exact solution of the sketched system.

On Sketch and Project Methods. Variants of the sketch-and-project method have been recently proposed for solving several other problems. [69, 75] use sketch-and-project ideas for the development of linearly convergent randomized iterative methods for computing/estimating the inverse and pseudoinverse of a large matrix, respectively. A limited memory variant of the stochastic block BFGS method for solving the empirical risk minimization problem arising in machine learning was proposed by [71]. Tu et al. [192] utilize the sketch-and-project framework to show that breaking block locality can accelerate block Gauss-Seidel methods. In addition, they develop an accelerated variant of the method for a specific distribution $\mathcal{D}$. An accelerated (in the sense of Nesterov) variant of the sketch and prokect method proposed in [72] for the more general Euclidean setting and applied to matrix inversion and quasi-Newton updates. As we have already mentioned, in [168], through the development of stochastic reformulations, a stochastic gradient descent interpretation of the sketch and project method have been proposed. Similar stochastic reformulations that have the sketch and project method as special case were also proposed in the more general setting of convex optimization [68] and in the context of variance-reduced methods [70].

The dual problem. Duality is an important tool in optimization literature and plays a major role in the development and understanding of many popular randomized optimization algorithms. In this thesis, we are also interested in the development of efficient, dual in nature, algorithms for directly solving the dual of the best approximation problem.

In particular, the Lagrangian dual of (1.22) is the (bounded) unconstrained concave quadratic maximization problem[3]

$$\max_{y\in\mathbb{R}^m} D(y) := (b-\mathbf{A}x^0)^\top y - \frac{1}{2}\|\mathbf{A}^\top y\|^2_{\mathbf{B}^{-1}}. \tag{1.26}$$

Boundedness follows from consistency. It turns out that by varying $\mathbf{A}, \mathbf{B}$ and b (but keeping consistency of the linear system), the dual problem in fact captures *all* bounded unconstrained concave quadratic maximization problems.

Let us define an affine mapping from $\mathbb{R}^m$ to $\mathbb{R}^n$ as follows:

$$\phi(y) := x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y. \tag{1.27}$$

It turns out, from Fenchel duality[4], that for any dual optimal y^*, the vector $\phi(y^*)$ must be primal optimal [74]. That is:

$$x^* = \phi(y^*) = x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^*. \tag{1.28}$$

A dual variant of the basic method for solving problem (1.26) was first proposed in [74]. The dual method—*Stochastic Dual Subspace Ascent (SDSA)*—updates the dual vectors y^k as follows:

$$y^{k+1} = y^k + \omega\mathbf{S}_k\lambda^k, \tag{1.29}$$

where the random matrix $\mathbf{S}_k$ is sampled afresh in each iteration from distribution $\mathcal{D}$, and λ^k is chosen to maximize the dual objective D: $\lambda^k \in \arg\max_\lambda D(y^k+\mathbf{S}_k\lambda)$. More specifically, SDSA is defined by picking the maximizer with the smallest (standard Euclidean) norm. This leads to the formula:

$$\lambda^k = \left(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top\left(b-\mathbf{A}(x^0+\mathbf{B}^{-1}\mathbf{A}^\top y^k)\right). \tag{1.30}$$

Note that, SDSA proceeds by moving in random subspaces spanned by the random columns of $\mathbf{S}_k$. In the special case when $\omega = 1$ and $y^0 = 0$, Gower and Richtárik [74] established the following relationship (affine mapping $\phi : \mathbb{R}^m \mapsto \mathbb{R}^n$) between the iterates $\{x^k\}_{k\geq 0}$ produced by the primal methods (1.17), (1.18), (1.19), (1.25) (which are equivalent), and the iterates $\{y^k\}_{k\geq 0}$ produced by the dual method (1.29):

$$x^k = \phi(y^k) \overset{(1.27)}{=} x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k. \tag{1.31}$$

In Section 1.5 we show with a simple proof how this equivalence extends beyond the $\omega = 1$ case, specifically for $0 < \omega < 2$ (see Proposition 4). Later, a similar approach will be used in the derivation of the momentum and inexact variant of SDSA in Chapters 2 and 3 respectively.

An interesting property that holds between the suboptimalities of the primal methods and SDSA is that the dual suboptimality of y in terms of the dual function values is equal to the primal suboptimality of $\phi(y)$ in terms of distance [74]. That is,

$$D(y^*) - D(y) = \frac{1}{2}\|\phi(y^*) - \phi(y)\|^2_{\mathbf{B}}. \tag{1.32}$$

This simple-to-derive result (by combining the expression of the dual function $D(y)$ (1.26) and the equation (1.27)) gives for free the convergence analysis of SDSA, in terms of dual function suboptimality once the analysis of the primal methods is available (see Proposition 5 in Section 1.5).

Note that SDSA update (1.29)+(1.30) depends on the same two parameters, matrix $\mathbf{B}$ and distribution $\mathcal{D}$, of the basic method (SGD, SN and SPP). Therefore, similar to the previous sub-

[3] Technically problem (1.26) is both the Lagrangian and the Fenchel dual of (1.22) [74].

[4] For more details on Fenchel duality, see [15].

section, by choosing appropriately the two parameters we can recover many known algorithms as special cases of SDSA.

Example 2. Let $\mathbf{B} = \mathbf{I}$ and $\mathbf{S}_k = e_i$, where $i \in [m]$ is chosen in each iteration independently, with probability $p_i > 0$. Here with $e_i \in \mathbb{R}^m$ we denote the i^{th} unit coordinate vector in $\mathbb{R}^m$. In this setup the update rule (1.29) simplifies to:

$$y^{k+1} = y^k + \omega \frac{b_i - \mathbf{A}_{i:}x^0 - \mathbf{A}_{i:}\mathbf{A}^\top y^k}{\|\mathbf{A}_{i:}\|^2} e_i. \tag{1.33}$$

where $\mathbf{A}_{i:}$ indicates the i^{th} row of matrix $\mathbf{A}$. This is the randomized coordinate ascent method [139] applied to the dual problem. Having said that, the analysis provided in [139] does not apply because the objective of the dual problem is not strongly concave function.

1.5 Simple Analysis of Baseline Methods

Having presented the problems that we are interested in this thesis and explained the relationships between them, let us know present some interesting properties of our setting and a simple convergence analysis of the baseline methods for solving them.

In Sections 1.3 and 1.4 we have introduced these baseline methods. As a reminder, these are the stochastic gradient descent (SGD) (1.17), stochastic Newton method (SN) (1.18), stochastic proximal point method (SPP) (1.19), sketch and project method (SPM) (1.25), and stochastic dual subspace ascent (SDSA)(1.29).

To simplify the presentation, in the remaining sections of the introduction we focus on two of these algorithms: SGD and SDSA. Recall at this point that SGD, SN, SPP and SPM have identical updates for the problems under study. This means that the analysis presented here for SGD holds for all of these methods.

We start by presenting some interesting properties of the stochastic quadratic optimization problem (1.6) and discussing connections with existing literature. Then we focus on the convergence analysis results.

1.5.1 Technical preliminaries

Recently, linear convergence of opimization methods has been established under several conditions that are satisfied in many realistic scenarios. We refer the interested reader to [92] and [129] for more details on these conditions and how they are related to each other.

In this thesis, we are particularly interested in the Quadratic Growth condition (QG). We say that a function f satisfies the QG inequality if the following holds for some $\mu > 0$:

$$\frac{\mu}{2}\|x - x^*\|^2 \leq f(x) - f(x^*). \tag{1.34}$$

Here x^* is the projection of vector x onto the solution set $\mathcal{X}$ of the optimization problem $\min_{x\in\mathbb{R}^n} f(x)$ and $f(x^*)$ denotes the optimal function value.

Under this condition it can be shown that SGD with constant stepsize ω converges with a linear rate up to a neighborhoud around the optimal point that is proportional to the value of ω [92]. For general convex function the convergence of SGD is sublinear [169, 136].

In [168] it was shown the the function of the stochastic quadratic optimization problem (1.6) satisfies the QG condition as well. In particular the following lemma was proved.

Lemma 1 (Quadratic bounds, [168]). *For all $x \in \mathbb{R}^n$ and $x^* \in \mathcal{L}$ the objective function of the stochastic optimization problem* (1.6) *satisfies:*

$$\lambda^+_{\min} f(x) \leq \frac{1}{2}\|\nabla f(x)\|^2_{\mathbf{B}} \leq \lambda_{\max} f(x) \tag{1.35}$$

and

$$f(x) \leq \frac{\lambda_{\max}}{2}\|x - x^*\|^2_{\mathbf{B}}. \tag{1.36}$$

Moreover, if exactness is satisfied, and we let $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x)$, we have

$$\frac{\lambda_{\min}^+}{2}\|x - x^*\|_{\mathbf{B}}^2 \leq f(x). \tag{1.37}$$

Note that inequality (1.37) is precisely the quadratic growth condition (1.34) with $\mu = \lambda_{\min}^+$ and $f(x^*) = 0$.

The following identities were also established in [168]. For completeness, we include different (and somewhat simpler) proofs here.

Lemma 2. *For all $x \in \mathbb{R}^n$ and any $\mathbf{S} \sim \mathcal{D}$ we have*

$$f_{\mathbf{S}}(x) = \frac{1}{2}\|\nabla f_{\mathbf{S}}(x)\|_{\mathbf{B}}^2. \tag{1.38}$$

Moreover, if $x^* \in \mathcal{L}$ (i.e., if x^* satisfies $\mathbf{A}x^* = b$), then for all $x \in \mathbb{R}^n$ we have

$$f_{\mathbf{S}}(x) = \frac{1}{2}\langle \nabla f_{\mathbf{S}}(x), x - x^*\rangle_{\mathbf{B}}, \tag{1.39}$$

and

$$f(x) = \frac{1}{2}\langle \nabla f(x), x - x^*\rangle_{\mathbf{B}}. \tag{1.40}$$

Proof. In view of (1.13), and since $\mathbf{Z}\mathbf{B}^{-1}\mathbf{Z} = \mathbf{Z}$ (see [168]), we have

$$\begin{aligned}\|\nabla f_{\mathbf{S}}(x)\|_{\mathbf{B}}^2 &\overset{(1.13)}{=} \|\mathbf{B}^{-1}\mathbf{Z}(x - x^*)\|_{\mathbf{B}}^2 = (x - x^*)^\top \mathbf{Z}\mathbf{B}^{-1}\mathbf{Z}(x - x^*) = (x - x^*)^\top \mathbf{Z}(x - x^*) \\ &\overset{(1.11)}{=} (x - x^*)^\top \mathbf{A}^\top \mathbf{H}\mathbf{A}(x - x^*) = (\mathbf{A}x - b)^\top \mathbf{H}(\mathbf{A}x - b) \overset{(1.7)}{=} 2f_{\mathbf{S}}(x).\end{aligned}$$

Moreover,

$$\begin{aligned}\langle \nabla f_{\mathbf{S}}(x), x - x^*\rangle_{\mathbf{B}} &\overset{(1.13)}{=} \langle \mathbf{B}^{-1}\mathbf{Z}(x - x^*), x - x^*\rangle_{\mathbf{B}} \\ &= (x - x^*)^\top \mathbf{Z}\mathbf{B}^{-1}\mathbf{B}(x - x^*) \overset{(1.12)}{=} 2f_{\mathbf{S}}(x).\end{aligned}$$

By taking expectations in the last identity with respect to the random matrix $\mathbf{S}$, we get $\langle \nabla f(x), x - x^*\rangle_{\mathbf{B}} = 2f(x)$. □

No need for variance reduction SGD is arguably one of the most popular algorithms in machine learning. Unfortunately, SGD suffers from slow convergence, which is due to the fact that the variance of the stochastic gradient as an estimator of the gradient does not naturally diminish. For this reason, SGD is typically used with a decreasing stepsize rule, which ensures that the variance converges to zero. However, this has an adverse effect on the convergence rate. For instance, SGD has a sublinear rate even if the function to be minimized is strongly convex (conergence to the optimum point, not to a neighborhoud around it). To overcome this problem, a new class of so-called *variance-reduced* methods was developed over the last 8 years, including SAG [173], SDCA [178, 166], SVRG/S2GD [89, 97], minibatch SVRG/S2GD [96], and SAGA [34, 33].

In our setting, we assume that the linear system (1.1) is feasible. Thus, it follows that the stochastic gradient vanishes at the optimal point (i.e., $\nabla f_{\mathbf{S}}(x^*) = 0$ for any $\mathbf{S}$). This suggests that additional variance reduction techniques are not necessary since the variance of the stochastic gradient drops to zero as we approach the optimal point x^*. In particular, in our context, SGD with fixed stepsize enjoys linear rate without any variance reduction strategy (see Theorem 3). Hence, in this thesis we can bypass the development of variance reduction techniques, which allows us to focus on the momentum term in Chapter 2 on the inexact computations in Chapter 3 and on gossip protocols that converge to consensus in Chapters 4 and 5.

1.5.2 Theoretical guarantees

The following convergence rates of SGD and SDSA are easy to establish, having the bounds and identities of Lemmas 1 and 2. Nevertheless we present the statements of the theorems and proofs for completeness and because we use similar ideas and approaches in the rest of the thesis. For the benefit of the reader, we also include the derivations of the two equations (1.31) and (1.32) presented in the previous section which connect the primal and the dual iterates.

Theorem 3 ([168]). *Let assume exactness and let $\{x^k\}_{k=0}^\infty$ be the iterates produced by SGD with constant stepsize $\omega \in (0,2)$. Set $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Then,*

$$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \left[1 - \omega(2-\omega)\lambda_{\min}^+\right]^k \|x^0 - x^*\|_{\mathbf{B}}^2. \tag{1.41}$$

Proof.

$$\begin{aligned}
\|x^{k+1} - x^*\|_{\mathbf{B}}^2 &\overset{(1.17)}{=} \|x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*\|_{\mathbf{B}}^2 \\
&= \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega\langle x^k - x^*, \nabla f_{\mathbf{S}_k}(x^k)\rangle_{\mathbf{B}} + \omega^2\|\nabla f_{\mathbf{S}_k}(x^k)\|_{\mathbf{B}}^2 \\
&\overset{(1.38),(1.39)}{=} \|x^k - x^*\|_{\mathbf{B}}^2 - 4\omega f_{\mathbf{S}_k}(x^k) + 2\omega^2 f_{\mathbf{S}_k}(x^k) \\
&= \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega(2-\omega) f_{\mathbf{S}_k}(x^k).
\end{aligned} \tag{1.42}$$

By taking expectation with respect to $\mathbf{S}_k$ and using quadratic growth inequality (1.37):

$$\begin{aligned}
\mathbb{E}_{\mathbf{S}_k}[\|x^{k+1} - x^*\|_{\mathbf{B}}^2] &= \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega(2-\omega) f(x^k) \\
&\overset{\omega\in(0,2),\,(1.37)}{\leq} \|x^k - x^*\|_{\mathbf{B}}^2 - \omega(2-\omega)\lambda_{\min}^+\|x^k - x^*\|_{\mathbf{B}}^2 \\
&= \left[1 - \omega(2-\omega)\lambda_{\min}^+\right]\|x^k - x^*\|_{\mathbf{B}}^2.
\end{aligned} \tag{1.43}$$

Taking expectation again and by unrolling the recurrence we obtain (1.41). □

Proposition 4. *Let $\{x^k\}_{k=0}^\infty$ be the iterates produced by SGD(1.17) with $\omega \in (0,2)$. Let $y^0 = 0$, and let $\{y^k\}_{k=0}^\infty$ be the iterates of SDSA (1.29). Assume that the methods use the same stepsize $\omega \in (0,2)$ and the same sequence of random matrices $\mathbf{S}_k$. Then $x^k = \phi(y^k) = x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k$ holds for all k. That is, the primal iterates arise as affine images of the dual iterates.*

Proof. First note that

$$\nabla f_{\mathbf{S}_k}(\phi(y^k)) \overset{(1.13)}{=} \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top(\mathbf{A}\phi(y^k) - b) \overset{(1.30)}{=} -\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda^k. \tag{1.44}$$

We now use this to show that

$$\begin{aligned}
\phi(y^{k+1}) &\overset{(1.31)}{=} x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^{k+1} \overset{(1.29)}{=} x^0 + \mathbf{B}^{-1}\mathbf{A}^\top\left[y^k + \omega\mathbf{S}_k\lambda^k\right] \\
&= x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k + \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda^k \\
&\overset{(1.31),(1.44)}{=} \phi(y^k) - \omega\nabla f_{\mathbf{S}_k}(\phi(y^k)).
\end{aligned}$$

So, the sequence of vectors $\{\phi(y^k)\}$ satisfies the same recursion to the sequence $\{x^k\}$ defined by SGD. It remains to check that the starting vectors of both recursions coincide. Indeed, since $y^0 = 0$, we have $x^0 = \phi(0) = \phi(y^0)$. □

Proposition 5 ([74]). *Let y^* be a solution of the dual problem (1.26). Then,*

$$D(y^*) - D(y^k) = \frac{1}{2}\|x^k - x^*\|_{\mathbf{B}}^2.$$

Proof.

$$\begin{aligned}
D(y^*) - D(y^k) &\overset{(1.26)}{=} (b - \mathbf{A}x^0)^\top y^* - \frac{1}{2}\|\mathbf{A}^\top y^*\|^2_{\mathbf{B}^{-1}} - (b - \mathbf{A}x^0)^\top y^k + \frac{1}{2}\|\mathbf{A}^\top y^k\|^2_{\mathbf{B}^{-1}} \\
&= (b - \mathbf{A}x^0)^\top (y^* - y^k) - \frac{1}{2}(\mathbf{A}^\top y^*)^\top \mathbf{B}^{-1}\mathbf{A}^\top y^* \\
&\quad + \frac{1}{2}(\mathbf{A}^\top y^k)^\top \mathbf{B}^{-1}\mathbf{A}^\top y^k \\
&\overset{(*)}{=} (\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top y^*)^\top (y^* - y^k) - \frac{1}{2}(\mathbf{A}^\top y^*)^\top \mathbf{B}^{-1}\mathbf{A}^\top y^* \\
&\quad + \frac{1}{2}(\mathbf{A}^\top y^k)^\top \mathbf{B}^{-1}\mathbf{A}^\top y^k \\
&= \frac{1}{2}(y^* - y^k)\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top (y^* - y^k) \\
&= \frac{1}{2}\|\mathbf{B}^{-1}\mathbf{A}^\top (y^* - y^k)\|^2_{\mathbf{B}} \\
&\overset{(1.28)+(1.31)}{=} \frac{1}{2}\|x^k - x^*\|^2_{\mathbf{B}}. \qquad (1.45)
\end{aligned}$$

In the $(*)$ equality above we use (1.28) for the optimal primal and dual values. In particular, $x^0 = x^* - \mathbf{B}^{-1}\mathbf{A}^\top y^* \Rightarrow \mathbf{A}x^0 = \mathbf{A}x^* - \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top y^* = b - \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top y^*$. □

The next theorem has been proved in [74] for the case of $\omega = 1$. Here we extend this convergence to the more general case of $0 < \omega < 2$.

Theorem 6. *Let us assume exactness. Choose $y^0 = 0 \in \mathbb{R}^m$. Let $\{y^k\}_{k=0}^\infty$ be the sequence of random iterates produced by SDSA with stepsize $0 \le \omega \le 2$. Then,*

$$\mathbb{E}[D(y^*) - D(y^k)] \le \left[1 - \omega(2-\omega)\lambda^+_{\min}\right]^k \left(D(y^*) - D(y^0)\right). \qquad (1.46)$$

Proof. This follows by applying Theorem 3 together with Proposition 5. □

1.5.3 Iteration Complexity

In several parts of this thesis we compare the performance of linearly convergent algorithms using their iteration complexity bounds. That is, we derive a lower bound on the number of iterations that are sufficient to achieve a prescribed accuracy. The following lemma shows the derivation of this bound for the sequence $\{A^k\}_{k=0}^\infty$.

Lemma 7. *Consider a non-negative sequence $\{A^k\}_{k=0}^\infty$ satisfying*

$$A^k \le \rho^k A^0, \qquad (1.47)$$

where $\rho \in (0,1)$. Then, for a given $\epsilon \in (0,1)$ and for:

$$k \ge \frac{1}{1-\rho}\log\left(\frac{1}{\epsilon}\right) \qquad (1.48)$$

it holds that:

$$A^k \le \epsilon A^0 \qquad (1.49)$$

Proof. Note that since $\rho \in (0,1)$, we have:

$$\frac{1}{1-\rho}\log\left(\frac{1}{\rho}\right) > 1. \qquad (1.50)$$

Therefore,

$$\log\left(\frac{A^0}{A^k}\right) \overset{(1.47)}{\ge} k\log\left(\frac{1}{\rho}\right) \overset{(1.48)}{\ge} \frac{1}{1-\rho}\log\left(\frac{1}{\epsilon}\right)\log\left(\frac{1}{\rho}\right) \overset{(1.50)}{\ge} \log\left(\frac{1}{\epsilon}\right). \qquad (1.51)$$

Applying exponentials to the above inequality completes the proof. □

As an instance, of how the above lemma can be used, recall the convergence result of Theorem 3, where we have proved that SGD with constant stepsize $\omega \in (0,2)$ converges as follows:

$$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \left[1 - \omega(2-\omega)\lambda_{\min}^+\right]^k \|x^0 - x^*\|_{\mathbf{B}}^2.$$

In this setting, Lemma 7 can be utilized with $A^k = \mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2]$ and $\rho = 1 - \omega(2-\omega)\lambda_{\min}^+$ to obtain:

$$k \geq \frac{1}{\omega(2-\omega)\lambda_{\min}^+} \log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad \mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \epsilon \|x^0 - x^*\|_{\mathbf{B}}^2.$$

In this case we say that SGD converges with iteration complexity

$$\mathcal{O}\left(\frac{1}{\omega(2-\omega)\lambda_{\min}^+} \log\left(\frac{1}{\epsilon}\right)\right).$$

1.6 Structure of the Thesis

In the remainder of this section we give a summary of each chapter of this thesis. The detailed proofs and careful deductions of any claims made here are left to the chapters.

1.6.1 Chapter 2: Randomized Iterative Methods with Momentum and Stochastic Momentum

The baseline first-order method for minimizing a differentiable function f is the *gradient descent (GD)* method,

$$x^{k+1} = x^k - \omega^k \nabla f(x^k),$$

where $\omega^k > 0$ is a stepsize [21]. For convex functions with L-Lipschitz gradient, GD converges at at the rate of $\mathcal{O}(L/\epsilon)$. When, in addition, f is μ-strongly convex, the rate is linear: $\mathcal{O}((L/\mu)\log(1/\epsilon))$ [140]. To improve the convergence behavior of the method, Polyak proposed to modify GD by the introduction of a (heavy ball) momentum term[5], $\beta(x^k - x^{k-1})$ [156, 157]. This leads to the gradient descent method with momentum (mGD), popularly known as the *heavy ball method:*

$$x^{k+1} = x^k - \omega^k \nabla f(x^k) + \beta(x^k - x^{k-1}).$$

More specifically, Polyak proved that with the correct choice of the stepsize parameters ω^k and momentum parameter β, a *local* accelerated linear convergence rate of $\mathcal{O}(\sqrt{L/\mu}\log(1/\epsilon))$ can be achieved in the case of twice continuously differentiable, μ-strongly convex objective functions with L-Lipschitz gradient [156, 157].

The theoretical behavior of the above deterministic heavy ball method is now well understood in different settings. In contrast to this, there has been less progress in understanding the convergence behavior of *stochastic* variants of the heavy ball method. The key method in this category is stochastic gradient descent with momentum (mSGD; stochastic heavy ball method): $x^{k+1} = x^k - \omega^k g(x^k) + \beta(x^k - x^{k-1})$, where g is an unbiased estimator of the true gradient $\nabla f(x^k)$. In our setting, where our goal is to solve the stochastic optimization problem (1.6), mSGD takes the following form:

$$x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k) + \beta(x^k - x^{k-1}),$$

[5] A more popular, and certainly theoretically much better understood alternative to Polyak's momentum is the momentum introduced by Nesterov [138, 140], leading to the famous *accelerated gradient descent (AGD)* method. This method converges non-asymptotically and globally; with optimal sublinear rate $\mathcal{O}(\sqrt{L/\epsilon})$ [137] when applied to minimizing a smooth convex objective function (class $\mathcal{F}_{0,L}^{1,1}$), and with the optimal linear rate $\mathcal{O}(\sqrt{L/\mu}\log(1/\epsilon))$ when minimizing smooth strongly convex functions (class $\mathcal{F}_{\mu,L}^{1,1}$). Recently, variants of Nesterov's momentum have also been introduced for the acceleration of stochastic gradient descent. We refer the interested reader to [65, 1, 88, 87, 214, 215, 98] and the references therein. Both Nesterov's and Polyak's update rules are known in the literature as "momentum" methods. In Chapter 2, however, we focus exclusively on Polyak's heavy ball momentum.

where $\omega > 0$ denotes a fixed stepsize and matrix $\mathbf{S}_k$ is sampled afresh in each iteration from distribution $\mathcal{D}$.

In Chapter 2, we study several classes of stochastic optimization algorithms enriched with the *heavy ball momentum* for solving the three closely related problems already described in the introduction. These are the stochastic quadratic optimization problem (1.6), the best approximation problem (1.22) and the dual quadratic optimization problem (1.26). Among the methods studied are: stochastic gradient descent (1.17), stochastic Newton (1.18), stochastic proximal point (1.19) and stochastic dual subspace ascent (1.29). This is the first time momentum variants of several of these methods are studied. We prove global non-asymptotic linear convergence rates for all methods and various measures of success, including primal function values, primal iterates, and dual function values. We also show that the primal iterates converge at an accelerated linear rate in a somewhat weaker sense. This is the first time a linear rate is shown for the stochastic heavy ball method (i.e., stochastic gradient descent method with momentum). Under somewhat weaker conditions, we establish a sublinear convergence rate for Cesàro averages of primal iterates. Moreover, we propose a novel concept, which we call *stochastic momentum*, aimed at decreasing the cost of performing the momentum step. We prove linear convergence of several stochastic methods with stochastic momentum, and show that in some sparse data regimes and for sufficiently small momentum parameters, these methods enjoy better overall complexity than methods with deterministic momentum. Finally, we perform extensive numerical testing on artificial and real datasets.

1.6.2 Chapter 3: Inexact Randomized Iterative Methods

A common feature of existing randomized iterative methods is that in their update rule a particular subproblem needs to be solved *exactly*. In the large scale setting, often this step can be computationally very expensive. The purpose of the work in Chapter 3 is to reduce the cost of this step by incorporating *inexact updates* in the stochastic methods under study.

From a stochastic optimization viewpoint, we analyze the performance of inexact SGD (iSGD):

$$x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k) + \epsilon^k,$$

where $\omega > 0$ denotes a fixed stepsize, matrix $\mathbf{S}_k$ is sampled afresh in each iteration from distribution $\mathcal{D}$ and ϵ^k represents a (possibly random) error coming from inexact computations.

In Chapter 3, we propose and analyze *inexact* variants of the exact algorithms presented in previous sections for solving the stochastic optimization problem (1.6), the best approximation problem (1.22) and the dual problem (1.26). Among the methods studied are: stochastic gradient descent (SGD), stochastic Newton (SN), stochastic proximal point (SPP), sketch and project method (SPM) and stochastic subspace ascent (SDSA). In all of these methods, a certain potentially expensive calculation/operation needs to be performed in each step; it is this operation that we propose to be performed *inexactly*. For instance, in the case of SGD, it is the computation of the stochastic gradient $\nabla f_{\mathbf{S}_k}(x^k)$, in the case of SPM is the computation of the projection $\Pi_{\mathcal{L}_\mathbf{S},\mathbf{B}}(x^k)$, and in the case of SDSA it is the computation of the dual update $\mathbf{S}_k\lambda^k$.

We perform an iteration complexity analysis under an abstract notion of inexactness and also under a more structured form of inexactness appearing in practical scenarios. Typically, an inexact solution of these subproblems can be obtained much more quickly than the exact solution. Since in practical applications the savings thus obtained are larger than the increase in the number of iterations needed for convergence, our inexact methods can be dramatically faster.

Inexact variants of many popular and some more exotic methods, including randomized block Kaczmarz, randomized Gaussian Kaczmarz and randomized block coordinate descent, can be cast as special cases of our analysis. Finally, we present numerical experiments which demonstrate the benefits of allowing inexactness.

1.6.3 Chapter 4: Revisiting Randomized Gossip Algorithms

In Chapter 4 we present a new framework for the analysis and design of randomized gossip algorithms for solving the average consensus (AC) problem, a fundamental problem in distributed computing and multi-agent systems.

In the AC problem we are given an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E}$. Each node $i \in \mathcal{V}$ "knows" a private value $c_i \in \mathbb{R}$. The goal of AC is for every node to compute the average of these private values, $\bar{c} := \frac{1}{n}\sum_i c_i$, in a decentralized fashion. That is, the exchange of information can only occur between connected nodes (neighbors).

In an attempt to connect the AC problem to optimization, consider the simple optimization problem:

$$\min_{x=(x_1,\ldots,x_n)\in\mathbb{R}^n} \frac{1}{2}\|x - c\|^2 \quad \text{subject to} \quad x_1 = x_2 = \cdots = x_n, \tag{1.52}$$

where $c = (c_1, \ldots, c_n)^\top$ is the vector of the initial private values c_i. Observe that its optimal solution x^* must necessarily satisfy $x_i^* = \bar{c}$ for all $i \in [n]$, where $\bar{c}$ is the value that each node needs to compute in the AC problem. Note also that, if we represent the constraint $x_1 = x_2 = \cdots = x_n$ of (1.52) as a linear system then the optimization problem (1.52) is an instance of the best approximation problem (1.22) with $\mathbf{B} = \mathbf{I}$ (identity matrix). Perhaps, there is a deeper link here? Indeed, it turns out that properly chosen randomized algorithms for solving (1.52) can be interpreted as decentralized protocols for solving the AC problem.

A simple way to express the constraint of problem (1.52) as linear system $\mathbf{A}x = b$ is by selecting $\mathbf{A}$ to be the incidence matrix of the network and the right hand side to be the zero vector ($b = 0 \in \mathbb{R}^m$). By using this system the most basic randomized gossip algorithm ("randomly pick an edge $e = (i, j) \in \mathcal{E}$ and then replace the values stored at vertices i and j by their average") is an instance of the randomized Kaczmarz (RK) method (1.21) for solving consistent linear systems, applied to this system.

Using this observation as a starting point, in Chapter 4 we show how classical randomized iterative methods for solving linear systems can be interpreted as gossip algorithms when applied to special systems encoding the underlying network and explain in detail their decentralized nature. Our general framework recovers a comprehensive array of well-known gossip algorithms as special cases, including the pairwise randomized gossip algorithm and path averaging gossip, and allows for the development of provably faster variants. The flexibility of the new approach enables the design of a number of new specific gossip methods. For instance, we propose and analyze novel *block* and the first provably *accelerated* randomized gossip protocols, and *dual* randomized gossip algorithms.

From a numerical analysis viewpoint, our work is the first that explores in depth the decentralized nature of randomized iterative methods for linear systems and proposes them as methods for solving the average consensus problem.

We evaluate the performance of the proposed gossip protocols by performing extensive experimental testing on typical wireless network topologies.

1.6.4 Chapter 5: Privacy Preserving Randomized Gossip Algorithms

In Chapter 5, we present three different approaches to solving the Average Consensus problem while at the same time protecting the information about the initial values of the nodes. To the best of our knowledge, this work is the first which combines the *gossip framework* with the privacy concept of protection of the initial values.

The randomized methods we propose are all dual in nature. That is, they solve directly the dual problem (1.26). In particular, the three different techniques that we use for preserving the privacy are "Binary Oracle", "ϵ-Gap Oracle" and "Controlled Noise Insertion".

Binary Oracle: We propose to reduce the amount of information transmitted in each iteration to a single bit. More precisely, when an edge is selected, each corresponding node will only receive information whether the value on the other node is smaller or larger. Instead of setting the value on the selected nodes to their average, each node increases or decreases its value by a pre-specified amount.

ϵ-Gap Oracle: In this case, we have an oracle that returns one of three options and is parametrized by ϵ. If the difference in values of sampled nodes is larger than ϵ, an update similar to the one in Binary Oracle is taken. Otherwise, the values remain unchanged. An advantage compared to the Binary Oracle is that this approach will converge to a certain accuracy and stop there, determined by ϵ (Binary Oracle will oscillate around optimum for a fixed stepsize). However, in general, it will disclose more information about the initial values.

Controlled Noise Insertion: This approach protects the initial values by inserting noise in the process. Broadly speaking, in each iteration, each of the sampled nodes first adds a noise to its current value, and an average is computed afterwards. Convergence is guaranteed due to the correlation in the noise across iterations. Each node remembers the noise it added last time it was sampled, and in the following iteration, the previously added noise is first subtracted, and a fresh noise of smaller magnitude is added. Empirically, the protection of initial values is provided by first injecting noise into the system, which propagates across the network, but is gradually withdrawn to ensure convergence to the true average.

We give iteration complexity bounds for all proposed privacy preserving randomized gossip algorithms and perform extensive numerical experiments.

1.7 Summary

The content of this thesis is based on the following publications and preprints:

Chapter 2

- Nicolas Loizou and Peter Richtárik."Momentum and Stochastic Momentum for Stochastic Gradient, Newton, Proximal Point and Subspace Descent Methods", arXiv preprint arXiv:1712.09677 (2017). [115]

- Nicolas Loizou and Peter Richtárik. "Linearly Convergent Stochastic Heavy Ball Method for Minimizing Generalization Error", Workshop on Optimization for Machine Learning, NIPS 2017. [114]

Chapter 3

- Nicolas Loizou and Peter Richtárik. "Convergence Analysis of Inexact Randomized Iterative Methods", arXiv preprint arXiv:1903.07971 (2019). [117]

Chapter 4

- Nicolas Loizou and Peter Richtárik. "A New Perspective on Randomized Gossip Algorithms", IEEE Global Conference on Signal and Information Processing (GlobalSIP), pp.440-444, 2016 [113]

- Nicolas Loizou and Peter Richtárik. "Accelerated Gossip via Stochastic Heavy Ball Method." 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton) (pp. 927-934), 2018. [116]

- Nicolas Loizou, Mike Rabbat and Peter Richtárik. "Provably Accelerated Randomized Gossip Algorithms" 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7505-7509 [112]

- Nicolas Loizou and Peter Richtárik. "Revisiting Randomized Gossip Algorithms: General Framework, Convergence Rates and Novel Block and Accelerated Protocols", arXiv preprint arXiv:1905.08645, [118].

Chapter 5

- Filip Hanzely, Jakub Konečný, Nicolas Loizou, Peter Richtárik and Dmitry Grishchenko. "A Privacy Preserving Randomized Gossip Algorithm via Controlled Noise Insertion", NeurIPS 2018 - Privacy Preserving Machine Learning Workshop, [80]

- Filip Hanzely, Jakub Konečný, Nicolas Loizou, Peter Richtárik and Dmitry Grishchenko. "Privacy Preserving Randomized Gossip Algorithms", arXiv preprint arXiv:1706.07636, 2017 [79]

During the course of my study, I also co-authored the following works which were not used in the formation of this thesis:

- Mahmoud Assran, Nicolas Loizou, Nicolas Ballas and Mike Rabbat. "Stochastic Gradient Push for Distributed Deep Learning", Proceedings of the 36th International Conference on Machine Learning (ICML), 2019 [4]

- Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin and Peter Richtárik. "SGD: General Analysis and Improved Rates" Proceedings of the 36th International Conference on Machine Learning (ICML), 2019 [68]

- Nicolas Loizou. "Distributionally Robust Games with Risk-Averse Players", In Proceedings of 5th International Conference on Operations Research and Enterprise Systems (ICORES), 186-196, 2016 [111]

In [68], we propose a general theory describing the convergence of Stochastic Gradient Descent (SGD) under the "arbitrary sampling paradigm". Our theory describes the convergence of an infinite array of variants of SGD, each of which is associated with a specific probability law governing the data selection rule used to form minibatches. This is the first time such an analysis is performed, and most of our variants of SGD were never explicitly considered in the literature before.

In [4], we study Stochastic Gradient Push (SGP), an algorithm which combines PushSum gossip protocol with stochastic gradient updates for distributed deep learning. We prove that SGP converges to a stationary point of smooth, non-convex objectives at the same sub-linear rate as SGD, that all nodes achieve consensus, and that SGP achieves a linear speedup with respect to the number of compute nodes. Furthermore, we empirically validate the performance of SGP on image classification (ResNet-50, ImageNet) and machine translation (Transformer, WMT16 En- De) workloads.

In [111] we present a new model of incomplete information games without private information in which the players use a distributionally robust optimization approach to cope with the payoff uncertainty.

Chapter 2

Randomized Iterative Methods with Momentum and Stochastic Momentum

2.1 Introduction

Two of the most popular algorithmic ideas for solving optimization problems involving big volumes of data are *stochastic approximation* and *momentum*. By stochastic approximation we refer to the practice pioneered by Robins and Monro [169] of replacement of costly-to-compute quantities (e.g., gradient of the objective function) by cheaply-to-compute *stochastic* approximations thereof (e.g., unbiased estimate of the gradient). By momentum we refer to the *heavy ball* technique originally developed by Polyak [156] to accelerate the convergence rate of gradient-type methods.

While much is known about the effects of *stochastic approximation* and *momentum* in isolation, surprisingly little is known about the *combined effect* of these two popular algorithmic techniques. For instance, to the best of our knowledge, there is no context in which a method combining stochastic approximation with momentum is known to have a linear convergence rate. One of the contributions of this work is to show that there are important problem classes for which a linear rate can indeed be established for a range of stepsize and momentum parameters.

2.1.1 The setting

In this chapter we study the three closely related problems described in the introduction of this thesis. These are:

(i) stochastic quadratic optimization (1.6),

(ii) best approximation (1.22), and

(iii) (bounded) concave quadratic maximization (1.26).

In particular we are interested in the complexity analysis and efficient implementation of momentum variants of the baseline algorithms presented in the introduction. As a reminder, these methods are the following: stochastic gradient descent (SGD), stochastic Newton method (SN), stochastic proximal point methods (SPP), sketch and project method (SPM) and stochastic dual subspace ascent (SDSA).

We are not aware of any successful attempts to analyze momentum variants of SN and SPP, SPM and SDSA and to the best of our knowledge there are no linearly convergent variants of SGD with momentum in any setting.

In addition, we propose and analyze a novel momentum strategy for SGD, SN, SPP and SPM, which we call *stochastic momentum*. It is a stochastic approximation of the popular deterministic heavy ball momentum which in some situations could be particularly beneficial

in terms of overall complexity. Similar to the classical momentum, we prove linear convergence rates for this momentum strategy.

2.1.2 Structure of the chapter

This chapter is organized as follows. In Section 2.2 we summarize our contributions in the context of existing literature. In Section 2.3 we describe and analyze primal methods with momentum (mSGD, mSN and mSPP), and in Section 2.4 we describe and analyze the dual method with momentum (mSDSA). In Section 2.5 we describe and analyze primal methods with stochastic momentum (smSGD, smSN and smSPP). Numerical experiments are presented in Section 2.7. Proofs of all key results can be found in Section 2.9.

2.1.3 Notation

The following notational conventions are used in this chapter. Boldface upper-case letters denote matrices; $\mathbf{I}$ is the identity matrix. By $\mathcal{L}$ we denote the solution set of the linear system $\mathbf{A}x = b$. By $\mathcal{L}_{\mathbf{S}}$, where $\mathbf{S}$ is a random matrix, we denote the solution set of the *sketched* linear system $\mathbf{S}^\top \mathbf{A}x = \mathbf{S}^\top b$. By $\mathbf{A}_{i:}$ and $\mathbf{A}_{:j}$ we indicate the i_{th} row and the j_{th} column of matrix $\mathbf{A}$, respectively. Unless stated otherwise, throughout the chapter, x^* is the projection of x^0 onto $\mathcal{L}$ in the $\mathbf{B}$-norm: $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. We also write $[n] := \{1, 2, \ldots, n\}$. Finally, we say that a function $f : \mathbb{R}^n \to \mathbb{R}$ belongs to the class $\mathcal{F}^{1,1}_{0,L}$ if it is convex, continuously differentiable, and its gradient is Lipschitz continuous with constant L. If in addition the function f is μ-strongly convex with strong convexity constant $\mu > 0$, then we say that it belongs to the class $\mathcal{F}^{1,1}_{\mu,L}$. When it is also twice continuously differentiable, it belongs to the function class $\mathcal{F}^{2,1}_{\mu,L}$.

2.2 Momentum Methods and Main Contributions

In this section we give a brief review of the relevant literature, and provide a summary of our contributions.

2.2.1 Heavy ball method

As we have already mentioned in Section 1.6, Polyak's seminal work [156, 157] showed that deterministic heavy ball method:

$$x^{k+1} = x^k - \omega^k \nabla f(x^k) + \beta(x^k - x^{k-1}),$$

converges with a *local* accelerated linear convergence rate of $\mathcal{O}(\sqrt{L/\mu}\log(1/\epsilon))$ in the case of twice continuously differentiable, μ-strongly convex objective functions with L-Lipschitz gradient (function class $\mathcal{F}^{2,1}_{\mu,L}$).

Recently, Ghadimi et al. [62] performed a *global* convergence analysis for the heavy ball method. In particular, the authors showed that for a certain combination of the stepsize and momentum parameter, the method converges sublinearly to the optimum when the objective function is convex and has Lipschitz gradient ($f \in \mathcal{F}^{1,1}_{0,L}$), and linearly when the function is also strongly convex ($f \in \mathcal{F}^{1,1}_{\mu,L}$). A particular selection of the parameters ω and β that gives the desired accelerated linear rate was not provided.

To the best of our knowledge, despite considerable amount of work on the heavy ball method, there is still no global convergence analysis which would guarantee an accelerated linear rate for $f \in \mathcal{F}^{1,1}_{\mu,L}$. However, in the special case of a strongly convex quadratic, an elegant proof was recently proposed in [103]. Using the notion of integral quadratic constraints from robust control theory, the authors proved that by choosing $\omega^k = \omega = 4/(\sqrt{L}+\sqrt{\mu})^2$ and $\beta = (\sqrt{L/\mu}-1)^2/(\sqrt{L/\mu}+1)^2$, the heavy ball method enjoys a global *asymptotic* accelerated convergence rate of $\mathcal{O}(\sqrt{L/\mu}\log(1/\epsilon))$. The aforementioned results are summarized in the first part of Table 2.1.

Extensions of the heavy ball method have been recently proposed in the proximal setting [145], non-convex setting [146, 210] and for distributed optimization [63]. For more recent analysis under several combinations of assumptions we suggest [184, 183, 100].

2.2.2 Stochastic heavy ball method

In contrast to the recent advances in our theoretical understanding of the (classical) heavy ball method, there has been less progress in understanding the convergence behavior of *stochastic* variants of the heavy ball method. The key method in this category is stochastic gradient descent with momentum (mSGD; aka: stochastic heavy ball method):

$$x^{k+1} = x^k - \omega^k g(x^k) + \beta(x^k - x^{k-1}),$$

where g is an unbiased estimator of the true gradient $\nabla f(x^k)$. While mSGD is used extensively in practice, especially in deep learning [185, 186, 99, 196], its convergence behavior is not very well understood.

In fact, we are aware of only two papers, both recent, which set out to study the complexity of mSGD: the work of Yang et al. [206], and the work of Gadat et al. [60]. In the former paper, a unified convergence analysis for stochastic gradient methods with momentum (heavy ball and Nesterov's momentum) was proposed; and an analysis for both convex and non convex functions was performed. For a general Lipschitz continuous convex objective function with bounded variance, a rate of $\mathcal{O}(1/\sqrt{k})$ was proved. For this, the authors employed a decreasing stepsize strategy: $\omega^k = \omega^0/\sqrt{k+1}$, where ω^0 is a positive constant. In [60], the authors first describe several almost sure convergence results in the case of general non-convex coercive functions, and then provide a complexity analysis for the case of quadratic strongly convex function. However, the established rate is slow. More precisely, for strongly convex quadratic and coercive functions, mSGD with diminishing stepsizes $\omega^k = \omega^0/k^\beta$ was shown to convergence as $\mathcal{O}(1/k^\beta)$ when the momentum parameter is $\beta < 1$, and with the rate $\mathcal{O}(1/\log k)$ when $\beta = 1$. The convergence rates established in both of these papers are sublinear. In particular, no insight is provided into whether the inclusion of the momentum term provides what it was aimed to provide: acceleration.

The above results are summarized in the second part of Table 2.1. From this perspective, our contribution lies in providing an in-depth analysis of mSGD (and, additionally, of SGD with stochastic momentum).

Many recent papers have built upon our analysis [115, 114] and have already extended our results in several settings. For more details see [18, 121, 40, 3, 41].

On definitions of convergence presented in Table 2.1. In Table 2.1 we present two main notions to characterize the convergence guarantees presented in the literature for the analysis of deterministic and stochastic heavy ball methods. These are, (i) Local/ Global convergence and (ii) Asymptotic/ Non-asymptotic convergence. For clarity, in this paragraph, we present these definitions of convergence.

Local convergence we have only if the convergence guarantees depend on the starting point of the method. That is, if we can guarantee convergence only if x^0 is in a neighborhood of the optimal point x^*. If the method convergence for any starting point then we have *global convergence.*

We have *asymptotic convergence* when the provided rate can be shown to hold only after specific number of iterations. For example, we say that a deterministic method converges with asymptotic linear rate if there is $K > 0$ such that for $k > K$ we have $\|x^k - x^*\|^2 < \rho^k \|x^0 - x^*\|$, where $\rho \in (0, 1)$. We have a *non-asymptotic convergence* if the rate satisfy the above definition for $K = 0$.

2.2.3 Connection to incremental gradient methods

Assuming $\mathcal{D}$ is discrete distribution (i.e., we sample from M matrices, $\mathbf{S}^1, \ldots, \mathbf{S}^M$, where $\mathbf{S}^i$ is chosen with probability $p_i > 0$. Here, $0 < M \in R$ is fixed.), we can write the stochastic

Method	Paper	Rate	Assumptions on f	Convergence
Heavy Ball (mGD)	Polyak, 1964 [156]	accelerated linear	$\mathcal{F}^{2,1}_{\mu,L}$	local
	Ghadimi et al, 2014 [62]	sublinear	$\mathcal{F}^{1,1}_{0,L}$	global
	Ghadimi et al, 2014 [62]	linear	$\mathcal{F}^{1,1}_{\mu,L}$	global
	Lessard et al, 2016 [103]	accelerated linear	$\mathcal{F}^{1,1}_{\mu,L}$ + quadratic	global, asymptotic
Stochastic Heavy Ball (mSGD)	Yang et al. 2016 [206]	sublinear	$\mathcal{F}^{1,1}_{0,L}$ + bounded variance	global, non-asymptotic
	Gadat et al, 2016 [60]	sublinear	$\mathcal{F}^{1,1}_{\mu,L}$ + other assumptions	global, non-asymptotic
	THIS CHAPTER	**see Table 2.3**	$\mathcal{F}^{1,1}_{0,L}$ + quadratic	global, non-asymptotic

Table 2.1: Known complexity results for gradient descent with momentum (mGD, aka: heavy ball method), and stochastic gradient descent with momentum (mSGD, aka: stochastic heavy ball method). We give the first linear and accelerated rates for mSGD. For full details on iteration complexity results we obtain, refer to Table 2.3.

optimization problem (1.6) in the *finite-sum* form

$$\min_{x\in\mathbb{R}^n} f(x) = \sum_{i=1}^{M} p_i f_{\mathbf{S}^i}(x). \tag{2.1}$$

Choosing $x^0 = x^1$, mSGD with fixed stepsize $\omega^k = \omega$ applied to (2.1) can be written in the form

$$x^{k+1} = x^k - \omega \sum_{t=1}^{k} \beta^{k-t} \nabla f_{\mathbf{S}_t}(x^t) + \beta^k (x^1 - x^0) \overset{x^0 \equiv x^1}{=} x^k - \omega \sum_{t=1}^{k} \beta^{k-t} \nabla f_{\mathbf{S}_t}(x^t), \tag{2.2}$$

where $\mathbf{S}_t = \mathbf{S}^i$ with probability p_i. Problem (2.1) can be also solved using incremental average/aggregate gradient methods, such as the IAG method of Blatt et al. [13]. These methods have a similar form to (2.2); however the past gradients are aggregated somewhat differently. While (2.2) uses a geometric weighting of the gradients, the incremental average gradient methods use a uniform/arithmetic weighting. The stochastic average gradient (SAG) method of Schmidt et al. [173] can be also written in a similar form. Note that mSGD uses a geometric weighting of previous gradients, while the the incremental and stochastic average gradient methods use an arithmetic weighting. Incremental and incremental average gradient methods are widely studied algorithms for minimizing objective functions which can expressed as a sum of finite convex functions. For a review of key works on incremental methods and a detailed presentation of the connections with stochastic gradient descent, we refer the interested reader to the excellent survey of Bertsekas [11]; see also the work of Tseng [188].

In [77], an incremental average gradient method with momentum was proposed for minimizing strongly convex functions. It was proved that the method converges to the optimum with linear rate. The rate is always worse than that of the no-momentum variant. However, it was shown experimentally that in practice the method is faster, especially in problems with high condition number. In our setting, the objective function has a very specifc structure (1.6). It is not a finite sum problem as the distribution $\mathcal{D}$ could be continous; and we also do not assume strong convexity. Thus, the convergence analysis of [77] can not be directly applied to our problem.

2.2.4 Summary of contributions

We now summarize the contributions of this chapter.

New momentum methods. We study several classes of stochastic optimization algorithms (SGD, SN, SPP and SDSA) *with momentum*, which we call mSGD, mSN, mSPP and mSDSA, respectively (see the first and second columns of Table 2.2). We do this in a simplified setting with quadratic objectives where all of these algorithms are equivalent. These methods can be seen as solving three related optimization problems: the stochastic optimization problem (1.6), the best approximation problem (1.22) and its dual. To the best of our knowledge, momentum variants of SN, SPP and SDSA were not analyzed before.

no momentum ($\beta = 0$)	momentum ($\beta \geq 0$)	stochastic momentum ($\beta \geq 0$)
SGD [73, $\omega = 1$], [168, $\omega > 0$] $x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k)$	**mSGD** [Sec 2.3] $+\beta(x^k - x^{k-1})$	**smSGD** [Sec 2.5] $+n\beta e_{i_k}^\top (x^k - x^{k-1}) e_{i_k}$
SN [168] $x^{k+1} = x^k - \omega(\nabla^2 f_{\mathbf{S}_k}(x^k))^{\dagger_{\mathbf{B}}} \nabla f_{\mathbf{S}_k}(x^k)$	**mSN** [Sec 2.3] $+\beta(x^k - x^{k-1})$	**smSN** [Sec 2.5] $+n\beta e_{i_k}^\top (x^k - x^{k-1}) e_{i_k}$
SPP [168] $x^{k+1} = \arg\min_x \left\{ f_{\mathbf{S}_k}(x) + \frac{1-\omega}{2\omega} \|x - x^k\|_{\mathbf{B}}^2 \right\}$	**mSPP** [Sec 2.3] $+\beta(x^k - x^{k-1})$	**smSPP** [Sec 2.5] $+n\beta e_{i_k}^\top (x^k - x^{k-1}) e_{i_k}$
SDSA [74, $\omega = 1$] $y^{k+1} = y^k + \mathbf{S}_k \lambda^k$	**mSDSA** [Sec 2.4] $+\beta(y^k - y^{k-1})$	

Table 2.2: All methods analyzed in this chapter. The methods highlighted in bold (with momentum and stochastic momentum) are new. SGD = Stochastic Gradient Descent, SN = Stochastic Newton, SPP = Stochastic Proximal Point, SDSA = Stochastic Dual Subspace Ascent. At iteration k, matrix $\mathbf{S}_k$ is drawn in an i.i.d. fashion from distribution $\mathcal{D}$, and a stochastic step is performed.

Algorithm	ω	momentum β	Quantity converging to 0	Rate (all: global, non-asymptotic)	Theorem
mSGD/mSN/mSPP	$(0, 2)$	≥ 0	$\mathbb{E}\left[\|x^k - x^*\|_{\mathbf{B}}^2\right]$	linear	8
mSGD/mSN/mSPP	$(0, 2)$	≥ 0	$\mathbb{E}[f(x^k) - f(x^*)]$	linear	8
mSGD/mSN/mSPP	$(0, 2)$	≥ 0	$\mathbb{E}\left[f(\hat{x}^k) - f(x^*)\right]$	sublinear: $\mathcal{O}(1/k)$	10
mSGD/mSN/mSPP	1	$\left(1 - \sqrt{0.99\lambda_{\min}^+}\right)^2$	$\|\mathbb{E}[x^k - x^*]\|_{\mathbf{B}}^2$	accelerated linear	11
mSGD/mSN/mSPP	$\frac{1}{\lambda_{\max}}$	$\left(1 - \sqrt{0.99\frac{\lambda_{\min}^+}{\lambda_{\max}}}\right)^2$	$\|\mathbb{E}[x^k - x^*]\|_{\mathbf{B}}^2$	accelerated linear (better than for $\omega = 1$)	11
mSDSA	$(0, 2)$	≥ 0	$\mathbb{E}\left[D(y^*) - D(y^k)\right]$	linear	13
smSGD/smSN/smSPP	$(0, 2)$	≥ 0	$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2]$	linear	14
smSGD/smSN/smSPP	$(0, 2)$	≥ 0	$\mathbb{E}[f(x^k) - f(x^*)]$	linear	14

Table 2.3: Summary of the iteration complexity results obtained in this chapter. Parameters of the methods: ω (stepsize) and β (momentum term). In all cases, $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ is the solution of the best approximation problem. Theorem 10 refers to Cesàro averages: $\hat{x}^k = \frac{1}{k}\sum_{t=0}^{k-1} x^t$. Theorem 13 refers to suboptimality in dual function values (D is the dual function).

Linear rate. We prove several (global and non-asymptotic) linear convergence results for our primal momentum methods mSGD/mSN/mSPP. First, we establish a linear rate for the decay of $\mathbb{E}\left[\|x^k - x^*\|_{\mathbf{B}}^2\right]$ to zero, for a range of stepsizes $\omega > 0$ and momentum parameters $\beta \geq 0$. We show that the same rate holds for the decay of the expected function values $\mathbb{E}\left[f(x^k) - f(x^*)\right]$ of (1.6) to zero. Further, the same rate holds for mSDSA, in particular, this is for the convergence of the dual objective to the optimum. For a summary of these results, and pointers to the relevant theorems, refer to lines 1, 2 and 6 of Table 2.3. Unfortunately, the theoretical rate for all our momentum methods is optimized for $\beta = 0$, and gets worse as the momentum parameter increases. However, no prior linear rate for any of these methods with momentum are known. We give the first linear convergence rate for SGD with momentum (i.e., for the stochastic heavy ball method).

Accelerated linear rate. We then study the decay of the larger quantity $\|\mathbb{E}\left[x^k - x^*\right]\|_{\mathbf{B}}^2$ to zero. In this case, we establish an *accelerated* linear rate, which depends on the square root of the condition number (of the Hessian of f). This is a quadratic speedup when compared to the no-momentum methods as these depend on the condition number. See lines 4 and 5 of Table 2.3. To the best of our knowledge, this is the first time an accelerated rate is obtained for the stochastic heavy ball method (mSGD). Note that there are no global non-asymptotic accelerated linear rates proved even in the non-stochastic setting (i.e., for the heavy ball method). Moreover, we are not aware of any accelerated linear convergence results for the stochastic proximal point method.

Sublinear rate for Cesàro averages. We show that the Cesàro averages, $\hat{x}^k = \frac{1}{k}\sum_{t=0}^{k-1} x^t$, of all primal momentum methods enjoy a sublinear $\mathcal{O}(1/k)$ rate (see line 3 of Table 2.3). This holds under weaker assumptions than those which lead to the linear convergence rate.

Primal-dual correspondence. We show that SGD, SN and SPP with momentum arise as affine images of SDSA with momentum (see Theorem 12). This extends the result of [74] where this was shown for the no-momentum methods ($\beta = 0$) and in the special case of the unit stepsize ($\omega = 1$).

Stochastic momentum. We propose a new momentum strategy, which we call *stochastic momentum*. Stochastic momentum is a stochastic (coordinate-wise) approximation of the deterministic momentum, and hence is much less costly, which in some situations leads to computational savings in each iteration. On the other hand, the additional noise introduced this way increases the number of iterations needed for convergence. We analyze the SGD, SN and SPP methods with stochastic momentum, and prove linear convergence rates. We prove that in some settings the overall complexity of SGD with stochastic momentum is better than the overall complexity of SGD with momentum. For instance, this is the case if we consider the randomized Kaczmarz (RK) method as a special case of SGD, and if $\mathbf{A}$ is sparse.

Space for generalizations. We hope that the present work can serve as a starting point for the development of SN, SPP and SDSA methods with momentum for more general classes (beyond special quadratics) of convex and perhaps also nonconvex optimization problems. In such more general settings, however, the symmetry which implies equivalence of these algorithms will break, and hence a different analysis will be needed for each method.

2.3 Primal Methods with Momentum

Applied to problem (1.6), i.e., $\min_{x\in\mathbb{R}^n} f(x) = \mathbb{E}\left[f_{\mathbf{S}}(x)\right]$, the gradient descent method with momentum (also known as the heavy ball method) of Polyak [156, 157] takes the form

$$x^{k+1} = x^k - \omega\nabla f(x^k) + \beta(x^k - x^{k-1}), \tag{2.3}$$

where $\omega > 0$ is a stepsize and $\beta \geq 0$ is a momentum parameter. Instead of marrying the momentum term with gradient descent, we can marry it with SGD. This leads to SGD with momentum (mSGD), also known as the *stochastic heavy ball method*:

$$x^{k+1} = x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \beta(x^k - x^{k-1}). \tag{2.4}$$

Since SGD is equivalent to SN and SPP, this way we obtain momentum variants of the stochastic Newton (mSN) and stochastic proximal point (mSPP) methods. The method is formally described below:

Algorithm 1 mSGD / mSN / mSPP

Input: Distribution $\mathcal{D}$ from which method samples matrices; positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$; stepsize/relaxation parameter $\omega \in \mathbb{R}$; the heavy ball/momentum parameter β.

Initialize: Choose initial points $x^0, x^1 \in \mathbb{R}^n$

1: **for** $k = 1, 2, \cdots$ **do**
2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
3: Set $x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k) + \beta(x^k - x^{k-1})$
4: **end for**
5: **Output:** The last iterate x^k

To the best of our knowledge, momentum variants of SN and SPP were not considered in the literature before. Moreover, as far as we know, there are no momentum variants of even deterministic variants of (1.18), (1.19) and (1.25), such as incremental or batch Newton method, incremental or batch proximal point method and incremental or batch projection method; not even for a problem formulated differently.

In the rest of this section we state our convergence results for mSGD/mSN/mSPP.

2.3.1 Convergence of iterates and function values: linear rate

In this section we study the convergence rate of the quantity $\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}]$ to zero for mSGD/mSN/mSPP. We show that for a range of stepsize parameters $\omega > 0$ and momentum terms $\beta \geq 0$, the method enjoys global linear convergence rate; see (2.5). To the best of our knowledge, these results are the first of their kind for the stochastic heavy ball method. As a corollary of this result, we obtain convergence of the expected function values; see (2.6).

Theorem 8. *Choose $x^0 = x^1 \in \mathbb{R}^n$. Assume exactness. Let $\{x^k\}_{k=0}^\infty$ be the sequence of random iterates produced by mSGD/mSN/mSPP. Assume $0 < \omega < 2$ and $\beta \geq 0$ and that the expressions*

$$a_1 := 1 + 3\beta + 2\beta^2 - (\omega(2-\omega) + \omega\beta)\lambda^+_{\min}, \qquad \text{and} \qquad a_2 := \beta + 2\beta^2 + \omega\beta\lambda_{\max}$$

satisfy $a_1 + a_2 < 1$. Let $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Then

$$\mathbb{E}\left[\|x^k - x^*\|^2_{\mathbf{B}}\right] \leq q^k(1+\delta)\|x^0 - x^*\|^2_{\mathbf{B}} \tag{2.5}$$

and

$$\mathbb{E}\left[f(x^k)\right] \leq q^k \frac{\lambda_{\max}}{2}(1+\delta)\|x^0 - x^*\|^2_{\mathbf{B}}, \tag{2.6}$$

where $q = \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ and $\delta = q - a_1$. Moreover, $a_1 + a_2 \leq q < 1$.

Proof. See Section 2.9.2. □

In the above theorem we obtain a global linear rate. To the best of our knowledge, this is the first time that linear rate is established for a stochastic variant of the heavy ball method (mSGD) in any setting. All existing results are sublinear. These seem to be the first momentum variants of SN and SPP methods.

If we choose $\omega \in (0, 2)$, then the condition $a_1 + a_2 < 1$ is satisfied for all

$$0 \leq \beta < \frac{1}{8}\left(-4 + \omega\lambda^+_{\min} - \omega\lambda_{\max} + \sqrt{(4 - \omega\lambda^+_{\min} + \omega\lambda_{\max})^2 + 16\omega(2-\omega)\lambda^+_{\min}}\right). \tag{2.7}$$

If $\beta = 0$, mSGD reduces to SGD analyzed in [168]. In this special case, $q = 1 - \omega(2-\omega)\lambda^+_{\min}$, which is the rate established in [168]. Hence, our result is more general.

Let $q(\beta)$ be the rate as a function of β. Note that since $\beta \geq 0$, we have

$$\begin{aligned} q(\beta) &\geq a_1 + a_2 \\ &= 1 + 4\beta + 4\beta^2 + \omega\beta(\lambda_{\max} - \lambda^+_{\min}) - \omega(2-\omega)\lambda^+_{\min} \\ &\geq 1 - \omega(2-\omega)\lambda^+_{\min} = q(0). \end{aligned} \tag{2.8}$$

Clearly, the lower bound on q is an increasing function of β. Also, for any β the rate is always inferior to that of SGD ($\beta = 0$). It is an open problem whether one can prove a strictly better rate for mSGD than for SGD.

Our next proposition states that $\Pi_{\mathcal{L},\mathbf{B}}(x^k) = x^*$ (recall that $x^* := \Pi_{\mathcal{L},\mathbf{B}}(x^0)$) for all iterations k of mSGD. This invariance property plays an important role in our convergence analysis, and "explains" why the algorithm to converges to x^*.

Proposition 9. *Let $x^0 = x^1 \in \mathbb{R}^n$ be the starting points of the mSGD method and let $\{x^k\}$ be the random iterates generated by mSGD. Then $\Pi_{\mathcal{L},\mathbf{B}}(x^k) = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ for all $k \geq 0$.*

Proof. Note that in view of (1.7), $\nabla f_{\mathbf{S}}(x) = \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{H}(\mathbf{A}x - b) \in \text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)$. Since

$$x^{k+1} = x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \beta(x^k - x^{k-1}),$$

and since $x^0 = x^1$, it can shown by induction that $x^k \in x^0 + \text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)$ for all k. However, $\text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)$ is the orthogonal complement to $\text{Null}(\mathbf{A})$ in the $\mathbf{B}$-inner product. Since $\mathcal{L}$ is parallel to $\text{Null}(\mathbf{A})$, vectors x^k must have the same $\mathbf{B}$-projection onto $\mathcal{L}$ for all k: $\Pi_{\mathcal{L},\mathbf{B}}(x^0) = x^*$. □

This property also intuitively explains why mSGD converges to the projection of the *starting* point onto $\mathcal{L}$. Indeed, one may ask: why is the starting point special? After all, each iterate depends on the previous two iterates only, and all older iterates, including the starting point x^0, seem to be eventually "forgotten". Still, the iterative process has the property that all iterates live in the affine space passing through x^0 and orthogonal to $\mathcal{L}$, which means that the projection of *all* iterates onto $\mathcal{L}$ is identical.

2.3.2 Cesàro average: sublinear rate without exactness assumption

In this section we present the convergence analysis of the function values computed on the Cesàro average. Again our results are global in nature. To the best of our knowledge these are the first results that show $\mathcal{O}(1/k)$ convergence of the stochastic heavy ball method. Existing results apply in more general settings at the expense of slower rates. In particular, [206] and [60] get $\mathcal{O}(1/\sqrt{k})$ and $\mathcal{O}(1/k^\beta)$ convergence when $\beta \in (0, 1)$, respectively. When $\beta = 1$, [60] gets $\mathcal{O}(1/\log(k))$ rate.

Theorem 10. *Choose $x^0 = x^1$ and let $\{x^k\}_{k=0}^\infty$ be the random iterates produced by mSGD/mSN/mSPP, where the momentum parameter $0 \leq \beta < 1$ and relaxation parameter (stepsize) $\omega > 0$ satisfy $\omega + 2\beta < 2$. Let x^* be any vector satisfying $f(x^*) = 0$. If we let $\hat{x}^k = \frac{1}{k}\sum_{t=1}^k x^t$, then*

$$\mathbb{E}\left[f(\hat{x}^k)\right] \leq \frac{(1-\beta)^2\|x^0 - x^*\|_{\mathbf{B}}^2 + 2\omega\beta f(x^0)}{2\omega(2 - 2\beta - \omega)k}.$$

Proof. See Section 2.9.3. □

In the special case of $\beta = 0$, the above theorem gives the rate

$$\mathbb{E}\left[f(\hat{x}^k)\right] \leq \frac{\|x^0 - x^*\|_{\mathbf{B}}^2}{2\omega(2 - \omega)k}.$$

This is the convergence rate for Cesàro averages of the "basic method" (i.e., SGD) established in [168].

Our proof strategy is similar to [62] in which the first global convergence analysis of the (deterministic) heavy ball method was presented. There it was shown that when the objective function has a Lipschitz continuous gradient, the Cesàro averages of the iterates converge to the optimum at a rate of $\mathcal{O}(1/k)$. To the best of our knowledge, there are no results in the literature that prove the same rate of convergence in the stochastic case for any class of objective functions.

In [206] the authors analyzed mSGD for general Lipshitz continuous convex objective functions (with bounded variance) and proved the *sublinear* rate $\mathcal{O}(1/\sqrt{k})$. In [60], a complexity

analysis is provided for the case of quadratic strongly convex smooth coercive functions. A sublinear convergence rate of $\mathcal{O}(1/k^{\beta})$, where $\beta \in (0,1)$, was proved. In contrast to our results, where we assume fixed stepsize ω, both papers analyze mSGD with diminishing stepsizes.

2.3.3 Accelerated linear rate for expected iterates

In this section we show that by a proper combination of the relaxation (stepsize) parameter ω and the momentum parameter β, mSGD/mSN/mSPP enjoy an *accelerated* linear convergence rate in mean. That is, while SGD needs $\mathcal{O}(\theta \log(1/\epsilon))$ iterations to find x^k such that $\|\mathbb{E}\left[x^k\right] - x^*\|_{\mathbf{B}}^2 \leq \epsilon$ [168], mSGD only needs $\mathcal{O}(\sqrt{\theta}\log(1/\epsilon))$ iterations (see Theorem 11(ii)), where $\theta = \lambda_{\max}/\lambda_{\min}^+$. The word *acceleration* typically refers to improvement from a leading factor of θ to $\sqrt{\theta}$, which is significant in the ill-conditioned case, i.e., when θ is very large. In other words, the linear rate $(1-\sqrt{0.99/\theta})^k$ of mSGD is much better than linear rate $(1-1/\theta)^k$ of SGD, which in view of (2.8) is better than the linear rate of mSGD established in (2.5) (for a different quantity converging to zero) .

Theorem 11. *Assume exactness. Let $\{x^k\}_{k=0}^{\infty}$ be the sequence of random iterates produced by mSGD / mSN / mSPP, started with $x^0, x^1 \in \mathbb{R}^n$ satisfying the relation $x^0 - x^1 \in \text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)$, with relaxation parameter (stepsize) $0 < \omega \leq 1/\lambda_{\max}$ and momentum parameter $(1-\sqrt{\omega\lambda_{\min}^+})^2 < \beta < 1$. Let $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Then there exists constant $C > 0$ such that for all $k \geq 0$ we have*

$$\left\|\mathbb{E}\left[x^k - x^*\right]\right\|_{\mathbf{B}}^2 \leq \beta^k C.$$

(i) If we choose $\omega = 1$ and $\beta = \left(1 - \sqrt{0.99\lambda_{\min}^+}\right)^2$ then $\left\|\mathbb{E}\left[x^k - x^*\right]\right\|_{\mathbf{B}}^2 \leq \beta^k C$ and the iteration complexity becomes $\mathcal{O}\left(\sqrt{1/\lambda_{\min}^+}\log(1/\epsilon)\right)$.

(ii) If we choose $\omega = 1/\lambda_{\max}$ and $\beta = \left(1 - \sqrt{\frac{0.99\lambda_{\min}^+}{\lambda_{\max}}}\right)^2$ then $\left\|\mathbb{E}\left[x^k - x^*\right]\right\|_{\mathbf{B}}^2 \leq \beta^k C$ and the iteration complexity becomes $\mathcal{O}\left(\sqrt{\lambda_{\max}/\lambda_{\min}^+}\log(1/\epsilon)\right)$.

Proof. See Section 2.9.4. □

Note that the convergence factor is precisely equal to the value of the momentum parameter β. Let x be any random vector in $\mathbb{R}^n$ with finite mean $\mathbb{E}[x]$, and $x^* \in \mathbb{R}^n$ is any reference vector (for instance, any solution of $\mathbf{A}x = b$). Then we have the identity (see, for instance [73])

$$\mathbb{E}\left[\|x - x^*\|_{\mathbf{B}}^2\right] = \|\mathbb{E}\left[x - x^*\right]\|_{\mathbf{B}}^2 + \mathbb{E}\left[\|x - \mathbb{E}[x]\|_{\mathbf{B}}^2\right]. \tag{2.9}$$

This means that the quantity $\mathbb{E}\left[\|x - x^*\|_{\mathbf{B}}^2\right]$ appearing in the convergence result of Theorem 8 is larger than $\|\mathbb{E}\left[x - x^*\right]\|_{\mathbf{B}}^2$ appearing in the the convergence result of Theorem 11, and hence harder to push to zero. As a corollary, the convergence rate of $\mathbb{E}\left[\|x - x^*\|_{\mathbf{B}}^2\right]$ to zero established in Theorem 8) implies the same rate for the convergence of $\|\mathbb{E}\left[x - x^*\right]\|_{\mathbf{B}}^2$ to zero. However, note that in Theorem 11 we have established an *accelerated* rate for $\|\mathbb{E}\left[x - x^*\right]\|_{\mathbf{B}}^2$. A similar theorem, also obtaining an accelerated rate for $\|\mathbb{E}\left[x - x^*\right]\|_{\mathbf{B}}^2$, was established in [168] for an accelerated variant of SGD in the sense of Nesterov.

2.4 Dual Methods with Momentum

In the previous sections we focused on methods for solving the stochastic optimization problem (1.6) and the best approximation problem (1.22). In this section we focus on the dual of the best approximation problem, and propose a momentum variant of SDSA, which we call mSDSA.

Algorithm 2 Stochastic Dual Subspace Ascent with Momentum (mSDSA)

Input: Distribution $\mathcal{D}$ from which method samples matrices; positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$; stepsize/relaxation parameter $\omega \in \mathbb{R}$ the heavy ball/momentum parameter β.

Initialize: Choose initial points $y^0 = y^1 = 0 \in \mathbb{R}^m$

1: **for** $k = 1, 2, \cdots$ **do**
2: Draw a fresh $\mathbf{S}_k \sim \mathcal{D}$
3: Set $\lambda^k = \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k)\right)$
4: Set $y^{k+1} = y^k + \omega \mathbf{S}_k \lambda^k + \beta(y^k - y^{k-1})$
5: **end for**
6: **Output:** last iterate y^k

2.4.1 Correspondence between primal and dual methods

In our first result we show that the random iterates of the mSGD/mSN/mSPP methods arise as an affine image of mSDSA under the mapping ϕ defined in (1.27).

Theorem 12 (Correspondence Between Primal and Dual Methods)**.** *Let $x^0 = x^1$ and let $\{x^k\}$ be the iterates of mSGD/mSN/mSPP. Let $y^0 = y^1 = 0$, and let $\{y^k\}$ be the iterates of mSDSA. Assume that the methods use the same stepsize $\omega > 0$, momentum parameter $\beta \geq 0$, and the same sequence of random matrices $\mathbf{S}_k$. Then*

$$x^k = \phi(y^k) = x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k$$

for all k. That is, the primal iterates arise as affine images of the dual iterates.

Proof. First note that

$$\nabla f_{\mathbf{S}_k}(\phi(y^k)) \overset{(1.13)}{=} \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}\phi(y^k) - b) \;=\; -\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda^k.$$

We now use this to show that

$$\begin{aligned}
\phi(y^{k+1}) &\overset{(1.31)}{=} x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^{k+1} \\
&= x^0 + \mathbf{B}^{-1}\mathbf{A}^\top \left[y^k + \omega \mathbf{S}_k \lambda^k + \beta(y^k - y^{k-1})\right] \\
&= \underbrace{x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k}_{\phi(y^k)} + \omega \underbrace{\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda^k}_{-\nabla f_{\mathbf{S}_k}(\phi(y^k))} + \beta \mathbf{B}^{-1}\mathbf{A}^\top (y^k - y^{k-1}) \\
&= \phi(y^k) - \omega \nabla f_{\mathbf{S}_k}(\phi(y^k)) + \beta(\mathbf{B}^{-1}\mathbf{A}^\top y^k - \mathbf{B}^{-1}\mathbf{A}^\top y^{k-1}) \\
&\overset{(1.31)}{=} \phi(y^k) - \omega \nabla f_{\mathbf{S}_k}(\phi(y^k)) + \beta(\phi(y^k) - \phi(y^{k-1})).
\end{aligned}$$

So, the sequence of vectors $\{\phi(y^k)\}$ mSDSA satisfies the same recursion of degree as the sequence $\{x^k\}$ defined by mSGD. It remains to check that the first two elements of both recursions coincide. Indeed, since $y^0 = y^1 = 0$ and $x^0 = x^1$, we have $x^0 = \phi(0) = \phi(y^0)$, and $x^1 = x^0 = \phi(0) = \phi(y^1)$. □

2.4.2 Convergence

We are now ready to state a linear convergence convergence result describing the behavior of mSDSA in terms of the dual function values $D(y^k)$.

Theorem 13 (Convergence of dual objective)**.** *Choose $y^0 = y^1 \in \mathbb{R}^n$. Assume exactness. Let $\{y^k\}_{k=0}^\infty$ be the sequence of random iterates produced by mSDSA. Assume $0 \leq \omega \leq 2$ and $\beta \geq 0$ and that the expressions*

$$a_1 := 1 + 3\beta + 2\beta^2 - (\omega(2-\omega) + \omega\beta)\lambda_{\min}^+, \qquad \text{and} \qquad a_2 := \beta + 2\beta^2 + \omega\beta\lambda_{\max}$$

satisfy $a_1 + a_2 < 1$. Let $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ and let y^* be any dual optimal solution. Then

$$\mathbb{E}[D(y^*) - D(y^k)] \leq q^k(1+\delta)\left[D(y^*) - D(y^0)\right] \tag{2.10}$$

where $q = \frac{a_1+\sqrt{a_1^2+4a_2}}{2}$ *and* $\delta = q - a_1$. *Moreover,* $a_1 + a_2 \leq q < 1$.

Proof. This follows by applying Theorem 8 together with Theorem 12 and the identity $\frac{1}{2}\|x^k - x^0\|_{\mathbf{B}}^2 = D(y^*) - D(y^k)$. □

Note that for $\beta = 0$, mSDSA simplifies to SDSA. Also recall that for unit stepsize ($\omega = 1$), SDSA was analyzed in [73]. In the $\omega = 1$ and $\beta = 0$ case, our result specializes to that established in [73]. Following similar arguments to those in [73], the same rate of convergence can be proved for the duality gap $\mathbb{E}[P(x^k) - D(y^k)]$.

2.5 Methods with Stochastic Momentum

To motivate *stochastic momentum*, for simplicity fix $\mathbf{B} = \mathbf{I}$, and assume that $\mathbf{S}_k$ is chosen as the jth random unit coordinate vector of $\mathbb{R}^m$ with probability $p_j > 0$. In this case, SGD (1.17) reduces to the randomized Kaczmarz method for solving the linear system $\mathbf{A}x = b$, first analyzed for $p_j \sim \|\mathbf{A}_{j:}\|^2$ by Strohmer and Vershynin [182].

In this case, mSGD becomes the *randomized Kaczmarz method with momentum* (mRK), and the iteration (2.4) takes the explicit form

$$x^{k+1} = x^k - \omega \frac{\mathbf{A}_{j:}x^k - b_j}{\|\mathbf{A}_{j:}\|^2}\mathbf{A}_{j:}^\top + \beta(x^k - x^{k-1}).$$

Note that the cost of one iteration of this method is $\mathcal{O}(\|\mathbf{A}_{j:}\|_0 + n)$, where the cardinality term $\|\mathbf{A}_{j:}\|_0$ comes from the stochastic gradient part, and n comes from the momentum part. When $\mathbf{A}$ is sparse, the second term will dominate. Similar considerations apply for many other (but clearly not all) distributions $\mathcal{D}$.

In such circumstances, we propose to replace the expensive-to-compute momentum term by a cheap-to-compute stochastic approximation term. In particular, we let i_k be chosen from $[n]$ uniformly at random, and replace $x^k - x^{k-1}$ with $v_{i_k} := e_{i_k}^\top(x^k - x^{k-1})e_{i_k}$, where $e_{i_k} \in \mathbb{R}^n$ is the i_k-th unit basis vector in $\mathbb{R}^n$, and β with $\gamma := n\beta$. Note that v_{i_k} can be computed in $\mathcal{O}(1)$ time. Moreover,

$$\mathbb{E}_{i_k}[\gamma v_{i_k}] = \beta(x^k - x^{k-1}). \tag{2.11}$$

Hence, we replace the momentum term by an unbiased estimator, which allows us to cut the cost to $\mathcal{O}(\|\mathbf{A}_{j:}\|_0)$.

2.5.1 Primal methods with stochastic momentum

We now propose a variant of the SGD/SN/SPP methods employing stochastic momentum (smSGD/smSN/smSPP). Since SGD, SN and SPP are equivalent, we will describe the development from the perspective of SGD. In particular, we propose the following method:

$$x^{k+1} = x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \gamma e_{i_k}^\top(x^k - x^{k-1})e_{i_k}. \tag{2.12}$$

The method is formalized below:

Algorithm 3 smSGD/smSN/smSPP

Input: Distribution $\mathcal{D}$ from which the method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$ the heavy ball/momentum parameter β.
Initialize: Choose initial points $x^1 = x^0 \in \mathbb{R}^n$; set $\mathbf{B} = \mathbf{I} \in \mathbb{R}^{n\times n}$
1: **for** $k = 1, 2, \cdots$ **do**
2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$.
3: Pick $i_k \in [n]$ uniformly at random
4: Set $x^{k+1} = x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \gamma e_{i_k}^\top(x^k - x^{k-1})e_{i_k}$
5: **end for**
6: **Output:** The last iterate x^k

2.5.2 Convergence

In the next result we establish linear convergence of smSGD/smSN/smSPP. For this we will require the matrix $\mathbf{B}$ to be equal to the identity matrix.

Theorem 14. *Choose $x^0 = x^1 \in \mathbb{R}^n$. Assume exactness. Let $\mathbf{B} = \mathbf{I}$. Let $\{x^k\}_{k=0}^{\infty}$ be the sequence of random iterates produced by smSGD/smSN/smSPP. Assume $0 < \omega < 2$ and $\gamma \geq 0$ and that the expressions*

$$a_1 := 1 + 3\frac{\gamma}{n} + 2\frac{\gamma^2}{n} - \left(\omega(2-\omega) + \omega\frac{\gamma}{n}\right)\lambda_{\min}^{+}, \qquad \text{and} \qquad a_2 := \frac{1}{n}(\gamma + 2\gamma^2 + \omega\gamma\lambda_{\max}) \quad (2.13)$$

satisfy $a_1 + a_2 < 1$. Let $x^* = \Pi_{\mathcal{L},\mathbf{I}}(x^0)$. Then

$$\mathbb{E}\left[\|x^k - x^*\|^2\right] \leq q^k(1+\delta)\|x^0 - x^*\|^2 \quad (2.14)$$

and $\mathbb{E}\left[f(x^k)\right] \leq q^k \frac{\lambda_{\max}}{2}(1+\delta)\|x^0 - x^*\|^2$, where $q := \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ and $\delta := q - a_1$. Moreover, $a_1 + a_2 \leq q < 1$.

Proof. See Section 2.9.5. □

It is straightforward to see that if we choose $\omega \in (0,2)$, then the condition $a_1 + a_2 < 1$ is satisfied for all γ belonging to the interval

$$0 \leq \gamma < \frac{1}{8}\left(-4 + \omega\lambda_{\min}^{+} - \omega\lambda_{\max} + \sqrt{(4 - \omega\lambda_{\min}^{+} + \omega\lambda_{\max})^2 + 16n\omega(2-\omega)\lambda_{\min}^{+}}\right).$$

The upper bound is similar to that for mSGD/mSN/mSPP; the only difference is an extra factor of n next to the constant 16.

2.5.3 Momentum versus stochastic momentum

As indicated above, if we wish to compare mSGD with momentum parameter β to smSGD with momentum parameter γ, it makes sense to set $\gamma = \beta n$. Indeed, this is because in view of (2.11), the momentum term in smSGD will then be an unbiased estimator of the deterministic momentum term used in mSGD.

Let $q(\beta)$ be the convergence constant for mSGD with stepsize $\omega = 1$ and an admissible momentum parameter $\beta \geq 0$. Further, let $\bar{a}_1(\gamma), \bar{a}_2(\gamma), \bar{q}(\gamma)$ be the convergence constants for smSGD with stepsize $\omega = 1$ and momentum parameter γ. We have

$$\begin{aligned} \bar{q}(\beta n) \geq \bar{a}_1(\beta n) + \bar{a}_2(\beta n) &\overset{(2.13)}{=} 1 + 4\beta + 4\beta^2 n + \beta(\lambda_{\max} - \lambda_{\min}^{+}) - \lambda_{\min}^{+} \\ &\overset{(2.8)}{=} a_1(\beta) + a_2(\beta) + 4\beta^2(n-1) \\ &\geq a_1(\beta) + a_2(\beta). \end{aligned}$$

Hence, the lower bound on the rate for smSGD is worse than the lower bound for mSGD.

The same conclusion holds for the convergence rates themselves. Indeed, note that since $\bar{a}_1(\beta n) - a_1(\beta) = 2\beta^2(n-1) \geq 0$ and $\bar{a}_2(\beta n) - a_2(\beta) = 2\beta^2(n-1) \geq 0$, we have

$$\bar{q}(\beta n) = \frac{\bar{a}_1(\beta n) + \sqrt{\bar{a}_1^2(\beta n) + 4\bar{a}_2(\beta n)}}{2} \geq \frac{a_1(\beta) + \sqrt{a_1^2(\beta) + 4a_2(\beta)}}{2} = q(\beta),$$

and hence the rate of mSGD is always better than that of smSGD.

However, the expected cost of a single iteration of mSGD may be significantly larger than that of smSGD. Indeed, let g be the expected cost of evaluating a stochastic gradient. Then we need to compare $\mathcal{O}(g+n)$ (mSGD) against $\mathcal{O}(g)$ (smSGD). If $g \ll n$, then one iteration of smSGD is significantly cheaper than one iteration of mSGD. Let us now compare the total complexity to investigate the trade-off between the rate and cost of stochastic gradient evaluation. Ignoring constants, the total cost of the two methods (cost of a single iteration multiplied

by the number of iterations) is:

$$C_{\text{mSGD}}(\beta) := \frac{g+n}{1-q(\beta)} = \frac{g+n}{1-\frac{a_1(\beta)+\sqrt{a_1^2(\beta)+4a_2(\beta)}}{2}}, \tag{2.15}$$

and

$$C_{\text{smSGD}}(\beta n) := \frac{g}{1-\bar{q}(\beta n)} = \frac{g}{1-\frac{\bar{a}_1(\beta n)+\sqrt{\bar{a}_1^2(\beta n)+4\bar{a}_2(\beta n)}}{2}}. \tag{2.16}$$

Since

$$q(0) = \bar{q}(0n), \tag{2.17}$$

and since $q(\beta)$ and $\bar{q}(\beta n)$ are continuous functions of β, then because $g+n > g$, for small enough β we will have $C_{\text{mSGD}}(\beta) > C_{\text{smSGD}}(\beta n)$. In particular, the speedup of smSGD compared to mSGD for $\beta \approx 0$ will be close to

$$\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)} \approx \lim_{\beta' \to +0} \frac{C_{\text{mSGD}}(\beta')}{C_{\text{smSGD}}(\beta' n)} \overset{(2.15)+(2.16)+(2.17)}{=} \frac{g+n}{g} = 1 + \frac{n}{g}.$$

Thus, we have shown the following statement.

Theorem 15. *For small momentum parameters satisfying $\gamma = \beta n$, the total complexity of smSGD is approximately $1 + n/g$ times smaller than the total complexity of mSGD, where n is the number of columns of $\mathbf{A}$, and g is the expected cost of evaluating a stochastic gradient $\nabla f_{\mathbf{S}}(x)$.*

2.6 Special Cases: Randomized Kaczmarz with Momentum and Randomized Coordinate Descent with Momentum

In Table 2.4 we specify several special instances of mSGD by choosing distinct combinations of the parameters $\mathcal{D}$ and $\mathbf{B}$. We use e_i to denote the ith unit coordinate vector in $\mathbb{R}^m$, and $\mathbf{I}_{:C}$ for the column submatrix of the $m \times m$ identity matrix indexed by (a random) set C.

The updates for smSGD can be derived by substituting the momentum term $\beta(x^k - x^{k-1})$ with its stochastic variant $n\beta e_{i_k}^\top (x^k - x^{k-1}) e_{i_k}$. We do not aim to be comprehensive. For more details on the possible combinations of the parameters $\mathbf{S}$ and $\mathbf{B}$ we refer the interested reader to Section 3 of [73].

In the rest of this section we present in detail two special cases: the randomized Kaczmarz method with momentum (mRK) and the randomized coordinate descent method with momentum (mRCD). Further, we compare the convergence rates obtained in Theorem 11 (i.e., bounds on $\|\mathbb{E}\left[x^k\right] - x^*\|_{\mathbf{B}}^2$) with rates that can be inferred from known results for their no-momentum variants.

2.6.1 mRK: randomized Kaczmarz with momentum

We now provide a discussion on mRK (the method in the first row of Table 2.4). Let $\mathbf{B} = \mathbf{I}$ and let pick in each iteration the random matrix $\mathbf{S} = e_i$ with probability $p_i = \|\mathbf{A}_{i:}\|^2 / \|\mathbf{A}\|_F^2$. In this setup the update rule of the mSGD simplifies to

$$x^{k+1} = x^k - \omega \frac{\mathbf{A}_{i:}x^k - b_i}{\|\mathbf{A}_{i:}\|^2} \mathbf{A}_{i:}^\top + \beta(x^k - x^{k-1})$$

and

$$\begin{aligned} \mathbf{W} &\overset{(1.15)}{=} \mathbf{B}^{-1/2}\mathbf{A}^\top \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\mathbf{H}]\mathbf{A}\mathbf{B}^{-1/2} = \mathbb{E}[\mathbf{A}^\top \mathbf{H} \mathbf{A}] \\ &= \sum_{i=1}^m p_i \frac{\mathbf{A}_{i:}^\top \mathbf{A}_{i:}}{\|\mathbf{A}_{i:}\|^2} = \frac{1}{\|\mathbf{A}\|_F^2} \sum_{i=1}^m \mathbf{A}_{i:}^\top \mathbf{A}_{i:} = \frac{\mathbf{A}^\top \mathbf{A}}{\|\mathbf{A}\|_F^2}. \end{aligned} \tag{2.18}$$

Variants of mSGD			
Variant of mSGD	$\mathbf{S}$	$\mathbf{B}$	x^{k+1}
mRK: randomized Kaczmarz with momentum	e_i	$\mathbf{I}$	$x^k - \omega\frac{\mathbf{A}_{i:}x^k - b_i}{\|\mathbf{A}_{i:}\|^2}\mathbf{A}_{i:}^\top + \beta(x^k - x^{k-1})$
mRCD = mSDSA: randomized coordinate desc. with momentum	e_i	$\mathbf{A} \succ 0$	$x^k - \omega\frac{(\mathbf{A}_{i:})^\top x^k - b_i}{\mathbf{A}_{ii}}e_i + \beta(x^k - x^{k-1})$
mRBK: randomized block Kaczmarz with momentum	$\mathbf{I}_{:C}$	$\mathbf{I}$	$x^k - \omega\mathbf{A}_{C:}^\top(\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger(\mathbf{A}_{C:}x^k - b_C) + \beta(x^k - x^{k-1})$
mRCN = mSDSA: randomized coordinate Newton descent with momentum	$\mathbf{I}_{:C}$	$\mathbf{A} \succ 0$	$x^k - \omega\mathbf{I}_{:C}(\mathbf{I}_{:C}^\top\mathbf{A}\mathbf{I}_{:C})^\dagger\mathbf{I}_{:C}^\top(\mathbf{A}x^k - b) + \beta(x^k - x^{k-1})$
mRGK: randomized Gaussian Kaczmarz	$N(0,\mathbf{I})$	$\mathbf{I}$	$x^k - \omega\frac{\mathbf{S}^\top(\mathbf{A}x^k - b)}{\|\mathbf{A}^\top\mathbf{S}\|^2}\mathbf{A}^\top\mathbf{S} + \beta(x^k - x^{k-1})$
mRCD: randomized coord. descent (least squares)	$\mathbf{A}_{:i}$	$\mathbf{A}^\top\mathbf{A}$	$x^k - \omega\frac{(\mathbf{A}_{:i})^\top(\mathbf{A}x^k - b)}{\|\mathbf{A}_{:i}\|^2}e_i + \beta(x^k - x^{k-1})$

Table 2.4: Selected special cases of mSGD. In the special case of $\mathbf{B} = \mathbf{A}$, mSDSA is directly equivalent to mSGD (this is due to the primal-dual relationship (1.31); see also Theorem 12). Randomized coordinate Newton (RCN) method was first proposed in [162]; mRCN is its momentum variant. Randomized Gaussian Kaczmarz (RGK) method was first proposed in [73]; mRGK is its momentum variant.

The objective function takes the following form:

$$f(x) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[f_{\mathbf{S}}(x)] = \sum_{i=1}^{m} p_i f_{\mathbf{S}_i}(x) = \frac{\|\mathbf{A}x - b\|^2}{2\|\mathbf{A}\|_F^2}. \tag{2.19}$$

For $\beta = 0$, this method reduces to the *randomized Kaczmarz method* with relaxation, first analyzed in [168]. If we also have $\omega = 1$, this is equivalent with the *randomized Kaczmarz method* of Strohmer and Vershynin [182]. RK without momentum ($\beta = 0$) and without relaxation ($\omega = 1$) converges with iteration complexity [182, 73, 74] of

$$\mathcal{O}\left(\frac{1}{\lambda_{\min}^+(\mathbf{W})}\log(1/\epsilon)\right) = \mathcal{O}\left(\frac{\|\mathbf{A}\|_F^2}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}\log(1/\epsilon)\right). \tag{2.20}$$

In contrast, based on Theorem 11 we have

- For $\omega = 1$ and $\beta = \left(1 - \sqrt{0.99\lambda_{\min}^+}\right)^2 = \left(1 - \sqrt{\frac{0.99}{\|\mathbf{A}\|_F^2}\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}\right)^2$, the iteration complexity of the mRK is:

$$\mathcal{O}\left(\sqrt{\frac{\|\mathbf{A}\|_F^2}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}}\log(1/\epsilon)\right).$$

- For $\omega = \|\mathbf{A}\|_F^2/\lambda_{\max}(\mathbf{A}^\top\mathbf{A})$ and $\beta = \left(1 - \sqrt{\frac{0.99\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})}}\right)^2$ the iteration complexity becomes:

$$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}}\log(1/\epsilon)\right).$$

This is quadratic improvement on the previous best result (2.20).

Related Work. The Kaczmarz method for solving consistent linear systems was originally introduced by Kaczmarz in 1937 [91]. This classical method selects the rows to project onto in a cyclic manner. In practice, many different selection rules can be adopted. For non-random selection rules (cyclic, greedy, etc) we refer the interested reader to [158, 17, 144, 159, 27]. In this work we are interested in *randomized* variants of the Kaczmarz method, first analyzed by Strohmer and Vershynin [182]. In [182] it was shown that RK converges with a linear convergence rate to the unique solution of a full-rank consistent linear system. This result sparked renewed interest in design of randomized methods for solving linear systems [132, 134, 49, 120, 216, 135, 175, 114]. All existing results on accelerated variants of RK use the Nesterov's approach of acceleration [102, 107, 192, 168]. To the best of our knowledge, no convergence analysis of mRK exists in the literature (Polyak's momentum). Our work fills this gap.

2.6.2 mRCD: randomized coordinate descent with momentum

We now provide a discussion on the mRCD method (the method in the second row of Table 2.4). If the matrix $\mathbf{A}$ is positive definite, then we can choose $\mathbf{B} = \mathbf{A}$ and $\mathbf{S} = e_i$ with probability $p_i = \frac{\mathbf{A}_{ii}}{\text{Trace}(\mathbf{A})}$. It is easy to see that $\mathbf{W} = \frac{\mathbf{A}}{\text{Trace}(\mathbf{A})}$. In this case, $\mathbf{W}$ is positive definite and as a result, $\lambda_{\min}^+(\mathbf{W}) = \lambda_{\min}(\mathbf{W})$. Moreover, we have

$$f(x) = \mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[f_{\mathbf{S}}(x)] = \sum_{i=1}^m p_i f_{\mathbf{S}_i}(x) = \frac{\|\mathbf{A}x - b\|^2}{2\text{Trace}(\mathbf{A})}. \tag{2.21}$$

For $\beta = 0$ and $\omega = 1$ the method is equivalent with *randomized coordinate descent* of Leventhal and Lewis [104], which was shown to converge with iteration complexity

$$\text{Previous best result:} \qquad \mathcal{O}\left(\frac{\text{Trace}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}\log(1/\epsilon)\right). \tag{2.22}$$

In contrast, following Theorem 11, we can obtain the following iteration complexity results for mRCD:

- For $\omega = 1$ and $\beta = \left(1 - \sqrt{\frac{0.99}{\text{Trace}(\mathbf{A})}\lambda_{\min}(\mathbf{A})}\right)^2$, the iteration complexity is

$$\mathcal{O}\left(\sqrt{\frac{\text{Trace}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}}\log(1/\epsilon)\right).$$

- For $\omega = \text{Trace}(\mathbf{A})/\lambda_{\max}(\mathbf{A})$ and $\beta = \left(1 - \sqrt{\frac{0.99\lambda_{\min}(\mathbf{A})}{\lambda_{\max}(\mathbf{A})}}\right)^2$ the iteration complexity becomes

$$\mathcal{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})}}\log(1/\epsilon)\right).$$

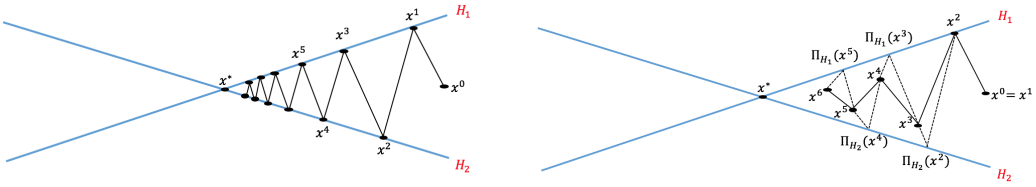
This is quadratic improvement on the previous best result (2.22).

Related Work. It is known that if $\mathbf{A}$ is positive definite, the popular *randomized Gauss-Seidel* method can be interpreted as randomized coordinate descent (RCD). RCD methods were first analyzed by Lewis and Leventhal in the context of linear systems and least-squares problems

[104], and later extended by several authors to more general settings, including smooth convex optimization [139], composite convex optimization [166], and parallel/subspace descent variants [167]. These results were later further extended to handle arbitrary sampling distributions [160, 161, 163, 22] . Accelerated variants of RCD were studied in [102, 51, 2]. For other non-randomized coordinate descent variants and their convergence analysis, we refer the reader to [199, 143, 27]. To the best of our knowledge, mRCD and smRCD have never been analyzed before in any setting.

2.6.3 Visualizing the acceleration mechanism

We devote this section to the graphical illustration of the acceleration mechanism behind momentum. Our goal is to shed more light on how the proposed algorithm works in practice. For simplicity, we illustrate this by comparing RK and mRK.



(a) Randomized Kaczmarz Method [182]

(b) Randomized Kaczmarz Method with Momentum

Figure 2.1: Graphical interpretation of the *randomized Kaczmarz method* and the *randomized Kaczmarz method with momentum* in a simple example with only two hyperplanes $H_i = \{x \ : \mathbf{A}_{i:}x = b_i\}$ where $i = 1, 2$ and a unique solution x^*.

In Figure 2.1 we present in a simple $\mathbb{R}^2$ illustration of the difference between the workings of RK and mRK. Our goal is to show graphically how the addition of momentum leads to acceleration. Given iterate x^k, one can think of the update rule of the mRK (2.4) in two steps:

1. *The Projection:* The projection step corresponds to the first part $x^k - \omega\nabla f_{\mathbf{S}_k}(x^k)$ of the mRK update (2.4) and it means that the current iterate x^k is projected onto a randomly chosen hyperplane H_i[1]. The value of the stepsize $\omega \in (0, 2)$ defines whether the projection is exact or not. When $\omega = 1$ (no relaxation) the projection is exact, that is the point $\Pi_{H_i}(x^k)$ belongs in the hyperplane H_i. In Figure 2.1 all projections are exact.

2. *Addition of the momentum term*: The momentum term (right part of the update rule) $\beta(x^k - x^{k-1})$ forces the next iterate x^{k+1} to be closer to the solution x^* than the corresponding point $\Pi_{H_i}(x^k)$. Note also that the vector $x^{k+1} - \Pi_{H_i}(x^k)$ is always parallel to $x^k - x^{k-1}$ for all $k \geq 0$.

Remark 1. *In the example of Figure 2.1, the performance of mRK is similar to the performance of RK until iterate x^3. After this point, the momentum parameter becomes more effective and the mRK method accelerates. This behavior appears also in our experiments in the next section where we work with matrices with many rows. There we can notice that the momentum parameter seems to become more effective after the first $m + 1$ iterations.*

2.7 Numerical Experiments

In this section we study the computational behavior of the two proposed algorithms, mSGD and smSGD. In particular, we focus mostly on the evaluation of the performance of mSGD. To highlight the usefulness of smSGD, an empirical verification of Theorem 15 is presented in subsection 2.7.2. As we have already mentioned, both mSGD and smSGD can be interpreted as sketch-and-project methods (with relaxation), and as a result a comprehensive array of

[1] In the plots of Figure 2.1, the hyperplane of each update is chosen in an alternating fashion for illustration purposes

well-known algorithms can be recovered as special cases by varying the main parameters of the methods (check Section 2.6). In our experiments we focus on the popular special cases of randomized Kaczmarz method (RK) and the randomized coordinate descent method (RCD) without relaxation ($\omega = 1$), and show the practical benefits of adding the momentum term[2]. The choice of the stepsize $\omega = 1$ is not arbitrary. Recently, in [168] both relaxed RK and relaxed RCD were analyzed, and it was proved that the quantity $\mathbb{E}\left[\|x^k - x^*\|_{\mathbf{B}}^2\right]$ converges linearly to zero for $\omega \in (0, 2)$, and that the best convergence rate is obtained precisely for $\omega = 1$. Thus the comparison is with the best-in-theory no-momentum variants.

Note that, convergence analysis of the error $\mathbb{E}\left[\|x^k - x^*\|_{\mathbf{B}}^2\right]$ and of the expected function values $\mathbb{E}\left[f(x^k)\right]$ in Theorem 8 shows that mSGD enjoys global non-asymptotic linear convergence rate but not faster than the no-momentum method. The accelerated linear convergence rate has been obtained only in the weak sense (Theorem 11). Nevertheless, in practice as indicated from our experiments, mSGD is faster than its no momentum variant. Note also that in all of the presented experiments the momentum parameters β of the methods are chosen to be positive constants that do not depend on parameters that are not known to the users such as $\lambda_{\min}^+$ and $\lambda_{\max}$.

In comparing the methods with their momentum variants we use both the relative error measure $\|x^k - x^*\|_{\mathbf{B}}^2 / \|x^0 - x^*\|_{\mathbf{B}}^2$ and the function values $f(x^k)$[3]. In all implementations, except for the experiments on average consensus (Section 2.7.3), the starting point is chosen to be $x^0 = 0$. In the case of average consensus the starting point must be the vector with the initial private values of the nodes of the network. All the code for the experiments is written in the Julia programming language. For the horizontal axis we use either the number of iterations or the wall-clock time measured using the tic-toc Julia function.

This section is divided in three main experiments. In the first one we evaluate the performance of the mSGD method in the special cases of mRK and mRCD for solving both synthetic consistent Gaussian systems and consistent linear systems with real matrices. In the second experiment we computationally verify Theorem 15 (comparison between the mSGD and smSGD methods). In the last experiment building upon the recent results of [113] we show how the addition of the momentum accelerates the pairwise randomized gossip (PRG) algorithm for solving the average consensus problem.

Assumptions	No-momentum, $\beta = 0$	Momentum, $\beta \geq 0$	Stochastic Momentum, $\beta \geq 0$
$\mathbf{A}$ general, $\mathbf{B} = \mathbf{I}$	RK	mRK	smRK
$\mathbf{A} \succ 0$, $\mathbf{B} = \mathbf{A}$	RCD	mRCD	smRCD
$\mathbf{A}$ incidence matrix, $\mathbf{B} = \mathbf{I}$	PRG	mPRG	smPRG

Table 2.5: Abbreviations of the algorithms (special cases of general framework) that we use in the numerical evaluation section. In all methods the random matrices are chosen to be unit coordinate vectors in $\mathbb{R}^m$ ($\mathbf{S} = e_i$). With PRG we denote the Pairwise Randomized Gossip algorithm for solving the average consensus problem first proposed in [16]. Following similar notation with the rest of the chapter with mPRG and smPRG we indicate its momentum and stochastic momentum variants respectively.

2.7.1 Evaluation of mSGD

In this subsection we study the computational behavior of mRK and mRCD when they compared with their no momentum variants for both synthetic and real data.

[2]The experiments were repeated with various values of the main parameters and initializations, and similar results were obtained in all cases.

[3]Remember that in our setting we have $f(x^*) = 0$ for the optimal solution x^* of the best approximation problem; thus $f(x) - f(x^*) = f(x)$. The function values $f(x^k)$ refer to function (2.19) in the case of RK and to function (2.21) for the RCD. For block variants the objective function of problem (1.6) has also closed form expression but it can be very difficult to compute. In these cases one can instead evaluate the quantity $\|\mathbf{A}x - b\|_{\mathbf{B}}^2$.

Synthetic Data

The synthetic data for this comparison is generated as follows[4].

For mRK: All elements of matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ and vector $z \in \mathbb{R}^n$ are chosen to be i.i.d $\mathcal{N}(0,1)$. Then the right hand side of the linear system is set to $b = \mathbf{A}z$. With this way the consistency of the linear system with matrix $\mathbf{A}$ and right hand side b is ensured.

For mRCD: A Gaussian matrix $\mathbf{P} \in \mathbb{R}^{m\times n}$ is generated and then matrix $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{n\times n}$ is used in the linear system. The vector $z \in \mathbb{R}^n$ is chosen to be i.i.d $\mathcal{N}(0,1)$ and again to ensure consistency of the linear system, the right hand side is set to $b = \mathbf{A}z$.

In particular for the evaluation of mRK we generate Gaussian matrices with $m = 300$ rows and several columns while for the case of mRCD the matrix $\mathbf{P}$ is chosen to be Gaussian with $m = 500$ rows and several columns[5]. Linear systems of these forms were extensively studied [182, 61] and it was shown that the quantity $1/\lambda^+_{\min}$(condition number) can be easily controlled.

For each linear system we run mRK (Figures 2.2 and 2.3) and mRCD (Figures 2.4 and 2.5) for several values of momentum parameters β and fixed stepsize $\omega = 1$ and we plot the performance of the methods (average after 10 trials) for both the relative error measure and the function values. Note that for $\beta = 0$ the methods are equivalent with their no-momentum variants RK and RCD respectively.

From Figures 2.2, 2.3, 2.4 and 2.5 it is clear that the addition of momentum term leads to an improvement in the performance of both, RK and RCD. More specifically, from the four figures we observe the following:

- For the well conditioned linear systems ($1/\lambda^+_{\min}$ small) it is known that even the no-momentum variant converges rapidly to the optimal solution. In these cases the benefits of the addition of momentum are not obvious. The momentum term is beneficial for the case where the no-momentum variant ($\beta = 0$) converges slowly, that is when $1/\lambda^+_{\min}$ is large (ill-conditioned linear systems).

- For the case of fixed stepsize $\omega = 1$, the problems with small condition number require smaller momentum parameter β to have faster convergence. Note the first two rows of Figures 2.2 and 2.4, where $\beta = 0.3$ or $\beta = 0.4$, are good options.

- For large values of $1/\lambda^+_{\min}$, it seems that the choice of $\beta = 0.5$ is the best. As an example for matrix $\mathbf{A} \in \mathbb{R}^{300\times 280}$ in Figure 2.2, (where $1/\lambda^+_{\min} = 208,730$), note that to reach relative error 10^{-10}, RK needs around 2 million iterations, while mRK with momentum parameter $\beta = 0.5$ requires only half that many iterations. The acceleration is obvious also in terms of time where in 12 seconds the mRK with momentum parameter $\beta = 0.5$ achieves relative error of the order 10^{-9} and RK requires more than 25 seconds to obtain the same accuracy.

- We observe that both mRK and mRCD, with appropriately chosen momentum parameters $0 < \beta \leq 0.5$, always converge faster than their no-momentum variants, RK and RCD, respectively. This is a smaller momentum parameter than $\beta \approx 0.9$ which is being used extensively with mSGD for training deep neural networks [211, 196, 185].

- In [203] a stochastic power iteration with momentum is proposed for principal component analysis (PCA). There it was demonstrated empirically that a naive application of momentum to the stochastic power iteration does not result in a faster method. To achieve faster convergence, the authors proposed mini-batch and variance-reduction techniques on top of the addition of momentum. In our setting, mere addition of the momentum term to SGD (same is true for special cases such as RK and RCD) leads to empirically faster methods.

[4]Note that in the first experiment we use Gaussian matrices which by construction are full rank matrices with probability 1 and as a result the consistent linear systems have unique solution. Thus, for any starting point x^0, the vector z that is used to create the linear system is the solution mSGD converges to. This is not true for general consistent linear systems, with no full-rank matrix. In this case, the solution $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ that mSGD converges to is not necessarily equal to z. For this reason, in the evaluation of the relative error measure $\|x^k - x^*\|^2_{\mathbf{B}}/\|x^0 - x^*\|^2_{\mathbf{B}}$, one should be careful and use the value $x^* = x^0 + \mathbf{A}^\dagger(b - \mathbf{A}x^0) \overset{x^0=0}{=} \mathbf{A}^\dagger b$.

[5]RCD converge to the optimal solution only in the case of positive definite matrices. For this reason $\mathbf{A} = \mathbf{P}^\top\mathbf{P} \in \mathbb{R}^{n\times n}$ is used which with probability 1 is a full rank matrix

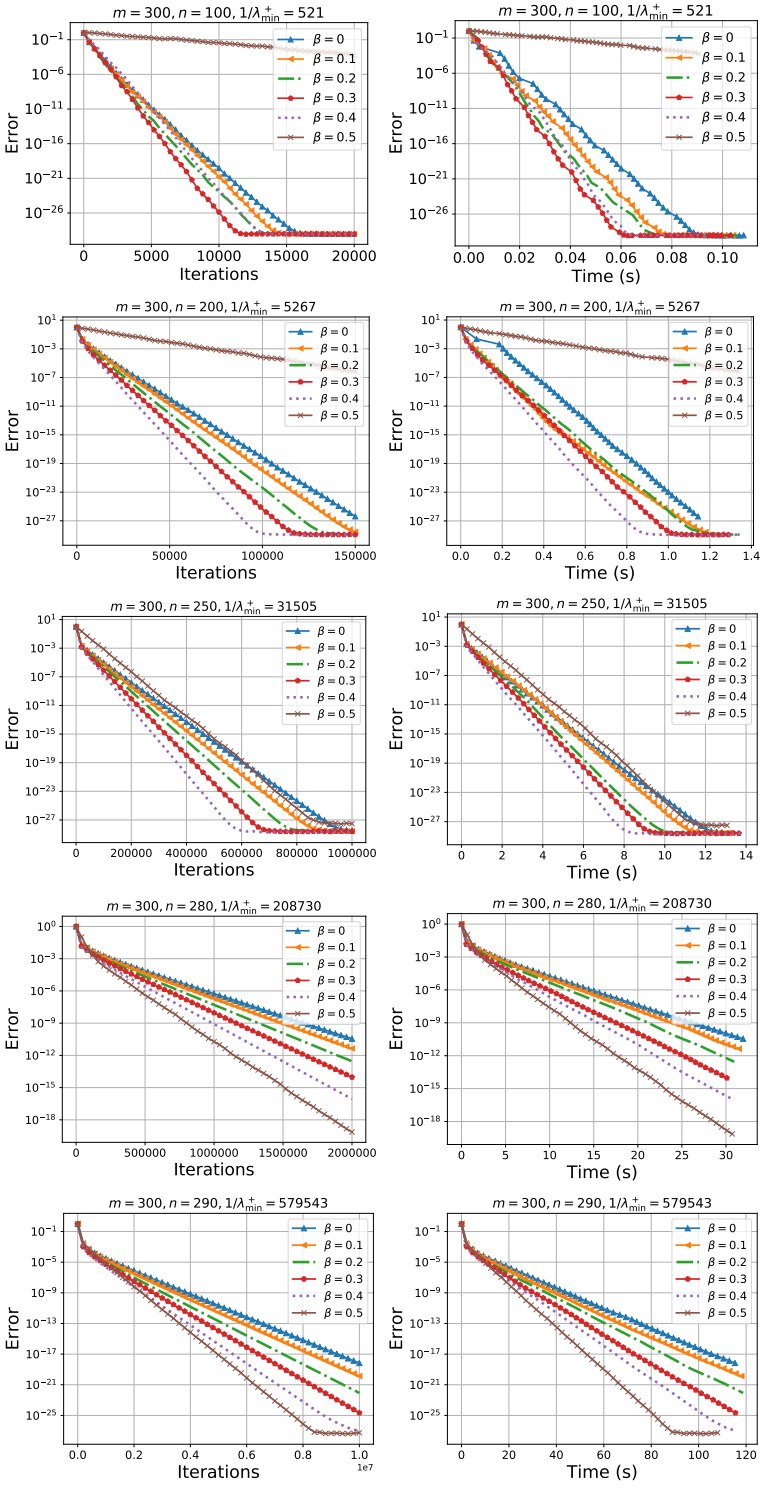


Figure 2.2: Performance of mRK (the method in the first row of Table 2.4) for fixed stepsize $\omega = 1$ and several momentum parameters β for consistent linear systems with Gaussian matrix $\mathbf{A}$ with $m = 300$ rows and $n = 100, 200, 250, 280, 290$ columns. The graphs in the first (second) column plot iterations (time) against residual error. All plots are averaged over 10 trials. The title of each plot indicates the dimensions of the matrix $\mathbf{A}$ and the value of $1/\lambda^+_{\min}$. The "Error" on the vertical axis represents the relative error $\|x^k - x^*\|^2_{\mathbf{B}}/\|x^0 - x^*\|^2_{\mathbf{B}} \overset{\mathbf{B}=\mathbf{I},x^0=0}{=} \|x^k - x^*\|^2/\|x^*\|^2_{\mathbf{B}}$.

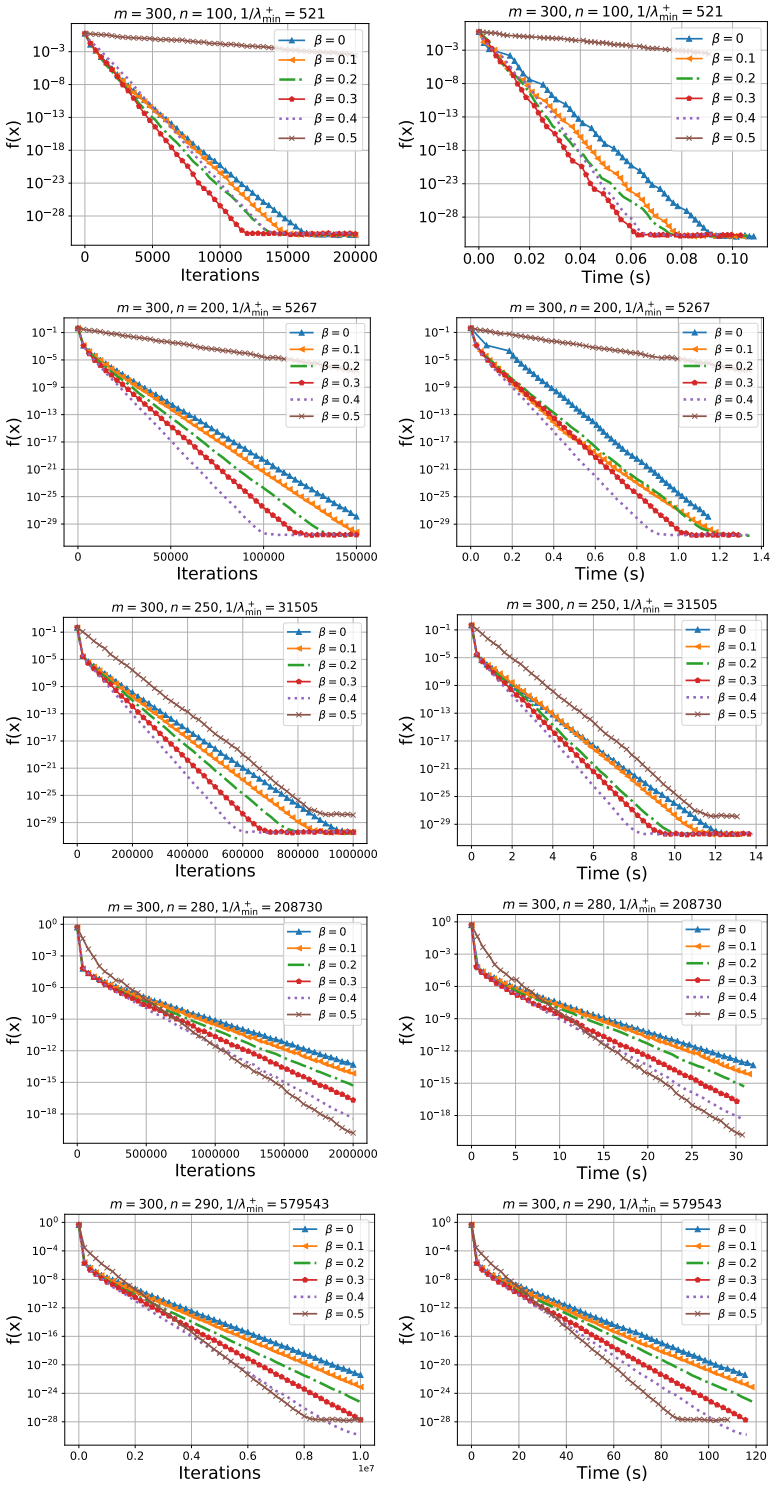


Figure 2.3: Performance of mRK (the method in the first row of Table 2.4) for fixed stepsize $\omega = 1$ and several momentum parameters β for consistent linear systems with Gaussian matrix $\mathbf{A}$ with $m = 300$ rows and $n = 100, 200, 250, 280, 290$ columns. The graphs in the first (second) column plot iterations (time) against function values. All plots are averaged over 10 trials. The title of each plot indicates the dimensions of the matrix $\mathbf{A}$ and the value of $1/\lambda^+_{\min}$. The function values $f(x^k)$ refer to function (2.19).

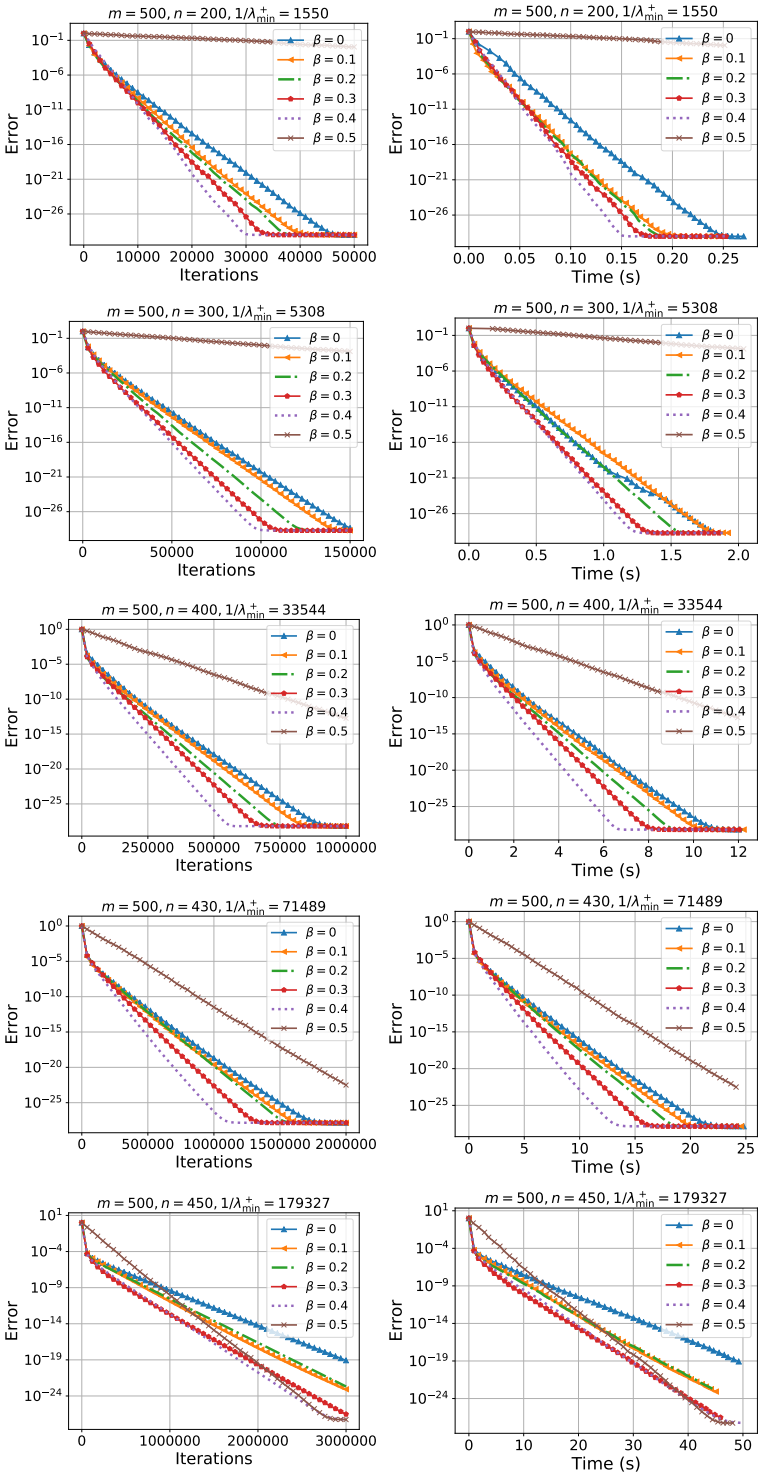


Figure 2.4: Performance of mRCD (the method in the second row of Table 2.4). for fixed stepsize $\omega = 1$ and several momentum parameters β for consistent linear systems with positive definite matrices $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ where $\mathbf{P} \in \mathbb{R}^{m \times n}$ is Gaussian matrix with $m = 500$ rows and $n = 200, 300, 400, 430, 450$. The graphs in the first (second) column plot iterations (time) against residual error. All plots are averaged over 10 trials. The title of each plot indicates the dimensions of the matrix $\mathbf{P}$ and the value of $1/\lambda^+_{\min}$. The "Error" on the vertical axis represents the relative error $\|x^k - x^*\|^2_{\mathbf{B}} / \|x^0 - x^*\|^2_{\mathbf{B}} \overset{\mathbf{B}=\mathbf{A}, x^0=0}{=} \|x^k - x^*\|^2_{\mathbf{A}} / \|x^*\|^2_{\mathbf{A}}$.

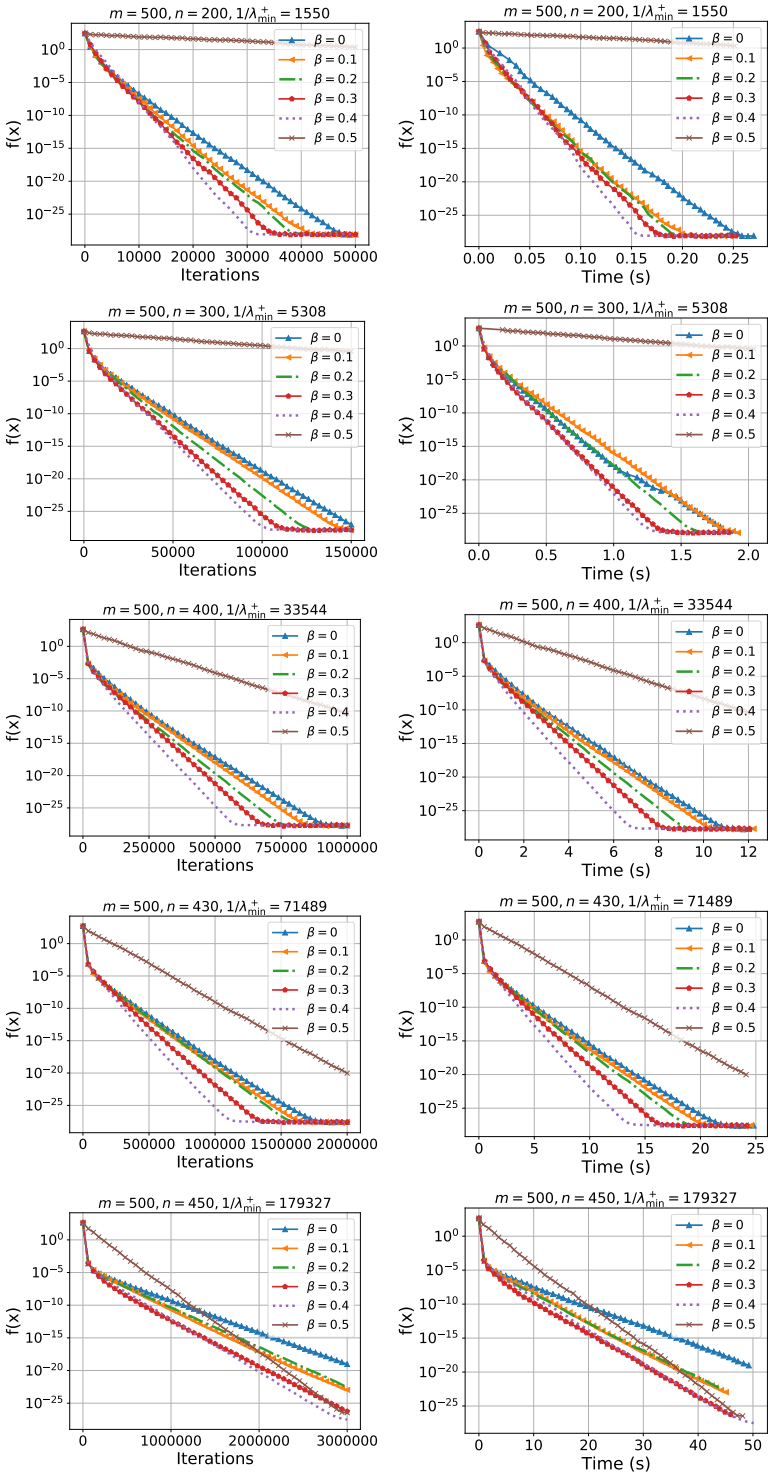


Figure 2.5: Performance of mRCD (the method in the second row of Table 2.4). for fixed stepsize $\omega = 1$ and several momentum parameters β for consistent linear systems with positive definite matrices $\mathbf{A} = \mathbf{P}^\top\mathbf{P}$ where $\mathbf{P} \in \mathbb{R}^{m\times n}$ is Gaussian matrix with $m = 500$ rows and $n = 200, 300, 400, 430, 450$. The graphs in the first (second) column plot iterations (time) against function values. All plots are averaged over 10 trials. The title of each plot indicates the dimensions of the matrix $\mathbf{P}$ and the value of $1/\lambda^+_{\min}$. The function values $f(x^k)$ refer to function (2.21).

Real Data

In the following experiments we test the performance of mRK using real matrices (datasets) from the library of support vector machine problems LIBSVM [23]. Each dataset consists of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m features and n characteristics) and a vector of labels $b \in \mathbb{R}^m$. In our experiments we choose to use only the matrices of the datasets and ignore the label vector. As before, to ensure consistency of the linear system, we choose a Gaussian vector $z \in \mathbb{R}^n$ and the right hand side of the linear system is set to $b = \mathbf{A}z$. Similarly as in the case of synthetic data, mRK is tested for several values of momentum parameters β and fixed stepsize $\omega = 1$.

In Figure 2.6 the performance of all methods for both relative error measure $\|x^k - x^*\|^2/\|x^*\|^2_{\mathbf{B}}$ and function values $f(x^k)$ is presented. Note again that $\beta = 0$ represents the baseline RK method. The addition of momentum parameter is again often beneficial and leads to faster convergence. As an example, inspect the plots for the *mushrooms* dataset in Figure 2.6, where mRK with $\beta = 0.5$ is much faster than the simple RK method in all presented plots, both in terms of iterations and time. In particular, the addition of a momentum parameter leads to visible speedup for the datasets *mushrooms*, *splice*, *a9a* and *ionosphere*. For these datasets the acceleration is obvious in all plots both in terms of relative error and function values. For the datasets *australian*, *gisette* and *madelon* the speedup is less obvious in the plots of the relative error, while for the plots of function values it is not present at all.

2.7.2 Comparison of momentum & stochastic momentum

In Theorem 15, the total complexities (number of operations needed to achieve a given accuracy) of mSGD and smSGD have been compared and it has been shown that for small momentum parameter β,

$$C_\beta = \frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)} \approx 1 + \frac{n}{g},$$

where C_{mSGD} and C_{smSGD} represent the total costs of the two methods. The goal of this experiment is to show that this relationship holds also in practice.

For this experiment we assume that the non-zeros of matrix $\mathbf{A}$ are not concentrated in certain rows but instead that each row has the same number of non-zero coordinates. We denote by g the number the non-zero elements per row. Having this assumption it can be shown that for the RK method the cost of one projection is equal to $4g$ operations while the cost per iteration of the mRK and of the smRK are $4g + 3n$ and $4g + 1$ respectively. For more details about the cost per iteration of the general mSGD and smSGD check Table 2.6.

As a first step a Gaussian matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ is generated. Then using this matrix several consistent linear systems are obtained as follows. Several values for $g \in [1, n]$ are chosen and for each one of these a matrix $\mathbf{A}_g \in \mathbb{R}^{m\times n}$ with the same elements as $\mathbf{A}$ but with $n - g$ zero coordinates per row is produced. For every matrix $\mathbf{A}_g$, a Gaussian vector $z_g \in \mathbb{R}^n$ is drawn and to ensure consistency of the linear system, the right hand side is set to $b_g = \mathbf{A}_g z$.

We run both mSGD and smSGD with small momentum parameter $\beta = 0.0001$ for solving the linear systems $\mathbf{A}_g x = b_g$ for all selected values of $g \in [1, n]$. The starting point for each run is taken to be $x^0 = 0 \in \mathbb{R}^n$. The methods run until $\epsilon = \|x^k - x^*\| < 0.001$, where $x^* = \Pi_{\mathcal{L}_g}(x^0)$[6] and $\mathcal{L}_g$ is the solution set of the linear system $\mathbf{A}_g x = b_g$. In each run the number of operations needed to achieve the accuracy ϵ have been counted. For each linear system the average after 10 trials of the value $\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)}$ is computed.

[6] To pre-compute the solution x^* for each linear system $\mathbf{A}_g x = b_g$ we use the closed form expression of the projection (1.14).

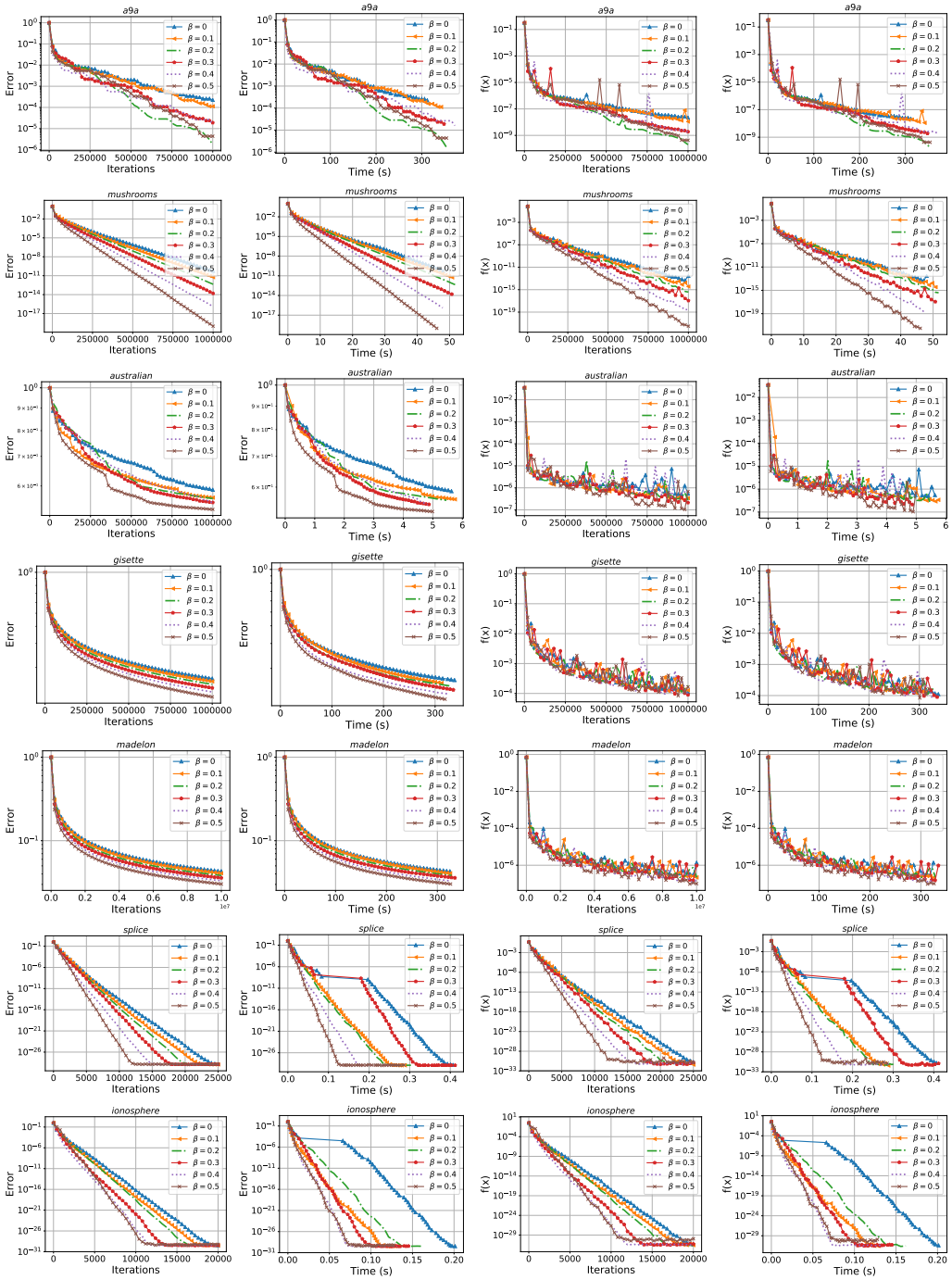
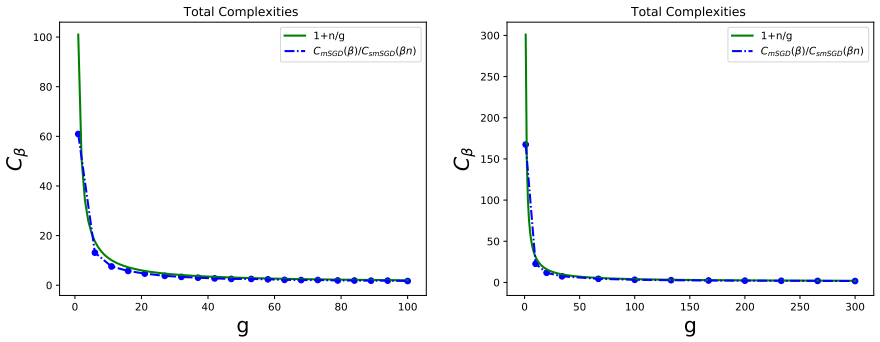


Figure 2.6: The performance of mRK for several momentum parameters β on real data from LIBSVM [23]. a9a: $(m,n) = (32561, 123)$, mushrooms: $(m,n) = (8124, 112)$, australian: $(m,n) = (690, 14)$, gisette: $(m,n) = (6000, 5000)$, madelon: $(m,n) = (2000, 500)$, splice: $(m,n) = (1000, 60)$, ionosphere: $(m,n) = (351, 34)$. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. The "Error" on the vertical axis represents the relative error $\|x^k - x^*\|_{\mathbf{B}}^2 / \|x^0 - x^*\|_{\mathbf{B}}^2 \overset{\mathbf{B}=\mathbf{I}, x^0=0}{=} \|x^k - x^*\|^2 / \|x^*\|_{\mathbf{B}}^2$ and the function values $f(x^k)$ refer to function (2.19).



(a) $\mathbf{A} \in \mathbb{R}^{200\times 100}$ (b) $\mathbf{A} \in \mathbb{R}^{1000\times 300}$

Figure 2.7: Comparison of the total complexities of mRK and smRK. The green continuous line denotes the theoretical relationship $1+\frac{n}{g}$ that we predict in Theorem 15. The blue dotted line shows the ratio of the total complexities $\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)}$ for several linear systems $\mathbf{A}_g x = b_g$ where $g \in [1, n]$. The momentum parameter $\beta = 0.0001$ is used for both methods.

Algorithm	Cost per iteration	Cost per Iteration (RK, mRK, smRK)
Basic Method ($\beta = 0$)	$\mathcal{O}(g)$	$4g$
mSGD	$\mathcal{O}(g) + \mathcal{O}(n) = \mathcal{O}(n)$	$4g + 3n$
smSGD	$\mathcal{O}(g) + \mathcal{O}(1) = \mathcal{O}(g)$	$4g + 1$

Table 2.6: Cost per iteration of the basic, mSGD and smSGD in the general setting and in the special cases of RK,mRK and smRK.

In Figure 2.7 the actual ratio $\frac{C_{\text{mSGD}}(\beta)}{C_{\text{smSGD}}(\beta n)}$ and the theoretical approximation $1+\frac{n}{g}$ are plot and it is shown that they have similar behavior. Thus the theoretical prediction of Theorem 15 is numerically confirmed. In particular in the implementations we use the Gaussian matrices $\mathbf{A} \in \mathbb{R}^{200\times 100}$ and $\mathbf{A} \in \mathbb{R}^{1000\times 300}$.

2.7.3 Faster method for average consensus

Background

Average consensus (AC) is a fundamental problem in distributed computing and multi-agent systems [42, 16]. Consider a connected undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E}$, ($|\mathcal{E}| = m$), where each node $i \in \mathcal{V}$ owns a private value $c_i \in \mathbb{R}$. The goal of the AC problem is for each node of the network to compute the average of these private values, $\bar{c} := \frac{1}{n}\sum_i c_i$, via a protocol which allows communication between neighbours only. The problem comes up in many real world applications such as coordination of autonomous agents, estimation, rumour spreading in social networks, PageRank and distributed data fusion on ad-hoc networks and decentralized optimization.

It was shown recently that several randomized methods for solving linear systems can be interpreted as randomized gossip algorithms for solving the AC problem when applied to a special system encoding the underlying network [74, 113]. As we have already explained both basic method [168] and basic method with momentum (this chapter) find the solution of the linear system that is closer to the starting point of the algorithms. That is, both methods converge linearly to $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$; the projection of the initial iterate onto the solution set of the linear system. As a result (check the Introduction), they can be interpreted as methods for solving the best approximation problem (1.22). In the special case that

1. the linear system in the constraints of (1.22) is the homogeneous linear system ($\mathbf{A}x = 0$) with matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ being the incidence matrix of the undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and

2. the starting point of the method are the initial values of the nodes $x^0 = c$,

it is straightforward to see that the solution of the best approximation problem is a vector with all components equal to the consensus value $\bar{c} := \frac{1}{n}\sum_i c_i$. Under this setting, the famous randomized pairwise gossip algorithm (randomly pick an edge $e \in E$ and replace the private values of its two nodes to their average) that was first proposed and analyzed in [16], is equivalent to the RK method without relaxation ($\omega = 1$) [74, 113].

Remark 2. *In the gossip framework, the condition number of the linear system when RK is used has a simple structure and it depends on the characteristics of the network under study. More specifically, it depends on the number of the edges m and on the Laplacian matrix of the network*[7]*:*

$$\frac{1}{\lambda_{\min}^{+}(\mathbf{W})} \overset{(2.18)}{=} \frac{1}{\lambda_{\min}^{+}(\mathbf{A}^\top\mathbf{A}/\|\mathbf{A}\|_F^2)} \overset{\|\mathbf{A}\|_F^2=2m}{=} \frac{2m}{\lambda_{\min}^{+}(\mathbf{A}^\top\mathbf{A})} = \frac{2m}{\lambda_{\min}^{+}(\mathbf{L})}, \tag{2.23}$$

where $\mathbf{L} = \mathbf{A}^\top\mathbf{A}$ *is the Laplacian matrix of the network and the quantity* $\lambda_{\min}^{+}(\mathbf{L})$ *is the very well studied algebraic connectivity of the graph [31].*

Remark 3. *The convergence analysis in this chapter holds for any consistent linear system* $\mathbf{A}x = b$ *without any assumption on the rank of the matrix* $\mathbf{A}$*. The lack of any assumption on the form of matrix* $\mathbf{A}$ *allows us to solve the homogeneous linear system* $\mathbf{A}x = 0$ *where* $\mathbf{A}$ *is the incidence matrix of the network which by construction is rank deficient. More specifically, it can be shown that* $\text{rank}(\mathbf{A}) = n - 1$ *[113]. Note that many existing methods for solving linear systems make the assumption that the matrix* $\mathbf{A}$ *of the linear systems is full rank [182, 132, 134] and as a result can not be used to solve the AC problem.*

Numerical Setup

Our goal in this experiment is to show that the addition of the momentum term to the randomized pairwise gossip algorithm (RK in the gossip setting) can lead to faster gossip algorithms and as a result the nodes of the network will converge to the average consensus faster both in number of iterations and in time. We do not intend to analyze the distributed behavior of the method (this is an ongoing research work). In our implementations we use three of the most popular graph topologies in the literature of wireless sensor networks. These are the line graph, cycle graph and the random geometric graph $G(n, r)$. In practice, $G(n, r)$ consider ideal for modeling wireless sensor networks, because of their particular formulation. In the experiments the 2-dimensional $G(n, r)$ is used which is formed by placing n nodes uniformly at random in a unit square with edges only between nodes that have euclidean distance less than the given radius r. To preserve the connectivity of $G(n, r)$ a radius $r = r(n) = \log(n)/n$ is used [152]. The AC problem is solved for the three aforementioned networks for both $n = 100$ and $n = 200$ number of nodes. We run mRK with several momentum parameters β for 10 trials and we plot their average. Our results are available in Figures 2.8 and 2.9.

Note that the vector of the initial values of the nodes can be chosen arbitrarily, and the proposed algorithms will find the average of these values. In Figures 2.8 and 2.9 the initial value of each node is chosen independently at random from the uniform distribution in the interval $(0, 1)$.

Experimental Results

By observing Figures 2.8 and 2.9, it is clear that the addition of the momentum term improves the performance of the popular pairwise randomized gossip (PRG) method [16]. The choice $\beta = 0.4$ as the momentum parameter improves the performance of the vanilla PRG for all networks under study and $\beta = 0.5$ is a good choice for the cases of the cycle and line graph. Note that for networks such as the cycle and line graphs there are known closed form expressions for the algebraic connectivity [31]. Thus, using equation (2.23), we can compute the exact values of the condition number $1/\lambda_{\min}^{+}$ for these networks. Interestingly, as we can see in Table 2.7

[7] Matrix $\mathbf{A}$ of the linear system is the incidence matrix of the graph and it is known that the Laplacian matrix is equal to $\mathbf{L} = \mathbf{A}^\top\mathbf{A}$, where $\|\mathbf{A}\|_F^2 = 2m$.

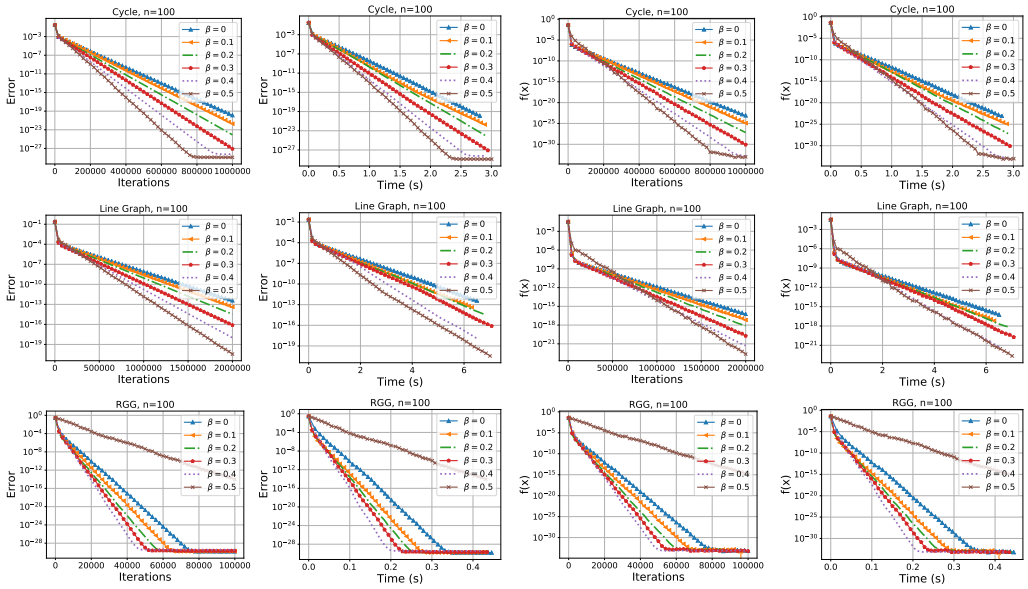


Figure 2.8: Performance of mPRG for several momentum parameters β for solving the average consensus problem in a cycle graph, line graph and random geometric graph $G(n,r)$ with $n = 100$ nodes. For the $G(n,r)$ to ensure connectivity of the network a radius $r = \sqrt{\log(n)/n}$ is used. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. The "Error" in the vertical axis represents the relative error $\|x^k - x^*\|^2_{\mathbf{B}}/\|x^0 - x^*\|^2_{\mathbf{B}} \overset{\mathbf{B}=\mathbf{I}, x^0=c}{=} \|x^k - x^*\|^2/\|c - x^*\|^2_{\mathbf{B}}$ and the function values $f(x^k)$ refer to function (2.19).

for $n = 100$ and $n = 200$ (number of nodes), the condition number $1/\lambda^+_{\min}$ appearing in the iteration complexity of our methods is not very large. This is in contrast with experimental observations from Section 2.7.1 where it was shown that the choice $\beta = 0.5$ is good for very ill conditioned problems only ($1/\lambda^+_{\min}$ very large).

Network	Formula for $\lambda^+_{\min}(\mathbf{L})$	$1/\lambda^+_{\min}$ for $n = 100$	$1/\lambda^+_{\min}$ for $n = 200$
Line	$2\,(1 - \cos(\pi/n))$	1013	4052
Cycle	$2\,(1 - \cos(2\pi/n))$	253	1013

Table 2.7: Algebraic connectivity of cycle and line graph for $n = 100$ and $n = 200$

2.8 Conclusion

In this chapter, we studied the convergence analysis of several stochastic optimization algorithms enriched with heavy ball momentum for solving stochastic optimization problems of special structure. We proved global, non-asymptotic linear convergence rates of all of these methods as well as accelerated linear rate for the case of the norm of expected iterates. We also introduced a new momentum strategy called *stochastic momentum* which is beneficial for the case of sparse data and proved linear convergence in this setting. We corroborated our theoretical results with extensive experimental testing.

Our work is amenable to further extensions. A natural extension of our results is the analysis of heavy ball momentum variants of our proposed methods (SGD,SN, SPP, etc) in the case of general convex or strongly convex functions. While we have shown that the expected iterates converge in an accelerated manner, it is an open problem whether an accelerated rate can be established for the expected distance, i.e., for $\mathbb{E}\left[\|x^k - x^*\|^2_{\mathbf{B}}\right]$. In our analysis we also focus on the case of fixed constant step-size and momentum parameters. A study of the effect of

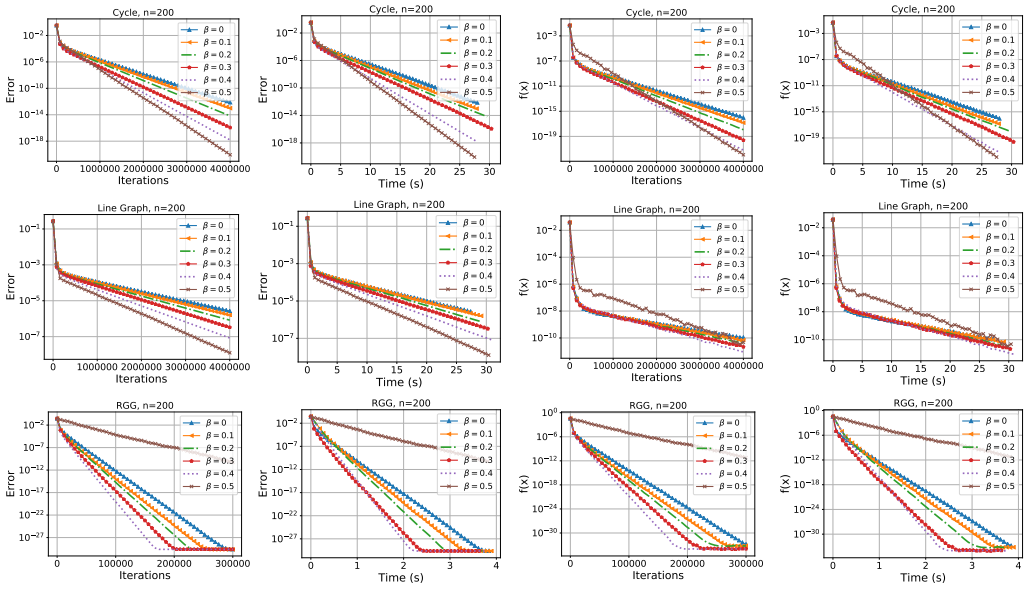


Figure 2.9: Performance of mPRG for several momentum parameters β for solving the average consensus problem in a cycle graph, line graph and random geometric graph $G(n,r)$ with $n = 200$ nodes. For the $G(n,r)$ to ensure connectivity of the network a radius $r = \sqrt{\log(n)/n}$ is used. The graphs in the first (second) column plot iterations (time) against residual error while those in the third (forth) column plot iterations (time) against function values. The "Error" in the vertical axis represents the relative error $\|x^k - x^*\|^2_{\mathbf{B}}/\|x^0 - x^*\|^2_{\mathbf{B}} \overset{\mathbf{B}=\mathbf{I}, x^0=c}{=} \|x^k - x^*\|^2/\|c - x^*\|^2$ and the function values $f(x^k)$ refer to function (2.19).

decreasing or adaptive choice of the parameters might provide novel insights.

The obtained results hold under the exactness condition which as we explain is very weak, allowing for virtually arbitrary distributions $\mathcal{D}$ from which the random matrices are drawn. One may wish to design optimized distributions in terms of the convergence rates or overall complexity.

Finally, we show how the addition of momentum on top of gossip algorithms can lead to faster convergence. An interesting question is to interpret the distributed nature of these algorithms and try to understand how we can improve the analysis using the properties of the underlying network. This is precisely what we are doing later in Chapter 4.

2.9 Proofs of Main Results

2.9.1 Technical lemmas

Lemma 16. *Fix $F^1 = F^0 \geq 0$ and let $\{F^k\}_{k\geq 0}$ be a sequence of nonnegative real numbers satisfying the relation*

$$F^{k+1} \leq a_1 F^k + a_2 F^{k-1}, \qquad \forall k \geq 1, \tag{2.24}$$

where $a_2 \geq 0$, $a_1 + a_2 < 1$ and at least one of the coefficients a_1, a_2 is positive. Then the sequence satisfies the relation $F^{k+1} \leq q^k(1+\delta)F^0$ for all $k \geq 1$, where $q = \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ and $\delta = q - a_1 \geq 0$. Moreover,

$$q \geq a_1 + a_2, \tag{2.25}$$

with equality if and only if $a_2 = 0$ (in which case $q = a_1$ and $\delta = 0$).

Proof. Choose $\delta = \frac{-a_1 + \sqrt{a_1^2 + 4a_2}}{2}$. We claim $\delta \geq 0$ and $a_2 \leq (a_1 + \delta)\delta$. Indeed, non-negativity of δ follows from $a_2 \geq 0$, while the second relation follows from the fact that δ satisfies

$$(a_1 + \delta)\delta - a_2 = 0. \tag{2.26}$$

In view of these two relations, adding δF_k to both sides of (2.24), we get

$$F^{k+1} + \delta F^k \leq (a_1 + \delta)F^k + a_2 F^{k-1} \leq (a_1 + \delta)(F^k + \delta F^{k-1}) = q(F^k + \delta F^{k-1}). \quad (2.27)$$

Let us now argue that $0 < q < 1$. Non-negativity of q follows from non-negativity of a_2. Clearly, as long as $a_2 > 0$, q is positive. If $a_2 = 0$, then $a_1 > 0$ by assumption, which implies that q is positive. The inequality $q < 1$ follows directly from the assumption $a_1 + a_2 < 1$. By unrolling the recurrence (2.27), we obtain $F^{k+1} \leq F^{k+1} + \delta F^k \leq q^k(F^1 + \delta F^0) = q^k(1 + \delta)F^0$.

Finally, let us establish (2.27). Noting that $a_1 = q - \delta$, and since in view of (2.26) we have $a_2 = q\delta$, we conclude that $a_1 + a_2 = q + \delta(q - 1) \leq q$, where the inequality follows from $q < 1$. □

Finally, let us present a simple lemma of an identity that we use in our main proofs. This preliminary result is known to hold for the case of Euclidean norms ($\mathbf{B} = \mathbf{I}$). We provide the proof for the more general $\mathbf{B}-$norm for completeness.

Lemma 17. *Let a, b, c be arbitrary vectors in $\mathbb{R}^n$ and let $\mathbf{B}$ be a positive definite matrix. Then the following identity holds: $2\langle a - c, c - b\rangle_{\mathbf{B}} = \|a - b\|^2_{\mathbf{B}} - \|c - b\|^2_{\mathbf{B}} - \|a - c\|^2_{\mathbf{B}}$,*

Proof.

$$\begin{aligned} LHS &= 2\langle a - c, c - b\rangle_{\mathbf{B}} = 2(a - c)^\top \mathbf{B}(c - b) \\ &= 2a^\top \mathbf{B}c - 2a^\top \mathbf{B}b - 2c^\top \mathbf{B}c + 2c^\top \mathbf{B}b \end{aligned}$$

and

$$\begin{aligned} RHS &= \|a - b\|^2_{\mathbf{B}} - \|c - b\|^2_{\mathbf{B}} - \|a - c\|^2_{\mathbf{B}} \\ &= (a - b)^\top \mathbf{B}(a - b) - (c - b)^\top \mathbf{B}(c - b) - (a - c)^\top \mathbf{B}(a - c) \\ &= a^\top \mathbf{B}a - a^\top \mathbf{B}b - b^\top \mathbf{B}a + b^\top \mathbf{B}b - c^\top \mathbf{B}c + c^\top \mathbf{B}b + b^\top \mathbf{B}c - b^\top \mathbf{B}b \\ &\quad -a^\top \mathbf{B}a + a^\top \mathbf{B}c + c^\top \mathbf{B}a - c^\top \mathbf{B}c \\ &= 2a^\top \mathbf{B}c - 2a^\top \mathbf{B}b - 2c^\top \mathbf{B}c + 2c^\top \mathbf{B}b \end{aligned}$$

LHS (left-hand side)=RHS (right-hand side) and this completes the proof. □

2.9.2 Proof of Theorem 8

First, we decompose

$$\begin{aligned} \|x^{k+1} - x^*\|^2_{\mathbf{B}} &= \|x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \beta(x^k - x^{k-1}) - x^*\|^2_{\mathbf{B}} \\ &= \underbrace{\|x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*\|^2_{\mathbf{B}}}_{①} \\ &\quad + \underbrace{2\langle x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*, \beta(x^k - x^{k-1})\rangle_{\mathbf{B}}}_{②} \\ &\quad + \underbrace{\beta^2\|x^k - x^{k-1}\|^2_{\mathbf{B}}}_{③}. \end{aligned} \quad (2.28)$$

We will now analyze the three expressions ①, ②, ③ separately. The first expression can be written as

$$\begin{aligned} ① &= \|x^k - x^*\|^2_{\mathbf{B}} - 2\omega\langle x^k - x^*, \nabla f_{\mathbf{S}_k}(x^k)\rangle_{\mathbf{B}} + \omega^2\|\nabla f_{\mathbf{S}_k}(x^k)\|^2_{\mathbf{B}} \\ &\overset{(1.38),(1.39)}{=} \|x^k - x^*\|^2_{\mathbf{B}} - 4\omega f_{\mathbf{S}_k}(x^k) + 2\omega^2 f_{\mathbf{S}_k}(x^k) \\ &= \|x^k - x^*\|^2_{\mathbf{B}} - 2\omega(2 - \omega)f_{\mathbf{S}_k}(x^k). \end{aligned} \quad (2.29)$$

We will now bound the second expression. First, we have

$$\begin{aligned}\textcircled{2} &= 2\beta\langle x^k - x^*, x^k - x^{k-1}\rangle_{\mathbf{B}} + 2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle_{\mathbf{B}} \\ &= 2\beta\langle x^k - x^*, x^k - x^*\rangle_{\mathbf{B}} + 2\beta\langle x^k - x^*, x^* - x^{k-1}\rangle_{\mathbf{B}} + 2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle_{\mathbf{B}} \\ &= 2\beta\|x^k - x^*\|^2_{\mathbf{B}} + 2\beta\langle x^k - x^*, x^* - x^{k-1}\rangle_{\mathbf{B}} + 2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle_{\mathbf{B}}.\end{aligned} \tag{2.30}$$

Using the identity from Lemma 17 for the vectors x^k, x^* and x^{k-1} we obtain:

$$2\langle x^k - x^*, x^* - x^{k-1}\rangle_{\mathbf{B}} = \|x^k - x^{k-1}\|^2_{\mathbf{B}} - \|x^{k-1} - x^*\|^2_{\mathbf{B}} - \|x^k - x^*\|^2_{\mathbf{B}}.$$

Substituting this into (2.30) gives

$$\textcircled{2} = \beta\|x^k - x^*\|^2_{\mathbf{B}} + \beta\|x^k - x^{k-1}\|^2_{\mathbf{B}} - \beta\|x^{k-1} - x^*\|^2_{\mathbf{B}} + 2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle_{\mathbf{B}}. \tag{2.31}$$

The third expression can be bounded as

$$\textcircled{3} = \beta^2\|(x^k - x^*) + (x^* - x^{k-1})\|^2_{\mathbf{B}} \leq 2\beta^2\|x^k - x^*\|^2_{\mathbf{B}} + 2\beta^2\|x^{k-1} - x^*\|^2_{\mathbf{B}}. \tag{2.32}$$

By substituting the bounds (2.29), (2.31), (2.32) into (2.28) we obtain

$$\begin{aligned}\|x^{k+1} - x^*\|^2_{\mathbf{B}} &\leq \|x^k - x^*\|^2_{\mathbf{B}} - 2\omega(2-\omega)f_{\mathbf{S}_k}(x^k) \\ &\quad +\beta\|x^k - x^*\|^2_{\mathbf{B}} + \beta\|x^k - x^{k-1}\|^2_{\mathbf{B}} - \beta\|x^{k-1} - x^*\|^2_{\mathbf{B}} \\ &\quad +2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle_{\mathbf{B}} + 2\beta^2\|x^k - x^*\|^2_{\mathbf{B}} + 2\beta^2\|x^{k-1} - x^*\|^2_{\mathbf{B}} \\ &\leq (1 + 3\beta + 2\beta^2)\|x^k - x^*\|^2_{\mathbf{B}} + (\beta + 2\beta^2)\|x^{k-1} - x^*\|^2_{\mathbf{B}} - 2\omega(2-\omega)f_{\mathbf{S}_k}(x^k) \\ &\quad +2\omega\beta\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle_{\mathbf{B}}.\end{aligned}$$

Now by first taking expectation with respect to $\mathbf{S}_k$, we obtain:

$$\begin{aligned}\mathbb{E}_{\mathbf{S}_k}[\|x^{k+1} - x^*\|^2_{\mathbf{B}}] &\leq (1 + 3\beta + 2\beta^2)\|x^k - x^*\|^2_{\mathbf{B}} + (\beta + 2\beta^2)\|x^{k-1} - x^*\|^2_{\mathbf{B}} \\ &\quad -2\omega(2-\omega)f(x^k) + 2\omega\beta\langle \nabla f(x^k), x^{k-1} - x^k\rangle_{\mathbf{B}} \\ &\leq (1 + 3\beta + 2\beta^2)\|x^k - x^*\|^2_{\mathbf{B}} + (\beta + 2\beta^2)\|x^{k-1} - x^*\|^2_{\mathbf{B}} \\ &\quad -2\omega(2-\omega)f(x^k) + 2\omega\beta(f(x^{k-1}) - f(x^k)) \\ &= (1 + 3\beta + 2\beta^2)\|x^k - x^*\|^2_{\mathbf{B}} + (\beta + 2\beta^2)\|x^{k-1} - x^*\|^2_{\mathbf{B}} \\ &\quad -(2\omega(2-\omega) + 2\omega\beta)f(x^k) + 2\omega\beta f(x^{k-1}).\end{aligned}$$

where in the second step we used the inequality $\langle \nabla f(x^k), x^{k-1} - x^k\rangle \leq f(x^{k-1}) - f(x^k)$ and the fact that $\omega\beta \geq 0$, which follows from the assumptions. We now apply inequalities (1.36) and (1.37), obtaining

$$\begin{aligned}\mathbb{E}_{\mathbf{S}_k}[\|x^{k+1} - x^*\|^2_{\mathbf{B}}] &\leq \underbrace{(1 + 3\beta + 2\beta^2 - (\omega(2-\omega) + \omega\beta)\lambda^+_{\min})}_{a_1}\|x^k - x^*\|^2_{\mathbf{B}} \\ &\quad + \underbrace{(\beta + 2\beta^2 + \omega\beta\lambda_{\max})}_{a_2}\|x^{k-1} - x^*\|^2_{\mathbf{B}}.\end{aligned}$$

By taking expectation again, and letting $F^k := \mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}]$, we get the relation

$$F^{k+1} \leq a_1 F^k + a_2 F^{k-1}. \tag{2.33}$$

It suffices to apply Lemma 16 to the relation (2.33). The conditions of the lemma are satisfied. Indeed, $a_2 \geq 0$, and if $a_2 = 0$, then $\beta = 0$ and hence $a_1 = 1 - \omega(2-\omega)\lambda^+_{\min} > 0$. The condition $a_1 + a_2 < 1$ holds by assumption.

The convergence result in function values, $\mathbb{E}[f(x^k)]$, follows as a corollary by applying inequality (1.36) to (2.5).

2.9.3 Proof of Theorem 10

Let $p^t = \frac{\beta}{1-\beta}(x^t - x^{t-1})$ and $d^t = \|x^t + p^t - x^*\|_{\mathbf{B}}^2$. In view of (2.4), we can write

$$\begin{aligned}
x^{t+1} + p^{t+1} &= x^{t+1} + \frac{\beta}{1-\beta}(x^{t+1} - x^t) \\
&\overset{(2.4)}{=} x^t - \omega \nabla f_{\mathbf{S}_t}(x^t) + \beta(x^t - x^{t-1}) + \frac{\beta}{1-\beta}\left(-\omega \nabla f_{\mathbf{S}_t}(x^t) + \beta(x^t - x^{t-1})\right) \\
&= x^t - \left[\omega + \frac{\beta}{1-\beta}\omega\right]\nabla f_{\mathbf{S}_t}(x^t) + \left[\beta + \frac{\beta^2}{1-\beta}\right](x^t - x^{t-1}) \\
&= x^t - \frac{\omega}{1-\beta}\nabla f_{\mathbf{S}_t}(x^t) + \frac{\beta}{1-\beta}(x^t - x^{t-1}) \\
&= x^t + p^t - \frac{\omega}{1-\beta}\nabla f_{\mathbf{S}_t}(x^t) \qquad (2.34)
\end{aligned}$$

and therefore

$$\begin{aligned}
d^{t+1} &\overset{(2.34)}{=} \left\| x^t + p^t - \frac{\omega}{1-\beta}\nabla f_{\mathbf{S}_t}(x^t) - x^* \right\|_{\mathbf{B}}^2 \\
&= d^t - 2\frac{\omega}{1-\beta}\langle x^t + p^t - x^*, \nabla f_{\mathbf{S}_t}(x^t)\rangle_{\mathbf{B}} + \frac{\omega^2}{(1-\beta)^2}\|\nabla f_{\mathbf{S}_t}(x^t)\|_{\mathbf{B}}^2 \\
&= d^t - \frac{2\omega}{1-\beta}\langle x^t - x^*, \nabla f_{\mathbf{S}_t}(x^t)\rangle_{\mathbf{B}} - \frac{2\omega\beta}{(1-\beta)^2}\langle x^t - x^{t-1}, \nabla f_{\mathbf{S}_t}(x^t)\rangle_{\mathbf{B}} \\
&\quad + \frac{\omega^2}{(1-\beta)^2}\|\nabla f_{\mathbf{S}_t}(x^t)\|_{\mathbf{B}}^2.
\end{aligned}$$

Taking expectation with respect to the random matrix $\mathbf{S}_t$ we obtain:

$$\begin{aligned}
\mathbb{E}_{\mathbf{S}_t}[d^{t+1}] &= d^t - \frac{2\omega}{1-\beta}\langle x^t - x^*, \nabla f(x^t)\rangle_{\mathbf{B}} - \frac{2\omega\beta}{(1-\beta)^2}\langle x^t - x^{t-1}, \nabla f(x^t)\rangle_{\mathbf{B}} \\
&\quad + \frac{\omega^2}{(1-\beta)^2}2f(x^t) \\
&\overset{(1.40)}{=} d^t - \frac{4\omega}{1-\beta}f(x^t) - \frac{2\omega\beta}{(1-\beta)^2}\langle x^t - x^{t-1}, \nabla f(x^t)\rangle_{\mathbf{B}} + \frac{\omega^2}{(1-\beta)^2}2f(x^t) \\
&\leq d^t - \frac{4\omega}{1-\beta}f(x^t) - \frac{2\omega\beta}{(1-\beta)^2}[f(x^t) - f(x^{t-1})] + \frac{\omega^2}{(1-\beta)^2}2f(x^t) \\
&= d^t + \left[-\frac{4\omega}{1-\beta} - \frac{2\omega\beta}{(1-\beta)^2} + \frac{2\omega^2}{(1-\beta)^2}\right]f(x^t) + \frac{2\omega\beta}{(1-\beta)^2}f(x^{t-1}),
\end{aligned}$$

where the inequality follows from convexity of f. After rearranging the terms we get

$$\mathbb{E}_{\mathbf{S}_t}[d^{t+1}] + \frac{2\omega\beta}{(1-\beta)^2}f(x^t) + \alpha f(x^t) \leq d^t + \frac{2\omega\beta}{(1-\beta)^2}f(x^{t-1}),$$

where $\alpha = \frac{4\omega}{1-\beta} - \frac{2\omega^2}{(1-\beta)^2} > 0$. Taking expectations again and using the tower property, we get

$$\theta^{t+1} + \alpha\mathbb{E}[f(x^t)] \leq \theta^t, \qquad t = 1, 2, \ldots, \qquad (2.35)$$

where $\theta^t = \mathbb{E}[d^t] + \frac{2\omega\beta}{(1-\beta)^2}\mathbb{E}[f(x^{t-1})]$. By summing up (2.35) for $t = 1, \ldots, k$ we get

$$\sum_{t=1}^{k}\mathbb{E}[f(x^t)] \leq \frac{\theta^1 - \theta^{k-1}}{\alpha} \leq \frac{\theta^1}{\alpha}. \qquad (2.36)$$

Finally, using Jensen's inequality, we get

$$\mathbb{E}[f(\hat{x}^k)] = \mathbb{E}\left[f\left(\frac{1}{k}\sum_{t=1}^{k} x^t\right)\right] \leq \mathbb{E}\left[\frac{1}{k}\sum_{t=1}^{k} f(x^t)\right] = \frac{1}{k}\sum_{t=1}^{k}\mathbb{E}[f(x^t)] \overset{(2.36)}{\leq} \frac{\theta^1}{\alpha k}.$$

It remains to note that $\theta^1 = \|x^0 - x^*\|^2_{\mathbf{B}} + \frac{2\omega\beta}{(1-\beta)^2} f(x^0)$.

2.9.4 Proof of Theorem 11

In the proof of Theorem 11 the following two lemmas are used.

Lemma 18 ([168]). *Assume exactness. Let $x \in \mathbb{R}^n$ and $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x)$. If $\lambda_i = 0$, then $u_i^\top \mathbf{B}^{1/2}(x - x^*) = 0$.*

Lemma 19 ([48, 53]). *Consider the second degree linear homogeneous recurrence relation:*

$$r^{k+1} = a_1 r^k + a_2 r^{k-1} \tag{2.37}$$

with initial conditions $r^0, r^1 \in \mathbb{R}$. Assume that the constant coefficients a_1 and a_2 satisfy the inequality $a_1^2 + 4a_2 < 0$ (the roots of the characteristic equation $t^2 - a_1 t - a_2 = 0$ are imaginary). Then there are complex constants C_0 and C_1 (depending on the initial conditions r^0 and r^1) such that:

$$r^k = 2M^k(C_0\cos(\theta k) + C_1 \sin(\theta k))$$

where $M = \left(\sqrt{\frac{a_1^2}{4} + \frac{(-a_1^2 - 4a_2)}{4}}\right) = \sqrt{-a_2}$ and θ is such that $a_1 = 2M\cos(\theta)$ and $\sqrt{-a_1^2 - 4a_2} = 2M\sin(\theta)$.

We can now turn to the proof of Theorem 11. Plugging in the expression for the stochastic gradient, mSGD can be written in the form

$$\begin{aligned} x^{k+1} &= x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \beta(x^k - x^{k-1}) \\ &\overset{(1.13)}{=} x^k - \omega\mathbf{B}^{-1}\mathbf{Z}_k(x^k - x^*) + \beta(x^k - x^{k-1}). \end{aligned} \tag{2.38}$$

Subtracting x^* from both sides of (2.38), we get

$$\begin{aligned} x^{k+1} - x^* &= (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*) + \beta(x^k - x^* + x^* - x^{k-1}) \\ &= \left((1+\beta)\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k\right)(x^k - x^*) - \beta(x^{k-1} - x^*). \end{aligned}$$

Multiplying the last identity from the left by $\mathbf{B}^{1/2}$, we get

$$\mathbf{B}^{1/2}(x^{k+1} - x^*) = \left((1+\beta)\mathbf{I} - \omega\mathbf{B}^{-1/2}\mathbf{Z}_k\mathbf{B}^{-1/2}\right)\mathbf{B}^{1/2}(x^k - x^*) - \beta\mathbf{B}^{1/2}(x^{k-1} - x^*).$$

Taking expectations, conditioned on x^k (that is, the expectation is with respect to $\mathbf{S}_k$):

$$\mathbf{B}^{1/2}\mathbb{E}[x^{k+1} - x^* \mid x^k] = \left((1+\beta)\mathbf{I} - \omega\mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}\right)\mathbf{B}^{1/2}(x^k - x^*) - \beta\mathbf{B}^{1/2}(x^{k-1} - x^*). \tag{2.39}$$

Taking expectations again, and using the tower property, we get

$$\begin{aligned} \mathbf{B}^{1/2}\mathbb{E}[x^{k+1} - x^*] &= \mathbf{B}^{1/2}\mathbb{E}\left[\mathbb{E}[x^{k+1} - x^* \mid x^k]\right] \\ &\overset{(2.39)}{=} \left((1+\beta)\mathbf{I} - \omega\mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}\right)\mathbf{B}^{1/2}\mathbb{E}[x^k - x^*] - \beta\mathbf{B}^{1/2}\mathbb{E}[x^{k-1} - x^*]. \end{aligned}$$

Plugging the eigenvalue decomposition $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ of the matrix $\mathbf{W} = \mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}$ into the above, and multiplying both sides from the left by $\mathbf{U}^\top$, we obtain

$$\mathbf{U}^\top\mathbf{B}^{1/2}\mathbb{E}[x^{k+1} - x^*] = \mathbf{U}^\top\left((1+\beta)\mathbf{I} - \omega\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top\right)\mathbf{B}^{1/2}\mathbb{E}[x^k - x^*] - \beta\mathbf{U}^\top\mathbf{B}^{1/2}\mathbb{E}[x^{k-1} - x^*]. \tag{2.40}$$

Let us define $s^k := \mathbf{U}^\top \mathbf{B}^{1/2}\mathbb{E}[x^k - x^*] \in \mathbb{R}^n$. Then relation (2.40) takes the form of the recursion

$$s^{k+1} = [(1+\beta)\mathbf{I} - \omega\mathbf{\Lambda}]s^k - \beta s^{k-1},$$

which can be written in a coordinate-by-coordinate form as follows:

$$s_i^{k+1} = [(1+\beta) - \omega\lambda_i]s_i^k - \beta s_i^{k-1} \quad \text{for all} \quad i = 1, 2, 3, ..., n, \tag{2.41}$$

where s_i^k indicates the ith coordinate of s^k.

We will now fix i and analyze recursion (2.41) using Lemma 19. Note that (2.41) is a second degree linear homogeneous recurrence relation of the form (2.37) with $a_1 = 1 + \beta - \omega\lambda_i$ and $a_2 = -\beta$. Recall that $0 \le \lambda_i \le 1$ for all i. Since we assume that $0 < \omega \le 1/\lambda_{\max}$, we know that $0 \le \omega\lambda_i \le 1$ for all i. We now consider two cases:

1. $\lambda_i = 0$.

   In this case, (2.41) takes the form:

   $$s_i^{k+1} = (1+\beta)s_i^k - \beta s_i^{k-1}. \tag{2.42}$$

   Applying Proposition 9, we know that $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0) = \Pi_{\mathcal{L},\mathbf{B}}(x^1)$. Using Lemma 18 twice, once for $x = x^0$ and then for $x = x^1$, we observe that $s_i^0 = u_i^\top \mathbf{B}^{1/2}(x^0 - x^*) = 0$ and $s_i^1 = u_i^\top \mathbf{B}^{1/2}(x^1 - x^*) = 0$. Finally, in view of (2.42) we conclude that

   $$s_i^k = 0 \quad \text{for all} \quad k \ge 0. \tag{2.43}$$

2. $\lambda_i > 0$.

   Since $0 < \omega\lambda_i \le 1$ and $\beta \ge 0$, we have $1 + \beta - \omega\lambda_i \ge 0$ and hence

   $$a_1^2 + 4a_2 = (1 + \beta - \omega\lambda_i)^2 - 4\beta \le (1 + \beta - \omega\lambda_{\min}^+)^2 - 4\beta < 0,$$

   where the last inequality can be shown to hold[8] for $\left(1 - \sqrt{\omega\lambda_{\min}^+}\right)^2 < \beta < 1$. Applying Lemma 19 the following bound can be deduced

   $$s_i^k \;=\; 2(-a_2)^{k/2}(C_0\cos(\theta k) + C_1\sin(\theta k)) \;\le\; 2\beta^{k/2}P_i, \tag{2.44}$$

   where P_i is a constant depending on the initial conditions (we can simply choose $P_i = |C_0| + |C_1|$).

Now putting the two cases together, for all $k \ge 0$ we have

$$\begin{aligned}
\|\mathbb{E}[x^k - x^*]\|_{\mathbf{B}}^2 &= \mathbb{E}[x^k - x^*]^\top \mathbf{B}\mathbb{E}[x^k - x^*] \;=\; \mathbb{E}[x^k - x^*]\mathbf{B}^{1/2}\mathbf{U}\mathbf{U}^\top\mathbf{B}^{1/2}\mathbb{E}[x^k - x^*] \\
&= \|\mathbf{U}^\top\mathbf{B}^{1/2}\mathbb{E}[x^k - x^*]\|^2 \;=\; \|s^k\|^2 \;=\; \sum_{i=1}^n (s_i^k)^2 \\
&= \sum_{i:\lambda_i=0}(s_i^k)^2 + \sum_{i:\lambda_i>0}(s_i^k)^2 \overset{(2.43)}{=} \sum_{i:\lambda_i>0}(s_i^k)^2 \\
&\overset{(2.44)}{\le} \sum_{i:\lambda_i>0} 4\beta^k P_i^2 \\
&= \beta^k C,
\end{aligned}$$

where $C = 4\sum_{i:\lambda_i>0} P_i^2$.

[8] The lower bound on β is tight. However, the upper bound is not. However, we do not care much about the regime of large β as β is the convergence rate, and hence is only interesting if smaller than 1.

2.9.5 Proof of Theorem 14

The proof follows a similar pattern to that of Theorem 8. However, stochasticity in the momentum term introduces an additional layer of complexity, which we shall tackle by utilizing a more involved version of the tower property.

For simplicity, let $i = i_k$ and $r_i^k := e_i^\top(x^k - x^{k-1})e_i$. First, we decompose

$$\begin{aligned}\|x^{k+1} - x^*\|^2 &= \|x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \gamma r_i^k - x^*\|^2 \\ &= \|x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*\|^2 + 2\langle x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*, \gamma r_i^k\rangle + \gamma^2\|r_i^k\|^2 \quad (2.45)\end{aligned}$$

We shall use the tower property in the form

$$\mathbb{E}[\mathbb{E}[\mathbb{E}[X \mid x^k, \mathbf{S}_k] \mid x^k]] = \mathbb{E}[X], \quad (2.46)$$

where X is some random variable. We shall perform the three expectations in order, from the innermost to the outermost. Applying the inner expectation to the identity (2.45), we get

$$\begin{aligned}\mathbb{E}[\|x^{k+1} - x^*\|^2 \mid x^k, \mathbf{S}_k] &= \underbrace{\mathbb{E}[\|x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*\|^2 \mid x^k, \mathbf{S}_k]}_{①} \\ &\quad + \underbrace{\mathbb{E}[2\langle x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*, \gamma r_i^k\rangle \mid x^k, \mathbf{S}_k]}_{②} \\ &\quad + \underbrace{\mathbb{E}[\gamma^2\|r_i^k\|^2 \mid x^k, \mathbf{S}_k]}_{③}. \quad (2.47)\end{aligned}$$

We will now analyze the three expressions ①, ②, ③ separately. The first expression is constant under the expectation, and hence we can write

$$\begin{aligned}① &= \|x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*\|^2 \\ &= \|x^k - x^*\|^2 - 2\omega\langle x^k - x^*, \nabla f_{\mathbf{S}_k}(x^k)\rangle + \omega^2\|\nabla f_{\mathbf{S}_k}(x^k)\|^2 \\ &\overset{(1.38)+(1.39)}{=} \|x^k - x^*\|^2 - 4\omega f_{\mathbf{S}_k}(x^k) + 2\omega^2 f_{\mathbf{S}_k}(x^k) \\ &= \|x^k - x^*\|^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x^k). \quad (2.48)\end{aligned}$$

We will now bound the second expression. Using the identity

$$\mathbb{E}[r_i^k \mid x^k, \mathbf{S}_k] = \mathbb{E}_i[r_i^k] = \sum_{i=1}^n \frac{1}{n} r_i^k = \frac{1}{n}(x^k - x^{k-1}), \quad (2.49)$$

we can write

$$\begin{aligned}② &= \mathbb{E}[2\langle x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*, \gamma r_i^k\rangle \mid x^k, \mathbf{S}_k] \\ &= 2\langle x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*, \gamma\mathbb{E}[r_i^k \mid x^k, \mathbf{S}_k]\rangle \\ &\overset{(2.49)}{=} 2\langle x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) - x^*, \frac{\gamma}{n}(x^k - x^{k-1})\rangle \\ &= 2\frac{\gamma}{n}\langle x^k - x^*, x^k - x^{k-1}\rangle + 2\omega\frac{\gamma}{n}\langle\nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle \\ &= 2\frac{\gamma}{n}\langle x^k - x^*, x^k - x^*\rangle + 2\frac{\gamma}{n}\langle x^k - x^*, x^* - x^{k-1}\rangle + 2\omega\frac{\gamma}{n}\langle\nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle \\ &= 2\frac{\gamma}{n}\|x^k - x^*\|^2 + 2\frac{\gamma}{n}\langle x^k - x^*, x^* - x^{k-1}\rangle + 2\omega\frac{\gamma}{n}\langle\nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle. \quad (2.50)\end{aligned}$$

Using the fact that for arbitrary vectors $a, b, c \in \mathbb{R}^n$ we have the identity $2\langle a - c, c - b\rangle = \|a - b\|^2 - \|c - b\|^2 - \|a - c\|^2$, we obtain

$$2\langle x^k - x^*, x^* - x^{k-1}\rangle = \|x^k - x^{k-1}\|^2 - \|x^{k-1} - x^*\|^2 - \|x^k - x^*\|^2.$$

Substituting this into (2.50) gives

$$\textcircled{2} = \frac{\gamma}{n}\|x^k - x^*\|^2 + \frac{\gamma}{n}\|x^k - x^{k-1}\|^2 - \frac{\gamma}{n}\|x^{k-1} - x^*\|^2 + 2\omega\frac{\gamma}{n}\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle. \tag{2.51}$$

The third expression can be bound as

$$\begin{aligned}
\textcircled{3} &= \mathbb{E}[\gamma^2\|r_i^k\|^2 \mid x^k, \mathbf{S}_k] \\
&= \gamma^2\mathbb{E}_i[\|r_i^k\|^2] \\
&= \gamma^2\sum_{i=1}^n \frac{1}{n}(x_i^k - x_i^{k-1})^2 \\
&= \frac{\gamma^2}{n}\|x^k - x^{k-1}\|^2 \\
&= \frac{\gamma^2}{n}\|(x^k - x^*) + (x^* - x^{k-1})\|^2 \\
&\leq \frac{2\gamma^2}{n}\|x^k - x^*\|^2 + \frac{2\gamma^2}{n}\|x^{k-1} - x^*\|^2.
\end{aligned} \tag{2.52}$$

By substituting the bounds (2.48), (2.51), (2.52) into (2.47) we obtain

$$\begin{aligned}
\mathbb{E}[\|x^{k+1} - x^*\|^2 \mid x^k, \mathbf{S}_k] \quad \leq \quad & \|x^k - x^*\|^2 - 2\omega(2-\omega)f_{\mathbf{S}_k}(x^k) \\
& + \frac{\gamma}{n}\|x^k - x^*\|^2 + \frac{\gamma}{n}\|x^k - x^{k-1}\|^2 - \frac{\gamma}{n}\|x^{k-1} - x^*\|^2 \\
& + 2\omega\frac{\gamma}{n}\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle + 2\frac{\gamma^2}{n}\|x^k - x^*\|^2 \qquad (2.53) \\
& + 2\frac{\gamma^2}{n}\|x^{k-1} - x^*\|^2 \\
\overset{(2.32)}{\leq} \quad & \left(1 + 3\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^k - x^*\|^2 + \left(\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^{k-1} - x^*\|^2 \\
& - 2\omega(2-\omega)f_{\mathbf{S}_k}(x^k) + 2\omega\frac{\gamma}{n}\langle \nabla f_{\mathbf{S}_k}(x^k), x^{k-1} - x^k\rangle. \qquad (2.54)
\end{aligned}$$

We now take the middle expectation (see (2.46)) and apply it to inequality (2.54):

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[\|x^{k+1} - x^*\|^2 \mid x^k, \mathbf{S}_k] \mid x^k] \quad \leq \quad & \left(1 + 3\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^k - x^*\|^2 + \left(\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^{k-1} - x^*\|^2 \\
& - 2\omega(2-\omega)f(x^k) + 2\omega\frac{\gamma}{n}\langle \nabla f(x^k), x^{k-1} - x^k\rangle \\
\leq \quad & \left(1 + 3\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^k - x^*\|^2 + \left(\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^{k-1} - x^*\|^2 \\
& - 2\omega(2-\omega)f(x^k) + 2\omega\frac{\gamma}{n}(f(x^{k-1}) - f(x^k)) \\
= \quad & \left(1 + 3\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^k - x^*\|^2 + \left(\frac{\gamma}{n} + 2\frac{\gamma^2}{n}\right)\|x^{k-1} - x^*\|^2 \\
& - \left(2\omega(2-\omega) + 2\omega\frac{\gamma}{n}\right)f(x^k) + 2\omega\frac{\gamma}{n}f(x^{k-1}).
\end{aligned}$$

where in the second step we used the inequality $\langle \nabla f(x^k), x^{k-1} - x^k\rangle \leq f(x^{k-1}) - f(x^k)$ and the fact that $\omega\gamma \geq 0$, which follows from the assumptions. We now apply inequalities (1.36)

and (1.37), obtaining

$$\mathbb{E}[\mathbb{E}[\|x^{k+1}-x^*\|^2 \mid x^k, \mathbf{S}_k] \mid x^k] \quad \leq \quad \underbrace{\left(1+3\frac{\gamma}{n}+2\frac{\gamma^2}{n}-\left(\omega(2-\omega)+\omega\frac{\gamma}{n}\right)\lambda_{\min}^+\right)}_{a_1}\|x^k-x^*\|^2$$
$$+\underbrace{\frac{1}{n}\left(\gamma+2\gamma^2+\omega\gamma\lambda_{\max}\right)}_{a_2}\|x^{k-1}-x^*\|^2.$$

By taking expectation again (outermost expectation in the tower rule (2.46)), and letting $F^k := \mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}]$, we get the relation

$$F^{k+1} \leq a_1 F^k + a_2 F^{k-1}. \tag{2.55}$$

It suffices to apply Lemma 16 to the relation (2.33). The conditions of the lemma are satisfied. Indeed, $a_2 \geq 0$, and if $a_2 = 0$, then $\gamma = 0$ and hence $a_1 = 1-\omega(1-\omega)\lambda_{\min}^+ > 0$. The condition $a_1 + a_2 < 1$ holds by assumption.

The convergence result in function values follows as a corollary by applying inequality (1.36) to (2.14).

Chapter 3

Inexact Randomized Iterative Methods

3.1 Introduction

In the era of big data where data sets become continuously larger, randomized iterative methods become very popular and are increasingly playing a major role in areas such as numerical linear algebra, scientific computing and optimization. They are preferred mainly because of their cheap per-iteration cost which leads to improvements in terms of complexity upon classical results by orders of magnitude. In addition they can easily scale to extreme dimensions. However, a common feature of these methods is that in their update rule a particular subproblem needs to be solved *exactly*. In a large scale setting, often this step is computationally very expensive. The purpose of this work is to reduce the cost of this step by allowing for *inexact updates* in the stochastic methods under study.

3.1.1 The setting

In this chapter we are interested to solve the three closely related problems described in the previous chapters. As a reminder, these are:

(i) stochastic quadratic optimization (1.6),

(ii) best approximation (1.22), and

(iii) (bounded) concave quadratic maximization (1.26).

In particular we propose and analyze *inexact* variants of the exact algorithms presented in the introduction of this thesis for solving the above problems. Among the methods studied are: stochastic gradient descent (SGD), stochastic Newton (SN), stochastic proximal point (SPP), sketch and project method (SPM) and stochastic subspace ascent (SDSA). In all of these methods, a certain potentially expensive calculation/operation needs to be performed in each step; it is this operation that we propose to be performed *inexactly*. For instance, in the case of SGD, it is the computation of the stochastic gradient $\nabla f_{\mathbf{S}_k}(x^k)$, in the case of SPM is the computation of the projection $\Pi_{\mathcal{L}_{\mathbf{S}},\mathbf{B}}(x^k)$, and in the case of SDSA it is the computation of the dual update $\mathbf{S}_k\lambda^k$.

We perform an iteration complexity analysis under an abstract notion of inexactness and also under a more structured form of inexactness appearing in practical scenarios. An inexact solution of these subproblems can be obtained much more quickly than the exact solution. Since in practical applications the savings thus obtained are larger than the increase in the number of iterations needed for convergence, our inexact methods can be dramatically faster.

3.1.2 Structure of the chapter and main contributions

Let us now outline the main contribution and the structure of this chapter.

Assumption on the Inexactness error ϵ^k	ω	Upper Bounds	Theorem
Assumption 1a	$(0,2)$	$\rho^{k/2}\|x^0-x^*\|_{\mathbf{B}}+\sum_{i=0}^{k-1}\rho^{\frac{k-1-i}{2}}\sigma_i$	20
Assumption 1b	$(0,2)$	$\left(\sqrt{\rho}+q\right)^{2k}\|x^0-x^*\|^2_{\mathbf{B}}$	22
Assumptions 1,2	$(0,2)$	$\rho^k\|x^0-x^*\|^2_{\mathbf{B}}+\sum_{i=0}^{k-1}\rho^{k-1-i}\bar{\sigma}_i^2$	23(i)
Assumptions 1b,2	$(0,2)$	$\left(\rho+q^2\right)^k\|x^0-x^*\|^2_{\mathbf{B}}$	23(ii)
Assumptions 1c,2	$(0,2)$	$\left(\rho+q^2\lambda^+_{\min}\right)^k\|x^0-x^*\|^2_{\mathbf{B}}$	23(iii)

Table 3.1: Summary of the iteration complexity results obtained in this chapter. ω denotes the stepsize (relaxation parameter) of the method. In all cases, $x^*=\Pi_{\mathcal{L},\mathbf{B}}(x^0)$ and $\rho=1-\omega(2-\omega)\lambda^+_{\min}\in(0,1)$ are the quantities appear in the convergence results (here $\lambda^+_{\min}$ denotes the minimum non zero eigenvalue of matrix $\mathbf{W}$, see equation (2.18)). Inexactness parameter q is chosen always in such a way to obtain linear convergence and it can be seen as the quantity that controls the inexactness. In all theorems the quantity of convergence is $\mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}]$ (except in Theorem 20 where we analyze $\mathbb{E}[\|x^k-x^*\|_{\mathbf{B}}]$). As we show in Section 3.5, under similar assumptions, iSDSA has exactly the same convergence with iBasic but the upper bounds of the third column are related to the dual function values $\mathbb{E}\left[D(y^*)-D(y^0)\right]$.

In Section 3.2 we describe the subproblems and introduce two notions of inexactness (abstract and structured) that will be used in the rest of this chapter. The Inexact Basic Method (iBasic) is also presented. iBasic is a method that simultaneously captures inexact variants of the algorithms (1.17), (1.18), (1.19) for solving the stochastic optimization problem (1.6) and algorithm (1.25) for solving the best approximation problem (1.22). It is an inexact variant of the *Basic Method*, first presented in [168], where the inexactness is introduced by the addition of an inexactness error ϵ^k in the original update rule. We illustrate the generality of iBasic by presenting popular algorithms that can be cast as special cases.

In Section 3.3 we establish convergence results of iBasic under general assumptions on the inexactness error ϵ^k of its update rule (see Algorithm 4). In this part we do not focus on any specific mechanisms which lead to inexactness; we treat the problem abstractly. However, such errors appear often in practical scenarios and can be associated with inaccurate numerical solvers, quantization, sparsification and compression mechanisms. In particular, we introduce several abstract assumptions on the inexactness level and describe our generic convergence results. For all assumptions we establish linear rate of decay of the quantity $\mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}]$ (i.e. L2 convergence)[1].

Subsequently, in Section 3.4 we apply our general convergence results to a more structured notion of inexactness error and propose a concrete mechanisms leading to such errors. We provide theoretical guarantees for this method in situations when a linearly convergent iterative method (e.g., Conjugate Gradient) is used to solve the subproblem inexactly. We also highlight the importance of the dual viewpoint through a sketch-and-project interpretation.

In Section 3.5 we study an inexact variant of SDSA, which we called iSDSA, for directly solving the dual problem (1.26). We provide a correspondence between iBasic and iSDSA and we show that the random iterates of iBasic arise as affine images of iSDSA. We consider both abstract and structured inexactness errors and provide linearly convergent rates in terms of the dual function suboptimality $\mathbb{E}\left[D(y^*)-D(y^0)\right]$.

Finally, in Section 3.6 we evaluate the performance of the proposed inexact methods through numerical experiments and show the benefits of our approach on both synthetic and real datasets. Concluding remarks are given in Section 3.7.

A summary of the convergence results of iBasic under several assumptions on the inexactness error with pointers to the relevant theorems is available in Table 3.1. We highlight that similar convergence results can be also obtained for iSDSA in terms of the dual function suboptimality $\mathbb{E}\left[D(y^*)-D(y^0)\right]$ (check Section 3.5 for more details on iSDSA).

[1] As we explain later, a convergence of the expected function values of problem 1.6 can be easily obtained as a corollary of L2 convergence.

3.1.3 Notation

Following the rest of this thesis, with boldface upper-case letters we denote matrices and $\mathbf{I}$ is the identity matrix. By $\mathcal{L}$ we denote the solution set of the linear system $\mathbf{A}x = b$. By $\mathcal{L}_{\mathbf{S}}$, where $\mathbf{S}$ is a random matrix, we denote the solution set of the *sketched* linear system $\mathbf{S}^\top\mathbf{A}x = \mathbf{S}^\top b$. In general, we use $*$ to express the exact solution of a sub-problem and $\approx$ to indicate its inexact variant. Unless stated otherwise, throughout the chapter, x^* is the projection of x^0 onto $\mathcal{L}$ in the $\mathbf{B}$-norm: $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$.

3.2 Inexact Update Rules

In this section we start by explaining the key sub-problems that need to be solved exactly in the update rules of the previously described methods. We present iBasic, a method that solves problems (1.6) and (1.22) and we show how by varying the main parameters of the method we recover inexact variants of popular algorithms as special cases. Finally closely related work on inexact algorithms for solving different problems is also presented.

3.2.1 Expensive sub-problems in update rules

Let us devote this subsection on explaining how the inexactness can be introduced in the current exact update rules of SGD[2] (1.17), Sketch and Project (1.25) and SDSA (1.29) for solving the stochastic optimization, best approximation and the dual problem respectively. As we have shown these methods solve closely related problems and the key subproblems in their update rule are similar. However the introduction of inexactness in the update rule of each one of them can have different interpretation.

For example for the case of SGD for solving the stochastic optimization problem (1.6) (see also Section 3.4.1 and 3.4.2 for more details), if we define $\lambda_*^k = (\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top(b - \mathbf{A}x^k)$ then the stochastic gradient of function f becomes $\nabla f_{\mathbf{S}_k}(x^k) \overset{(1.13)}{=} -\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_*^k$ and the update rule of SGD takes the form: $x^{k+1} = x^k + \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_*^k$. Clearly in this update the expensive part is the computation of the quantity λ_*^k that can be equivalently computed to be the least norm solution of the smaller (in comparison to $\mathbf{A}x = b$) linear system $\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda = \mathbf{S}_k^\top(b - \mathbf{A}x^k)$. In our work we are suggesting to use an approximation $\lambda_{\approx}^k$ of the exact solution and with this way avoid executing the possibly expensive step of the update rule. Thus the inexact update is taking the following form:

$$x^{k+1} = x^k + \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_{\approx}^k = x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \underbrace{\omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\lambda_{\approx}^k - \lambda_*^k)}_{\epsilon^k}.$$

Here ϵ^k denotes a more abstract notion of inexactness and it is not necessary to be always equivalent to the quantity $\omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\lambda_{\approx}^k - \lambda_*^k)$. It can be interpreted as an expression that acts as an perturbation of the exact update. In the case that ϵ^k has the above form we say that the notion of inexactness is structured. In our work we are interested in both the *abstract* and more *structured* notions of inexactness. We first present general convergence results where we require the error ϵ^k to satisfy general assumptions (without caring how this error is generated) and later we analyze the concept of structured inexactness by presenting algorithms where $\epsilon^k = \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\lambda_{\approx}^k - \lambda_*^k)$.

In similar way, the expensive operation of SPM (1.25) is the exact computation of the projection $\Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k)$. Thus we are suggesting to replace this step with an inexact variant and compute an approximation of this projection. The inexactness here can be also interpreted using both, the abstract ϵ^k error and its more structured version $\epsilon^k = \omega\left(\Pi^{\approx}_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) - \Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k)\right)$. At this point, observe that, by using the expression (1.14) the structure of the ϵ^k in SPM and SGD has the same form.

[2]Note that SGD has identical updates to the Stochastic Newton and Stochastic proximal point method. Thus the inexactness can be added to these updates in similar way.

Exact Algorithms	Key Subproblem (problem that we solve inexactly)	Inexact Update Rules (abstract and structured inexactness error)
SGD (1.17)	Exact computation of λ_*^k, where $\lambda_*^k = \arg\min_{\lambda:\mathbf{M}_k\lambda=d_k} \|\lambda\|$. Appears in the computation of $\nabla f_{\mathbf{S}_k}(x^k) = -\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_*^k$	$x^{k+1} = x^k + \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda_{\approx}^k$ $= x^k - \omega\nabla f_{\mathbf{S}_k}(x^k) + \underbrace{\omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\lambda_{\approx}^k - \lambda_*^k)}_{\epsilon^k}.$
SPM (1.25)	Exact computation of the projection $\Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) = \arg\min_{x'\in\mathcal{L}_{\mathbf{S}_k}} \|x' - x^k\|_{\mathbf{B}}$	$x^{k+1} = \omega\Pi^{\approx}_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) + (1-\omega)x^k$ $= \omega\Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) + (1-\omega)x^k + \underbrace{\omega\left(\Pi^{\approx}_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) - \Pi^*_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k)\right)}_{\epsilon^k}$
SDSA (1.29)	Exact computation of λ_*^k, where $\lambda_*^k \in \arg\max_\lambda D(y^k + \mathbf{S}_k\lambda)$.	$y^{k+1} = y^k + \omega\mathbf{S}_k\lambda_{\approx}^k = y^k + \omega\mathbf{S}_k\lambda_*^k + \underbrace{\omega\mathbf{S}_k(\lambda_{\approx}^k - \lambda_*^k)}_{\epsilon_d^k}$

Table 3.2: The exact algorithms under study with the potentially expensive to compute key subproblems of their update rule. The inexact update rules are presented in the last column for both notions of inexactness (abstract and more structured). We use $*$ to define the important quantity that needs to be computed exactly in the update rule of each method and $\approx$ to indicate the proposed inexact variant.

In the SDSA the expensive subproblem in the update rule is the computation of the λ_*^k that satisfy $\lambda_*^k \in \arg\max_\lambda D(y^k + \mathbf{S}_k\lambda)$. Using the definition of the dual function (1.26) this value can be also computed by evaluating the least norm solution of the linear system $\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\lambda = \mathbf{S}_k^\top\left(b - \mathbf{A}(x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k\right))$. Later in Section 3.5 we analyze both notions of inexactness (abstract and more structured) for inexact variants of SDSA.

Table 3.2 presents the key sub-problem that needs to be solved in each algorithm as well as the part where the inexact error is appeared in the update rule.

3.2.2 The inexact basic method

In each iteration of the all aforementioned exact methods a sketch matrix $\mathbf{S} \sim \mathcal{D}$ is drawn from a given distribution and then a certain subproblem is solved exactly to obtain the next iterate. The sketch matrix $\mathbf{S} \in \mathbb{R}^{m\times q}$ requires to have m rows but no assumption on the number of columns is made which means that the number of columns q allows to vary through the iterations and it can be very large. The setting that we are interested in is precisely that of having such large random matrices $\mathbf{S}$. In these cases we expect that having approximate solutions of the subproblems will be beneficial.

Recently randomized iterative algorithms that requires to solve large subproblems in each iteration have been extensively studied and it was shown that are really beneficial when they compared to their single coordinates variants ($\mathbf{S} \in \mathbb{R}^{m\times 1}$) [134, 135, 166, 113]. However, in theses cases the evaluation of an exact solution for the suproblem in the update rule can be computationally very expensive. In this work we propose and analyze inexact variants by allowing to solve the subproblem that appear in the update rules of the stochastic methods, inexactly. In particular, following the convention established in [168] of naming the main algorithm of the paper *Basic method* we propose the *inexact Basic method (iBasic)* (Algorithm 4).

Algorithm 4 Inexact Basic Method (iBasic)

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.

Initialize: $x^0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**
2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$
3: Set $x^{k+1} = x^k - \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top(\mathbf{A}x^k - b) + \epsilon^k$
4: **end for**

The ϵ^k in the update rule of the method represents the abstract inexactness error described in Subsection 3.2.1. Note that, iBasic can have several equivalent interpretations. This allow as to study the methods (1.17),(1.18),(1.19) for solving the stochastic optimization problem and the sketch and project method (1.25) for the best approximation problem in a single algorithm

only. In particular iBasic can be seen as inexact stochastic gradient descent (iSGD) with fixed stepsize applied to (1.6). From (1.13), $\nabla f_{\mathbf{S}_k}(x^k) = \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_k(\mathbf{A}x^k - b)$ and as a result the update rule of iBasic can be equivalently written as: $x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k) + \epsilon^k$. In the case of the best approximation problem (1.22), iBasic can be interpreted as inexact Sketch and Project method (iSPM) as follows:

$$\begin{aligned} x^{k+1} &= x^k - \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x^k - b) + \epsilon^k \\ &= \omega \left[x^k - \mathbf{B}^{-1}(\mathbf{S}_k^\top \mathbf{A})^\top (\mathbf{S}_k^\top \mathbf{A} B^{-1} (\mathbf{S}_k^\top \mathbf{A})^\top)^\dagger (\mathbf{S}_k^\top \mathbf{A}x^k - \mathbf{S}_k^\top b) \right] + (1-\omega)x^k + \epsilon^k \\ &\overset{(1.14)}{=} \omega \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) + (1-\omega)x^k + \epsilon^k \end{aligned} \tag{3.1}$$

For the dual problem (1.26) we devote Section 3.5 for presenting an inexact variant of the SDSA (iSDSA) and analyze its convergence using the rates obtained for the iBasic in Sections 3.3 and 3.4.

3.2.3 General framework and further special cases

The proposed inexact methods, iBasic (Algorithm 4) and iSDSA (Section 3.5), belong in the general *sketch and project* framework, first proposed from Gower and Richtarik in [73] for solving consistent linear systems and where a unified analysis of several randomized methods was studied. This interpretation of the algorithms allow us to recover a comprehensive array of well-known methods as special cases by choosing carefully the combination of the main parameters of the algorithms.

In particular, the iBasic has two main parameters (besides the stepsize $\omega > 0$ of the update rule). These are the distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$ and the positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$. By choosing carefully combinations of the parameters $\mathcal{D}$ and $\mathbf{B}$ we can recover several existing popular algorithms as special cases of the general method. For example, special cases of the exact Basic method are the Randomized Kaczmarz, Randomized Gaussian Kaczmarz[3], Randomized Coordinate Descent and their block variants. For more details about the generality of the sketch and project framework and further algorithms that can be cast as special cases of the analysis we refer the interested reader to Section 3 of [73]. Here we present only the inexact update rules of two special cases that we will later use in the numerical evaluation.

Special Cases: Let us define with $\mathbf{I}_{:C}$ the column concatenation of the $m \times m$ identity matrix indexed by a random subset C of $[m]$.

- *Inexact Randomized Block Kaczmarz (iRBK)*: Let $\mathbf{B} = \mathbf{I}$ and let pick in each iteration the random matrix $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$. In this setup the update rule of the iBasic simplifies to

$$x^{k+1} = x^k - \omega \mathbf{A}_{C:}^\top (\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger (\mathbf{A}_{C:}x^k - b_C) + \epsilon^k. \tag{3.2}$$

- *Inexact Randomized Block Coordinate Descent (iRBCD)*[4]: If the matrix $\mathbf{A}$ of the linear system is positive definite then we can choose $\mathbf{B} = \mathbf{A}$. Let also pick in each iteration the random matrix $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$. In this setup the update rule of the iBasic simplifies to

$$x^{k+1} = x^k - \omega \mathbf{I}_{:C} (\mathbf{I}_{:C}^\top \mathbf{A} \mathbf{I}_{:C})^\dagger \mathbf{I}_{:C}^\top (\mathbf{A}x^k - b) + \epsilon^k. \tag{3.3}$$

For more papers related to Kaczmarz method (randomized, greedy, cyclic update rules) we refer the interested reader to [91, 114, 158, 17, 144, 159, 27, 132, 134, 49, 120, 216, 135, 175]. For the coordinate descent method (a.k.a Gauss-Seidel for linear systems) and its block variant, Randomized Block Coordinate Descent we suggest [104, 139, 166, 167, 160, 161, 163, 22, 102, 51, 2, 192].

[3]Special case of the iBasic, when the random matrix $\mathbf{S}$ is chosen to be a Gaussian vector with mean $0 \in R^m$ and a positive definite covariance matrix $\Sigma \in \mathbb{R}^{m\times m}$. That is $\mathbf{S} \sim N(0, \Sigma)$ [73].

[4]In the setting of solving linear systems Randomized Coordinate Descent is known also as Gauss-Seidel method. Its block variant can be also interpret as randomized coordinate Newton method (see [162]).

3.2.4 Other related work on inexact methods

One of the current trends in the large scale optimization problems is the introduction of inexactness in the update rules of popular deterministic and stochastic methods. The rational behind this is that an approximate/inexact step can often computed very efficiently and can have significant computational gains compare to its exact variants.

In the area of deterministic algorithms, the inexact variant of the full gradient descent method, $x^{k+1} = x^k - \omega_k[\nabla f(x^k) + \epsilon^k]$, has received a lot of attention [174, 39, 180, 59, 128]. It has been analyzed for the cases of convex and strongly convex functions under several meaningful assumptions on the inexactness error ϵ^k and its practical benefit compared to the exact gradient descent is apparent. For further deterministic inexact methods check [36] for Inexact Newton methods, [181, 171] for Inexact Proximal Point methods and [12] for Inexact Fixed point methods.

In the recent years, with the explosion that happens in areas like machine learning and data science inexactness enters also the updating rules of several stochastic optimization algorithms and many new methods have been proposed and analyzed.

In the large scale setting, stochastic optimization methods are preferred mainly because of their cheap per iteration cost (compared to their deterministic variants), their property to scale to extreme dimensions and their improved theoretical complexity bounds. In areas like machine learning and data science, where the datasets become larger rapidly, the development of faster and efficient stochastic algorithms is crucial. For this reason, inexactness has recently introduced to the update rules of several stochastic optimization algorithms and new methods have been proposed and analyzed. One of the most interesting work on inexact stochastic algorithms appears in the area of second order methods. In particular on inexact variants of the Sketch-Newton method and subsampled Newton Method for minimize convex and non-convex functions [172, 9, 14, 204, 205, 207]. Note that our results are related also with this literature since our algorithm can be seen as inexact stochastic Newton method (see equation (1.18)). To the best or our knowledge our work is the first that provide convergence analysis of inexact stochastic proximal point methods (equation (1.19)) in any setting. From numerical linear algebra viewpoint inexact sketch and project methods for solving the best approximation problem and its dual problem where also never analyzed before.

As we already mentioned our framework is quite general and many algorithms, like iRBK (3.2) and iRBCD (3.3) can be cast as special cases. As a result, our general convergence analysis includes the analysis of inexact variants of all of these more specific algorithms as special cases. In [134] an analysis of the exact randomized block Kacmzarz method has been proposed and in the experiments an inexact variant was used to speedup the method. However, no iteration complexity results were presented for the inexact variant and both the analysis and numerical evaluation have been made for linear systems with full rank matrices that come with natural partition of the rows (this is a much more restricted case than the one analyzed in our setting). For inexact variants of the randomized block coordinate descent algorithm in different settings than ours we suggest [187, 54, 20, 46].

Finally an analysis of approximate stochastic gradient descent for solving the empirical risk minimization problem using quadratic constraints and sequential semi-definite programs has been presented in [85].

3.3 Convergence Results Under General Assumptions

In this section we consider scenarios in which the inexactness error ϵ^k can be controlled, by specifying a per iteration bound σ_k on the norm of the error. In particular, by making different assumptions on the bound σ_k we derive general convergence rate results. Our focus is on the abstract notion of inexactness described in Section 3.2.1 and we make no assumptions on how this error is generated.

An important assumption that needs to be hold in all of our results is *exactness*. A formal presentation of exactness was presented in the introduction of this thesis. We highlight that is a requirement for all of the convergence results of this chapter (It is also required in the analysis of the exact algorithms; see Theorems 3 and 6 in the introduction).

3.3.1 Assumptions on inexactness error

In the convergence analysis of iBasic the following assumptions on the inexactness error are used. We note that Assumptions $1a$, $1b$ and $1c$ are special cases of Assumption 1. Moreover Assumption 2 is algorithmic dependent and can hold in addition of any of the other four assumptions. In our analysis, depending on the result we aim at, we will require either one of the first four Assumptions to hold by itself, or to hold together with Assumption 2. We will always assume exactness.

In all assumptions the expectation on the norm of error ($\|\epsilon^k\|^2$) is conditioned on the value of the current iterate x^k and the random matrix $\mathbf{S}_k$. Moreover it is worth to mention that for the convergence analysis we never assume that the inexactness error has zero mean, that is $\mathbb{E}[\epsilon^k] = 0$.

Assumption 1.

$$\mathbb{E}[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \leq \sigma_k^2, \tag{3.4}$$

where the upper bound σ_k is a sequence of random variables (that can possibly depends on both the value of the current iterate x^k and the choice of the random $\mathbf{S}_k$ at the k^{th} iteration).

The following three assumptions on the sequence of upper bounds are more restricted however as we will later see allow us to obtain stronger and more controlled results.

Assumption 1a.

$$\mathbb{E}[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \leq \sigma_k^2, \tag{3.5}$$

where the upper bound $\sigma_k \in \mathbb{R}$ is a sequence of real numbers.

Assumption 1b.

$$\mathbb{E}[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \leq \sigma_k^2 = q^2\|x^k - x^*\|_{\mathbf{B}}^2, \tag{3.6}$$

where the upper bound is a special sequence that depends on a non-negative inexactness parameter q and the distance to the optimal value $\|x^k - x^*\|_{\mathbf{B}}^2$.

Assumption 1c.

$$\mathbb{E}[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \leq \sigma_k^2 = 2q^2 f_{\mathbf{S}_k}(x^k), \tag{3.7}$$

where the upper bound is a special sequence that depends on a non-negative inexactness parameter q and the value of the stochastic function $f_{\mathbf{S}_k}$ computed at the iterate x^k. Recall, that in our setting $f_{\mathbf{S}}(x) = \frac{1}{2}\|\nabla f_{\mathbf{S}}(x)\|_{\mathbf{B}}^2$ (1.38). Hence, the upper bound can be equivalently expressed as $\sigma_k^2 = q^2\|\nabla f_{\mathbf{S}}(x)\|_{\mathbf{B}}^2$.

Finally the next assumption is more algorithmic oriented. It holds in cases where the inexactness error ϵ^k in the update rule is chosen to be orthogonal with respect to the $\mathbf{B}$-inner product to the vector $\Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k) - x^* = (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)$. This statement may seem odd at this point but its usefulness will become more apparent in the next section where inexact algorithms with structured inexactness error will be analyzed. As it turns out, in the case of structured inexactness error (Algorithm 5) this assumption is satisfied.

Assumption 2.

$$\mathbb{E}[\left\langle (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*), \epsilon^k \right\rangle_{\mathbf{B}}] = 0. \tag{3.8}$$

3.3.2 Convergence results

In this section we present the analysis of the convergence rates of iBasic by assuming several combination of the previous presented assumptions.

All convergence results are described only in terms of convergence of the iterates x^k, that is $\|x^k - x^*\|_{\mathbf{B}}^2$, and not the objective function values $f(x^k)$. This is sufficient, because by $f(x) \leq \frac{\lambda_{\max}}{2}\|x - x^*\|_{\mathbf{B}}^2$ (see Lemma 1) we can directly deduce a convergence rate for the function values.

The exact Basic method (Algorithm 4 with $\epsilon^k = 0$), has been analyzed in [168] and it was shown to converge with $\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \rho^k\|x^0 - x^*\|_{\mathbf{B}}^2$ where $\rho = 1 - \omega(2-\omega)\lambda_{\min}^+$. Our analysis of iBasic is more general and includes the convergence of the exact Basic method as

special case when we assume that the upper bound is $\sigma_k = 0, \quad \forall k \geq 0$. For brevity, in he convergence analysis results of this chapter we also use

$$\rho = 1 - \omega(2-\omega)\lambda_{\min}^{+}.$$

Let us start by presenting the convergence of iBasic when only Assumption 1a holds for the inexactness error.

Theorem 20. *Let assume exactness and let $\{x^k\}_{k=0}^{\infty}$ be the iterates produced by iBasic with $\omega \in (0,2)$. Set $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ and consider the error ϵ^k be such that it satisfies Assumption 1a. Then,*

$$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}] \leq \rho^{k/2}\|x^0 - x^*\|_{\mathbf{B}} + \sum_{i=0}^{k-1} \rho^{\frac{k-1-i}{2}}\sigma_i. \tag{3.9}$$

Proof. See Section 3.8.1. □

Corollary 21. *In the special case that the upper bound σ_k in Assumption 1a is fixed, that is $\sigma_k = \sigma$ for all $k > 0$ then inequality (3.9) of Theorem 20 takes the following form:*

$$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}] \leq \rho^{k/2}\|x^0 - x^*\|_{\mathbf{B}} + \sigma\frac{\rho^{1/2}}{1-\rho}. \tag{3.10}$$

This means that we obtain a linear convergence rate up to a solution level that is proportional to the upper bound σ[5]*.*

Proof. See Section 3.8.2. □

Inspired from [59], let us now analyze iBasic using the sequence of upper bounds that described in Assumption 1b. This construction of the upper bounds allows us to obtain stronger and more controlled results. In particular using the upper bound of Assumption 1b the sequence of expected errors converge linearly to the exact x^* (not in a potential neighborhood like the previous result). In addition Assumption 1b guarantees that the distance to the optimal solution reduces with the increasing of the number of iterations. However for this stronger convergence a bound for $\lambda_{\min}^{+}$ is required, a quantity that in many problems is unknown to the user or intractable to compute. Nevertheless, there are cases that this value has a closed form expression and can be computed before hand without any further cost. See for example [113, 116, 112, 80] where methods for solving the average consensus were presented and the value of $\lambda_{\min}^{+}$ corresponds to the algebraic connectivity of the network under study.

Theorem 22. *Assume exactness. Let $\{x^k\}_{k=0}^{\infty}$ be the iterates produced by iBasic with $\omega \in (0,2)$. Set $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ and consider the inexactness error ϵ^k be such that it satisfies Assumption 1b, with $0 \leq q < 1 - \sqrt{\rho}$. Then*

$$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \left(\sqrt{\rho} + q\right)^{2k}\|x^0 - x^*\|_{\mathbf{B}}^2. \tag{3.11}$$

Proof. See Section 3.8.3. □

At Theorem 22, to guarantee linear convergence the *inexact parameter* q should live in the interval $\left[0, 1-\sqrt{\rho}\right)$. In particular, q is the parameter that controls the level of inexactness of Algorithm 4. Not surprisingly the fastest convergence rate is obtained when $q = 0$; in such case the method becomes equivalent with its exact variant and the convergence rate simplifies to $\rho = 1 - \omega(2-\omega)\lambda_{\min}^{+}$. Note also that similar to the exact case the optimal convergence rate is obtained for $\omega = 1$ [168].

Moreover, the upper bound σ_k of Assumption 1b depends on two important quantities, the $\lambda_{\min}^{+}$ (through the upper bound of the inexactness parameter q) and the distance to the optimal

[5]Several similar more specific assumptions can be made for the upper bound σ_k. For example if the upper bound satisfies $\sigma_k = \sigma^k$ with $\sigma \in (0,1)$ for all $k > 0$ then it can be shown that $C \in (0,1)$ exist such that inequality (3.9) of Theorem 20 takes the form: $\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}] \leq \mathcal{O}(C^k)$ (see [180, 59] for similar results).

solution $\|x^k - x^*\|^2_{\mathbf{B}}$. Thus, it can have natural interpretation. In particular the inexactness error is allowed to be large either when the current iterate is far from the optimal solution ($\|x^k - x^*\|^2_{\mathbf{B}}$ large) or when the problem is well conditioned and $\lambda^+_{\min}$ is large. In the opposite scenario, when we have ill conditioned problem or we are already close enough to the optimum x^* we should be more careful and allow less errors to the updates of the method.

In the next theorem we provide the complexity results of iBasic in the case that the Assumption 2 is satisfied combined with one of the previous assumptions.

Theorem 23. *Let assume exactness and let $\{x^k\}_{k=0}^{\infty}$ be the iterates produced by iBasic with $\omega \in (0,2)$. Set $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Let also assume that the inexactness error ϵ^k be such that it satisfies Assumption 2. Then:*

(i) If Assumption 1 holds:

$$\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}] \leq \rho^k \|x^0 - x^*\|^2_{\mathbf{B}} + \sum_{i=0}^{k-1} \rho^{k-1-i} \bar{\sigma}_i^2, \tag{3.12}$$

where $\bar{\sigma}_i^2 = \mathbb{E}[\sigma_i^2], \forall i \in [k-1]$.

(ii) If Assumption 1b holds with $q \in \left(0, \sqrt{\rho}\right)$:

$$\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}] \quad \leq \quad (\rho + q^2)^k \|x^0 - x^*\|^2_{\mathbf{B}}. \tag{3.13}$$

(iii) If Assumption 1c holds with $q \in \left(0, \sqrt{\omega(2-\omega)}\right)$:

$$\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}] \leq (1 - (\omega(2-\omega) - q^2)\lambda^+_{\min})^k \|x^0 - x^*\|^2_{\mathbf{B}} = (\rho + q^2\lambda^+_{\min})^k \|x^0 - x^*\|^2_{\mathbf{B}}. \tag{3.14}$$

Proof. See Section 3.8.4. □

Remark 4. *In the case that Assumptions 1a and 2 hold simultaneously, the convergence of iBasic is similar to* (3.12) *but in this case $\bar{\sigma}_i^2 = \sigma_i^2$, $\forall i \in [k-1]$ (due to Assumption 1a, $\sigma_k \in \mathbb{R}$ is a sequence of real numbers). In addition, note that for $q \in (0, \min\{\sqrt{\rho}, 1 - \sqrt{\rho}\})$ having Assumption 2 on top of Assumption 1b leads to improvement of the convergence rate. In particular, from Theorem 22, iBasic converges with rate $(\sqrt{\rho} + q)^2 = \rho + q^2 + 2\sqrt{\rho}q$ while having both assumptions this is simplified to the faster $\rho + q^2$* (3.13).

3.4 iBasic with Structured Inexactness Error

Up to this point, the analysis of iBasic was focused in more general abstract cases where the inexactness error ϵ^k of the update rule satisfies several general assumptions. In this section we are focusing on a more structured form of inexactness error and we provide convergence analysis in the case that a linearly convergent algorithm is used for the computation of the expensive key subproblem of the method.

3.4.1 Linear system in the update rule

As we already mentioned in Section 3.2.1 the update rule of the exact Basic method (Algorithm 4 with $\epsilon^k = 0$) can be expressed as $x^{k+1} = x^k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_*^k$, where $\lambda_*^k = (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (b - \mathbf{A}x^k)$.

Using this expression the exact Basic method can be equivalently interpreted as the following two step procedure:

1. Find the least norm solution[6] of $\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k} \lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}x^k)}_{d_k}$. That is find $\lambda_*^k = \arg\min_{\lambda \in \mathcal{Q}_k} \|\lambda\|$ where $\mathcal{Q}_k = \{\lambda \in \mathbb{R}^q : \mathbf{M}_k \lambda = d_k\}$.

[6] We are precisely looking for the least norm solution of the linear system $\mathbf{M}_k \lambda = d_k$ because this solution can be written down in a compact way using the Moore-Penrose pseudoinverse. This is equivalent with the

2. Compute the next iterate: $x^{k+1} = x^k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_*^k$.

In the case that the random matrix $\mathbf{S}_k$ is large (this is the case that we are interested in), solving exactly the linear system $\mathbf{M}_k\lambda = d_k$ in each step can be prohibitively expensive. To reduce this cost we allow the inner linear system $\mathbf{M}_k\lambda = d_k$ to be solved inexactly using an iterative method. In particular we propose and analyze the following inexact algorithm:

Algorithm 5 iBasic with structured inexactness error

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.

Initialize: $x^0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**

2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$

3: Using an iterative method compute an approximation $\lambda_{\approx}^k$ of the least norm solution of the linear system:

$$\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k}\lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}x^k)}_{d_k}. \tag{3.15}$$

4: Set $x^{k+1} = x^k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_{\approx}^k$.

5: **end for**

For the computation of the inexact solution of the linear system (3.15) any known iterative method for solving general linear systems can be used. In our analysis we focus on linearly convergent methods. For example based on the properties of the linear system (3.15), conjugate gradient (CG) or sketch and project method (SPM) can be used for the execution of step 3. In these cases, we name Algorithm 5, *InexactCG* and *InexactSP* respectively.

It is known that the classical CG can solve linear systems with positive definite matrices. In our approach matrix $\mathbf{M}_k$ is positive definite only when the original linear system $\mathbf{A}x = b$ has full rank matrix $\mathbf{A}$. On the other side SPM can solve any consistent linear system and as a result can solve the inner linear system $\mathbf{M}_k\lambda^k = d_k$ without any further assumption on the original linear system. In this case, one should be careful because the system has no unique solution. We are interested to find the least norm solution of $\mathbf{M}_k\lambda^k = d_k$ which means that the starting point of the sketch and project at the k^{th} iteration should be always $\lambda_0^k = 0$. Recall that any special case of the sketch and project method (Section 3.2.3) solves the best approximation problem.

Let us now define λ_r^k to be the approximate solution $\lambda_{\approx}^k$ of the $q \times q$ linear system (3.15) obtained after r steps of the linearly convergent iterative method. Using this, the update rule of Algorithm 5, takes the form:

$$x^{k+1} = x^k + \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_r^k. \tag{3.16}$$

Remark 5. *The update rule* (3.16) *of Algorithm 5 is equivalent to the update rule of iBasic (Algorithm 4) when the error ϵ^k is chosen to be,*

$$\epsilon^k = \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_r^k - \lambda_*^k). \tag{3.17}$$

This is precisely the connection between the abstract and more concrete/structured notion of inexactness that first presented in Table 3.2.

Let us now define a Lemma that is useful for the analysis of this section and it verifies that Algorithm 5 with unit stepsize satisfies the general Assumption 2 presented in Section 3.3.1.

Lemma 24. *Let us denote $x_*^k = \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k)$ the projection of x^k onto $\mathcal{L}_{\mathbf{S}_k}$ in the $\mathbf{B}$-norm and $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Let also assume that $\omega = 1$ (unit stepsize). Then for the updates of Algorithm 5*

expression that appears in our update: $\lambda_*^k = (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (b - \mathbf{A}x^k) = \mathbf{M}_k^\dagger d_k$. However it can be easily shown that the method will still converge with the same rate of convergence even if we choose any other solution of the linear system $\mathbf{M}_k\lambda = d_k$.

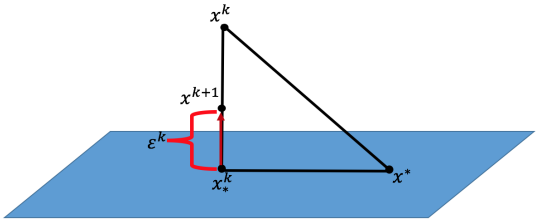


Figure 3.1: Graphical interpretation of orthogonality (justifies equation (3.18)). It shows that the two vectors, $x^k_* - x^*$ and ϵ^k, are orthogonal complements of each other with respect to the $\mathbf{B}$-inner product. x^{k+1} is the point that Algorithm 5 computes in each step. The colored region represents the $Null(\mathbf{S}_k^\top \mathbf{A})$. $x^k_* = \Pi_{\mathcal{L}_{\mathbf{S}_k},\mathbf{B}}(x^k)$, $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ and ϵ^k is the inexactness error.

it holds that:

$$\left\langle x^k_* - x^*, \epsilon^k \right\rangle_{\mathbf{B}} = \left\langle (\mathbf{I} - \omega \mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*), \epsilon^k \right\rangle_{\mathbf{B}} = 0, \quad \forall k \geq 0. \tag{3.18}$$

Proof. Note that $x^k_* - x^* = x^k - \nabla f_{\mathbf{S}_k}(x^k) - x^* \in Null(\mathbf{S}_k^\top \mathbf{A})$. Moreover $\epsilon^k \overset{(3.17)}{=} \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda^k_r - \lambda^k_*) \in Range(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)$. From the knowledge that the null space of an arbitrary matrix is the orthogonal complement of the range space of its transpose we have that $Null(\mathbf{S}_k^\top \mathbf{A})$ is orthogonal with respect to the $\mathbf{B}$-inner product to $Range(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)$. This completes the proof (see Figure 3.1 for the graphical interpretation). □

3.4.2 Sketch and project interpretation

Let us now give a different interpretation of the inexact update rule of Algorithm 5 using the sketch and project approach. That will make us appreciate more the importance of the dual viewpoint and make clear the connection between the primal and dual methods.

In general, execute a projection step is one of the most common task in numerical linear algebra/optimization literature. However in the large scale setting even this task can be prohibitively expensive and it can be difficult to execute inexactly. For this reason we suggest to move to the dual space where the inexactness can be easily controlled.

Observe that the update rule of the exact sketch and project method (1.23) has the same structure as the best approximation problem (1.22) where the linear system under study is the sketched system $\mathbf{S}_k^\top \mathbf{A}x = \mathbf{S}_k^\top b$ and the starting point is the current iterate x^k. Hence we can easily compute its dual:

$$\max_{\lambda \in \mathbb{R}^q} D_k(\lambda) := (\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A}x^k)^\top \lambda - \frac{1}{2}\|\mathbf{A}^\top \mathbf{S}_k \lambda\|^2_{\mathbf{B}^{-1}}. \tag{3.19}$$

where $\lambda \in \mathbb{R}^q$ is the dual variable. The λ^k_* (possibly more than one) that solves the dual problem in each iteration k, is the one that satisfies $\nabla D_k(\lambda^k_*) = 0$. By computing the derivative this is equivalent with finding the λ that satisfies the linear system $\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda = \mathbf{S}_k^\top (b - \mathbf{A}x^k)$. This is the same linear system we desire to solve inexactly in Algorithm 5. Thus, computing an inexact solution $\lambda^k_{\approx}$ of the linear system is equivalent with computing an inexact solution of the dual problem (3.19). Then by using the affine mapping (1.27) that connects the primal and the dual spaces we can also evaluate an inexact solution of the original primal problem (1.23).

The following result relates the inexact levels of these quantities. In particular it shows that dual suboptimality of λ^k in terms of dual function values is equal to the distance of the dual values λ^k in the $\mathbf{M}_k$-norm.

Lemma 25. *Let us define $\lambda^k_* \in \mathbb{R}^q$ be the exact solution of the linear system $\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda = \mathbf{S}_k^\top (b - \mathbf{A}x^k)$ or equivalently of dual problem (3.19). Let us also denote with $\lambda^k_{\approx} \in \mathbb{R}^q$ the inexact solution. Then:*

$$D_k(\lambda^k_*) - D_k(\lambda^k_{\approx}) = \frac{1}{2}\|\lambda^k_{\approx} - \lambda^k_*\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}.$$

Proof.

$$\begin{aligned} D_k(\lambda_*^k) - D_k(\lambda_{\approx}^k) &\overset{(3.19)}{=} [\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A} x^k]^\top [\lambda_*^k - \lambda_{\approx}^k] - \frac{1}{2}(\lambda_*^k)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_*^k \\ &\quad + \frac{1}{2}(\lambda_{\approx}^k)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_{\approx}^k \\ &\overset{(1.28)}{=} (\lambda_*^k)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k [\lambda_*^k - \lambda_{\approx}^k] - \frac{1}{2}(\lambda_*^k)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_*^k \\ &\quad + \frac{1}{2}(\lambda_{\approx}^k)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k \lambda_{\approx}^k \\ &= \frac{1}{2}(\lambda_{\approx}^k - \lambda_*^k)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\lambda_{\approx}^k - \lambda_*^k) \\ &= \frac{1}{2}\|\lambda_{\approx}^k - \lambda_*^k\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k} \end{aligned}$$

where in the second equality we use equation (1.28) to connect the optimal solutions of (1.23) and (3.19) and obtain $[\mathbf{S}_k^\top b - \mathbf{S}_k^\top \mathbf{A} x^k]^\top = (\lambda_*^k)^\top \mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k$. □

3.4.3 Complexity results

In this part we analyze the performance of Algorithm 5 when a linearly convergent iterative method is used for solving inexactly the linear system (3.15) in step 3 of Algorithm 5 . We denote with λ_r^k the approximate solution of the linear system after we run the iterative method for r steps.

Before state the main convergence result let us present a lemma that summarize some observations that are true in our setting.

Lemma 26. *Let $\lambda_*^k = (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (b - \mathbf{A}x^k)$ be the exact solution and λ_r^k be approximate solution of the linear system* (3.15)*. Then, $\|\lambda_*^k\|^2_{\mathbf{M}_k} = 2 f_{\mathbf{S}_k}(x^k)$ and $\|\epsilon^k\|^2_{\mathbf{B}} = \|\lambda_r^k - \lambda_*^k\|^2_{\mathbf{M}_k}$.*

Proof.

$$\begin{aligned} \|\lambda_*^k\|^2_{\mathbf{M}_k} &= \|\mathbf{M}_k^\dagger \mathbf{S}_k^\top \mathbf{A}(x^* - x^k)\|^2_{\mathbf{M}_k} = (x^k - x^*)^\top \mathbf{A}^\top \mathbf{S}_k \underbrace{\mathbf{M}_k^\dagger \mathbf{M}_k \mathbf{M}_k^\dagger}_{\mathbf{M}_k^\dagger} \mathbf{S}_k^\top \mathbf{A}(x^k - x^*) \\ &\overset{(1.11)}{=} (x^k - x^*)^\top \mathbf{Z}_k (x^k - x^*) \overset{(1.12)}{=} 2 f_{\mathbf{S}_k}(x^k). \end{aligned} \tag{3.20}$$

Moreover,

$$\|\epsilon^k\|^2_{\mathbf{B}} \overset{Remark\ 5}{=} \|\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_r^k - \lambda_*^k)\|^2_{\mathbf{B}} = \|\lambda_r^k - \lambda_*^k\|^2_{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k} = \|\lambda_r^k - \lambda_*^k\|^2_{\mathbf{M}_k} \tag{3.21}$$

□

Theorem 27. *Let us assume that for the computation of the inexact solution of the linear system* (3.15) *in step 3 of Algorithm 5, a linearly convergent iterative method is chosen such that* [7]*:*

$$\mathbb{E}[\|\lambda_r^k - \lambda_*^k\|^2_{\mathbf{M}_k} \mid x^k, \mathbf{S}_k] \leq \rho^r_{\mathbf{S}_k} \|\lambda_0^k - \lambda_*^k\|^2_{\mathbf{M}_k}, \tag{3.22}$$

where $\lambda_0^k = 0$ for any $k > 0$ and $\rho_{\mathbf{S}_k} \in (0,1)$ for every choice of $\mathbf{S}_k \sim \mathcal{D}$. Let exactness hold and let $\{x^k\}_{k=0}^\infty$ be the iterates produced by Algorithm 5 with unit stepsize ($\omega = 1$). Set $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Suppose further that there exists a scalar $\theta < 1$ such that with probability 1, $\rho_{\mathbf{S}_k} \leq \theta$. Then, Algorithm 5 converges linearly with:

$$\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}] \leq \left[1 - (1 - \theta^r)\,\lambda_{\min}^+\right]^k \|x^0 - x^*\|^2_{\mathbf{B}}.$$

[7] In the case that deterministic iterative method is used, like CG, we have that $\|\lambda_r^k - \lambda_*^k\|^2_{\mathbf{M}_k} \leq \rho^r_{\mathbf{S}_k} \|\lambda_0^k - \lambda_*^k\|^2_{\mathbf{M}_k}$ which is also true in expectation.

Proof. Theorem 27 can be interpreted as corollary of the general Theorem 23(iii). Thus, it is sufficient to show that Algorithm 5 satisfies the two Assumptions 1c and 2. Firstly, note that from Lemma 24, Assumption 2 is true. Moreover,

$$\begin{aligned}\mathbb{E}[\|\epsilon^k\|^2_{\mathbf{M}_k} \mid x^k, \mathbf{S}_k] &\overset{(3.21)}{=} \mathbb{E}[\|\lambda_r^k - \lambda_*^k\|^2_{\mathbf{M}_k} \mid x^k, \mathbf{S}_k] \overset{(3.22)}{\leq} \rho^r_{\mathbf{S}_k}\|\lambda_0^k - \lambda_*^k\|^2_{\mathbf{M}_k} \\ &\leq \theta^r\|\lambda_0^k - \lambda_*^k\|^2_{\mathbf{M}_k} \overset{\lambda_0^k=0}{=} \theta^r\|\lambda_*^k\|^2_{\mathbf{M}_k} \overset{(3.20)}{=} 2\theta^r f_{\mathbf{S}_k}(x^k)\end{aligned}$$

which means that Assumption 1c also holds with $q = \theta^{r/2} \in (0,1)$. This completes the proof. □

Having present the main result of this section let us now state some remarks that will help understand the convergence rate of the last Theorem.

Remark 6. *From its definition $\theta^r \in (0,1)$ and as a result $(1-\theta^r)\,\lambda^+_{\min} \leq \lambda^+_{\min}$. This means that the method converges linearly but always with worst rate than its exact variant.*

Remark 7. *Let as assume that θ is fixed. Then as the number of iterations in step 3 of the algorithm ($r \to \infty$) increasing $(1-\theta^r) \to 1$ and as a result the method behaves similar to the exact case.*

Remark 8. *The $\lambda^+_{\min}$ depends only on the random matrices $\mathbf{S} \sim \mathcal{D}$ and to the positive definite matrix $\mathbf{B}$ and is independent to the iterative process used in step 3. The iterative process of step 3 controls only the parameter θ of the convergence rate.*

Remark 9. *Let us assume that we run Algorithm 5 two separate times for two different choices of the linearly convergence iterative method of step 3. Let also assume that the distribution $\mathcal{D}$ of the random matrices and the positive definite matrix $\mathbf{B}$ are the same for both instances and that for step 3 the iterative method run for r steps for both algorithms. Let assume that $\theta_1 < \theta_2$ then we have that $\rho_1 = 1-(1-\theta_1^r)\,\lambda^+_{\min} < 1-(1-\theta_2^r)\,\lambda^+_{\min} = \rho_2$. This means in the case that θ is easily computable, we should always prefer the inexact method with smaller θ.*

The convergence of Theorem 27 is quite general and it holds for any linearly convergent methods that can inexactly solve (3.15). However, in case that the iterative method is known we can have more concrete results. See below the more specified results for the cases of Conjugate gradient (CG) and Sketch and project method (SPM).

Convergence of InexactCG: CG is deterministic iterative method for solving linear systems $\mathbf{A}x = b$ with symmetric and positive definite matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ in finite number of iterations. In particular, it can be shown that converges to the unique solution in at most n steps. The worst case behavior of CG is given by [198, 67] [8]:

$$\|x^k - x^*\|_{\mathbf{A}} \leq \left(\frac{\sqrt{\kappa(\mathbf{A})}-1}{\sqrt{\kappa(\mathbf{A})}+1}\right)^{2k}\|x^0 - x^*\|_{\mathbf{A}}, \tag{3.23}$$

where x^k is the k^{th} iteration of the method and $\kappa(\mathbf{A})$ the condition number of matrix $\mathbf{A}$.

Having present the convergence of CG for general linear systems, let us now return back to our setting. We denote $\lambda_r^k \in \mathbb{R}^q$ to be the approximate solution of the inner linear system (3.15) after r conjugate gradient steps. Thus using (3.23) we know that $\|\lambda_r^k - \lambda_*^k\|^2_{\mathbf{M}_k} \leq \rho^{4r}_{\mathbf{S}_k}\|\lambda_0^k - \lambda_*^k\|^2_{\mathbf{M}_k}$, where $\rho_{\mathbf{S}_k} = \left(\frac{\sqrt{\kappa(\mathbf{M}_k)}-1}{\sqrt{\kappa(\mathbf{M}_k)}+1}\right)$. Now by making the same assumption as the general Theorem 27 the InexactCG converges with $\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}] \leq \left[1-(1-\theta^r_{CG})\,\lambda^+_{\min}\right]^k\|x^0 - x^*\|^2_{\mathbf{B}}$, where $\theta_{CG} < 1$ such that $\rho_{\mathbf{S}_k} = \left(\frac{\sqrt{\kappa(\mathbf{M}_k)}-1}{\sqrt{\kappa(\mathbf{M}_k)}+1}\right)^4 \leq \theta_{CG}$ with probability 1.

[8] A sharper convergence rate of CG [198] for solving $\mathbf{A}x = b$ can be also used

$$\|x^k - x^*\|^2_{\mathbf{A}} \leq \left(\frac{\lambda_{n-k}-\lambda_1}{\lambda_{n-k}+\lambda_1}\right)^2\|x^0 - x^*\|^2_{\mathbf{A}},$$

where matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ has $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ eigenvalues.

Convergence of InexactSP: In this setting we suggest to run the sketch and project method (SPM) for solving inexactly the linear system (3.15). This allow us to have no assumptions on the structure of the original system $\mathbf{A}x = b$ and as a result we are able to solve more general problems compared to what problems InexactCG can solve[9]. Like before, by making the same assumptions as in Theorem 27 the more specific convergence $\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \left[1 - (1 - \theta_{SP}^r)\,\lambda_{\min}^+\right]^k \|x^0 - x^*\|_{\mathbf{B}}^2$, for the InexactSP can be obtained. Now the quantity $\rho_{\mathbf{S}_k}$ denotes the convergence rate of the exact Basic method[10] when this applied to solve linear system (3.15) and $\theta_{SP} < 1$ is a scalar such that $\rho_{\mathbf{S}_k} \leq \theta_{SP}$ with probability 1.

3.5 Inexact Dual Method

In the previous sections we focused on the analysis of inexact stochastic methods for solving the stochastic optimization problem (1.6) and the best approximation (1.22). In this section we turn into the dual of the best approximation (1.26) and we propose and analyze an inexact variant of the SDSA (1.29). We call the new method iSDSA and is formalized as Algorithm 6. In the update rule ϵ_d^k indicates the dual inexactness error that appears in the k^{th} iteration of iSDSA.

Algorithm 6 Inexact Stochastic Dual Subspace Ascent (iSDSA)

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.

Initialize: $y^0 = 0 \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**

2: Draw a fresh sample $\mathbf{S}_k \sim \mathcal{D}$

3: Set $y^{k+1} = y^k + \omega \mathbf{S}_k \left(\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k\right)^\dagger \mathbf{S}_k^\top \left(b - \mathbf{A}(x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k)\right) + \epsilon_d^k$

4: **end for**

3.5.1 Correspondence between the primal and dual methods

With the sequence of the dual iterates $\{y^k\}_{k=0}^\infty$ produced by the iSDSA we can associate a sequence of primal iterates $\{x^k\}_{k=0}^\infty$ using the affine mapping (1.31). In our first result we show that the random iterates produced by iBasic arise as an affine image of iSDSA under this affine mapping.

Theorem 28. *(Correspondence between the primal and dual methods) Let $\{x^k\}_{k=0}^\infty$ be the iterates produced by iBasic (Algorithm 4). Let $y^0 = 0$, and $\{y^k\}_{k=0}^\infty$ the iterates of the iSDSA. Assume that the two methods use the same stepsize $\omega > 0$ and the same sequence of random matrices $\mathbf{S}_k$. Assume also that $\epsilon^k = \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_d^k$ where ϵ^k and ϵ_d^k are the inexactness errors appear in the update rules of iBasic and iSDSA respectively. Then*

$$x^k = \phi(y^k) = x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k.$$

for all $k \geq 0$. That is, the primal iterates arise as affine images of the dual iterates.

Proof.

$$\begin{aligned}
\phi(y^{k+1}) &\overset{(1.31)}{=} x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^{k+1} \overset{(1.30),\text{Alg.6}}{=} x^0 + \mathbf{B}^{-1}\mathbf{A}^\top \left[y^k + \omega \mathbf{S}_k \lambda^k + \epsilon_d^k\right] \\
&\overset{(1.11),(1.30)}{=} \underbrace{x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k}_{\phi(y^k)} + \omega \mathbf{B}^{-1}\mathbf{Z}_k \left(x^* - (\underbrace{x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k}_{\phi(y^k)})\right) + \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_d^k \\
&= \phi(y^k) - \omega \mathbf{B}^{-1}\mathbf{Z}_k(\phi(y^k) - x^*) + \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_d^k
\end{aligned}$$

[9] Recall that InexactCG requires the matrix $\mathbf{M}_k$ to be positive definite (this is true when matrix $\mathbf{A}$ is a full rank matrix)

[10] Recall that iBasic and its exact variant ($\epsilon^k = 0$) can be expressed as sketch and project methods (3.1).

Thus by choosing the inexactness error of the primal method to be $\epsilon^k = \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_d^k$ the sequence of vectors $\{\phi(y^k)\}$ satisfies the same recursion as the sequence $\{x^k\}$ defined by iBasic. It remains to check that the first element of both recursions coincide. Indeed, since $y^0 = 0$, we have $x^0 = \phi(0) = \phi(y^0)$. □

3.5.2 iSDSA with structured inexactness error

In this subsection we present Algorithm 7. It can be seen as a special case of iSDSA but with a more structured inexactness error.

Algorithm 7 iSDSA with structured inexactness error

Input: Distribution $\mathcal{D}$ from which we draw random matrices $\mathbf{S}$, positive definite matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$, stepsize $\omega > 0$.

Initialize: $y^0 = 0 \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$

1: **for** $k = 0, 1, 2, \cdots$ **do**

2: Generate a fresh sample $\mathbf{S}_k \sim \mathcal{D}$

3: Using an Iterative method compute an approximation $\lambda_{\approx}^k$ of the least norm solution of the linear system:

$$\underbrace{\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k}_{\mathbf{M}_k} \lambda = \underbrace{\mathbf{S}_k^\top (b - \mathbf{A}(x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k)}_{d_k} \tag{3.24}$$

4: Set $y^{k+1} = y^k + \omega \mathbf{S}_k \lambda_{\approx}^k$

5: **end for**

Similar to their primal variants, it can be easily checked that Algorithm 7 is a special case of the iSDSA (Algorithm 6) when the dual inexactness error is chosen to be $\epsilon_d^k = \mathbf{S}_k(\lambda_r^k - \lambda_*^k)$. Note that, using the observation of Remark 5 that $\epsilon^k = \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k(\lambda_r^k - \lambda_*^k)$ and the above expression of ϵ_d^k we can easily verify that the expression $\epsilon^k = \mathbf{B}^{-1}\mathbf{A}^\top \epsilon_d^k$ holds. This is precisely the connection between the primal and dual inexactness errors that have already been used in the proof of Theorem 28.

3.5.3 Convergence of dual function values

We are now ready to state a linear convergence result describing the behavior of the inexact dual method in terms of the function values $D(y^k)$. The following result is focused on the convergence of iSDSA by making similar assumption to Assumption 1*b*. Similar convergence results can be obtained using any other assumption of Section 3.3.1. The convergence of Algorithm 7, can be also easily derived using similar arguments with the one presented in Section 3.4 and the convergence guarantees of Theorem 27.

Theorem 29. *(Convergence of dual objective). Assume exactness. Let $y^0 = 0$ and let $\{y^k\}_{k=0}^\infty$ to be the dual iterates of iSDSA (Algorithm 6) with $\omega \in (0, 2)$. Set $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$ and let y^* be any dual optimal solution. Consider the inexactness error ϵ_d^k be such that it satisfies $\mathbb{E}[\|\mathbf{B}^{-1}\mathbf{A}^\top \epsilon_d^k\|_{\mathbf{B}}^2 \mid y^k, \mathbf{S}_k] \le \sigma_k^2 = q^2 2\left[D(y^*) - D(y^k)\right]$ where $0 \le q < 1 - \sqrt{\rho}$. Then*

$$\mathbb{E}[D(y^*) - D(y^k)] \le (\sqrt{\rho} + q)^{2k}\left[D(y^*) - D(y^0)\right]. \tag{3.25}$$

Proof. The proof follows by applying Theorem 22 together with Theorem 28 and the identity $\frac{1}{2}\|x^k - x^*\|_{\mathbf{B}}^2 = D(y^*) - D(y^k)$ (1.32). □

Note that in the case that $q = 0$, iSDSA simplifies to its exact variant SDSA and the convergence rate coincide with the one presented in Theorem 6. Following similar arguments to those in [74], the same rate can be proved for the duality gap $\mathbb{E}[P(x^k) - D(y^k)]$.

3.6 Numerical Evaluation

In this section we perform preliminary numerical tests for studying the computational behavior of iBasic with structured inexactness error when is used to solve the best approximation problem (1.22) or equivalently the stochastic optimization problem (1.6)[11]. As we have already mentioned, iBasic can be interpreted as sketch-and-project method, and as a result a comprehensive array of well-known algorithms can be recovered as special cases by varying the main parameters of the methods (Section 3.2.3). In particular, in our experiments we focus on the evaluation of two popular special cases, the inexact Randomized Block Kaczmarz (iRBK) (equation (3.2)) and inexact randomized block coordinate descent method (iRBCD) (equation (3.3))We implement Algorithm 5 presented in Section 3.4 using CG [12] to inexactly solve the linear system of the update rule (equation (3.15)). Recall that in this case we named the method InexactCG.

The convergence analysis of previous sections is quite general and holds for several combinations of the two main parameters of the method, the positive definite matrix $\mathbf{B}$ and the distribution $\mathcal{D}$ of the random matrices $\mathbf{S}$. For obtaining iRBK as special case we have to choose $\mathbf{B} = \mathbf{I} \in \mathbb{R}^{n\times n}$ (Identity matrix) and for the iRBCD the given matrix $\mathbf{A}$ should be positive definite and choose $\mathbf{B} = \mathbf{A}$. For both methods the distribution $\mathcal{D}$ should be over random matrices $\mathbf{S} = \mathbf{I}_{:C}$ where $\mathbf{I}_{:C}$ is the column concatenation of the $m \times m$ identity matrix indexed by a random subset C of $[m]$. In our experiments we choose to have one specific distribution over these matrices. In particular, we assume that the random matrix in each iteration is chosen uniformly at random to be $\mathbf{S} = \mathbf{I}_{:d}$ with the subset d of $[m]$ to have fixed pre-specified cardinality.

The code for all experiments is written in the Julia 0.6.3 programming language and run on a Mac laptop computer (OS X El Capitan), 2.7 GHz Intel Core i5 with 8 GB of RAM.

To coincide with the theoretical convergence results of Algorithm 5 the relaxation parameter (stepsize) of the methods study in our experiments is chosen to be $\omega = 1$ (no relaxation). In all implementations, we use $x^0 = 0 \in \mathbb{R}^n$ as an initial point and in comparing the methods with their inexact variants we use the relative error measure $\|x^k - x^*\|^2_{\mathbf{B}}/\|x^0 - x^*\|^2_{\mathbf{B}} \overset{x^0=0}{=} \|x^k - x^*\|^2_{\mathbf{B}}/\|x^*\|^2_{\mathbf{B}}$. We run each method (exact and inexact) until the relative error is below 10^{-5}. For the horizontal axis we use either the number of iterations or the wall-clock time measured using the tic-toc Julia function. In the exact variants, the linear system (3.15) in Algorithm 5 needs to be solved exactly. In our experiments we follow the implementation of [73] for both exact RBCD and exact RBK where the built-in direct solver (sometimes referred to as "backslash") is used.

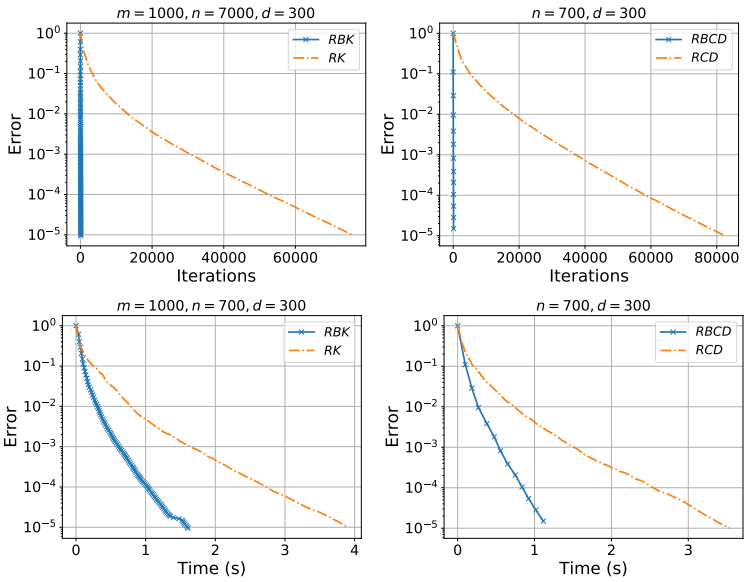
Experimental setup: For the construction of consistent linear systems $\mathbf{A}x = b$ we use the setup described in Section 2.7.1. In particular, the linear systems used for the numerical evaluation of iRBK and iRBCD have been generated as described in Section 2.7.1 for algorithms mRK and mRCD, respectively.

3.6.1 Importance of large block size

Many recent works have shown that using larger block sizes can be very beneficial for the performance of randomized iterative algorithms [73, 166, 134, 113]. In Figure 3.2 we numerically verify this statement. We show that both RBK and RBCD (no inexact updates) outperform in number of iterations and wall clock time their serial variants where only one coordinate is chosen (block of size $d = 1$) per iteration. This justify the necessity of choosing methods with large block sizes. Recall that this is precisely the class of algorithms that could have an expensive subproblem in their update rule which is required to be solved exactly and as a result can benefit the most from the introduction of inexactness.

[11] Note that from Section 3.5 and the correspondence between the primal and dual methods, iSDSA will have similar behavior when is applied to the dual problem (1.26).

[12] Recall that in order to use CG, the matrix $\mathbf{M}_k$ that appears in linear system (3.15) should be positive definite. This is true in the case that the matrix $\mathbf{A}$ of the original system has full column rank matrix. Note however that the analysis of Section 3.4 holds for any consistent linear system $\mathbf{A}x = b$ and without making any further assumption on its structure or the linearly convergence methods.



(a) RK vs RBK (b) RCD vs RBCD

Figure 3.2: Comparison of the performance of the exact RBK and RBCD with their non-block variants RK and RCD. For the Kaczmarz methods (first column) $\mathbf{A} \in \mathbb{R}^{1000,700}$ is a Gaussian matrix and for the Coordinate descent methods (second column) $\mathbf{A} = \mathbf{P}^\top \mathbf{P} \in \mathbb{R}^{700\times 700}$ where $\mathbf{P} \in \mathbb{R}^{1000\times 700}$ is Gaussian matrix. To guarantee consistency $b = \mathbf{A}z$ where z is also Gaussian vector. The block size that chosen for the block variants is $d = 300$.

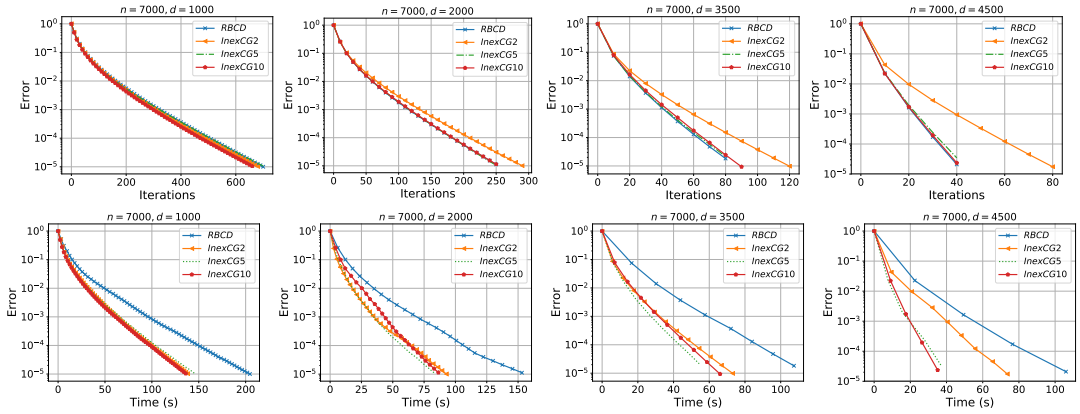
3.6.2 Inexactness and block size (iRBCD)

In this experiment, we first construct a positive definite linear system following the previously described procedure for iRBCD. We first generate a Gaussian matrix $\mathbf{P} \in \mathbb{R}^{10000\times 7000}$ and then the positive definite matrix $\mathbf{A} = \mathbf{P}^\top \mathbf{P} \in \mathbb{R}^{7000\times 7000}$ is used to define a consistent liner system. We run iRBCD in this specific linear system and compare its performance with its exact variance for several block sizes d (numbers of column of matrix $\mathbf{S}$). For evaluating the inexact solution of the linear system in the update rule we run CG for either 2, 5 or 10 iterations. In Figure 3.3, we plot the evolution of the relative error in terms of both the number of iterations and the wall-clock time.

We observe that for any block size the inexact methods are always faster in terms of wall clock time than their exact variants even if they require (as is expected) equal or larger number of iterations. Moreover it is obvious that the performance of the inexact method becomes much better than the exact variant as the size d increases and as a results the sub-problem that needs to be solved in each step becomes more expensive. It is worth to highlight that for the chosen systems, the exact RBCD behaves better in terms of wall clock time as the size of block increases (this coincides with the findings of the previous experiment).

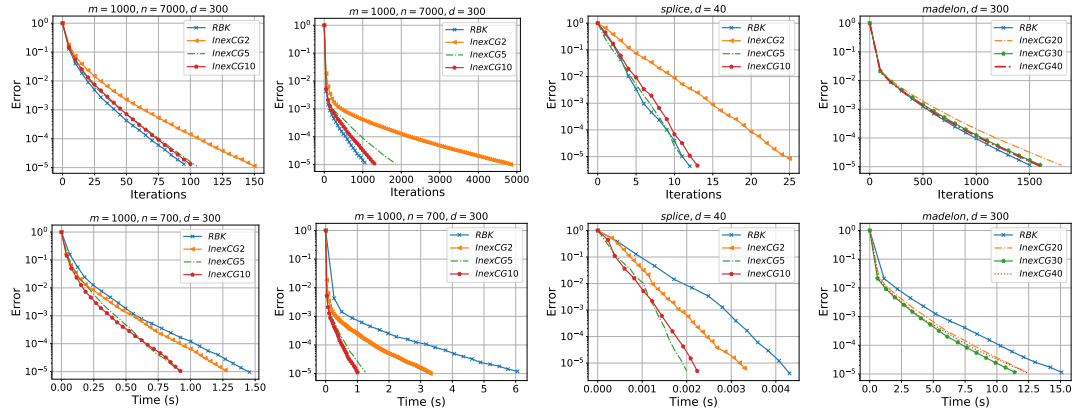
3.6.3 Evaluation of iRBK

In the last experiment we evaluate the performance of iRBK in both synthetic and real datasets. For computing the inexact solution of the linear system in the update rule we run CG for pre-specified number of iterations that can vary depending the datasets. In particular, we compare iRBK and RBK on synthetic linear systems generated with the Julia Gaussian matrix functions "randn(m,n)" and "sprandn(m,n,r)" (input r of sprandn function indicates the density of the matrix). For the real datasets, we test the performance of iRBK and RBK using real matrices from the library of support vector machine problems LIBSVM [23]. Each dataset of the LIBSVM consists of a matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m features and n characteristics) and a vector of labels $b \in \mathbb{R}^m$. In our experiments we choose to use only the matrices of the datasets and



(a) d=1000 (b) d=2000 (c) d=3500 (d) d=4500

Figure 3.3: Performance of iRBCD (InexactCG) and exact RBCD for solving a consistent linear systems with $\mathbf{A} = \mathbf{P}^\top \mathbf{P} \in \mathbb{R}^{7000\times 7000}$, where $\mathbf{P} \in \mathbb{R}^{10000\times 7000}$ is a Gaussian matrix. The right hand side for the system is chosen to be $b = \mathbf{A}z$ where z is also a Gaussian vector. Several block sizes are used: $d = 1000, 2000, 3500, 4500$. The graphs in the first (second) row plot the iterations (time) against relative error $\|x^k - x^*\|^2_{\mathbf{A}}/\|x^*\|^2_{\mathbf{A}}$.



(a) randn(m,n) (b) sprandn(m,n,0.01) (c) splice (d) madelon

Figure 3.4: The performance of iRBK (InexactCG) and RBK on synthetic and real datasets. Synthetic matrices: (a) randn(m,n) with (m,n)=(1000,700), (b) sprandn(m,n,0.01) with (m,n)=(1000,700). Real Matrices from LIBSVM [23] : (c) splice : (m,n)=(1000,60), (d) madelon: (m,n)=(2000,500). The graphs in the first (second) row plot the iterations (time) against relative error $\|x^k - x^*\|^2/\|x^*\|^2$. The quantity d in the title of each plot indicates the size of the block size for both iRBK and RBK.

ignore the label vectors [13]. As before, to ensure consistency of the linear system, we choose a Gaussian vector $z \in \mathbb{R}^n$ and the right hand side of the linear system is set to $b = \mathbf{A}z$ (for both the synthetic and the real matrices). By observing Figure 3.4 it is clear that for all problems under study the performance of iRBK in terms of wall clock time is much better than its exact variant RBK.

3.7 Conclusion

In this chapter we propose and analyze inexact variants of several stochastic algorithms for solving quadratic optimization problems and linear systems. We provide linear convergence rate under several assumptions on the inexactness error. The proposed methods require more iterations than their exact variants to achieve the same accuracy. However, as we show through

[13] Note that the real matrices of the Splice and Madelon datasets are full rank matrices.

our numerical evaluations, the inexact algorithms require significantly less time to converge.

With the continuously increasing size of datasets, inexactness should definitely be a tool that practitioners should use in their implementations even in the case of stochastic methods that have much cheaper-to-compute iteration complexity than their deterministic variants. Recently, accelerated and parallel stochastic optimization methods [115, 168, 192] have been proposed for solving linear systems. We speculate that the addition of inexactness to these update rules will lead to methods faster in practice. We also believe that our approach and complexity results can be extended to the more general case of minimization of convex and non-convex functions in the stochastic setting.

3.8 Proofs of Main Results

In our convergence analysis we use several popular inequalities. Look Table 3.3 for the abbreviations and the relevant formulas.

A key step in the proofs of the theorems is to use the tower property of the expectation. We use it in the form

$$\mathbb{E}[\mathbb{E}[\mathbb{E}[X \mid x^k, \mathbf{S}_k] \mid x^k]] = \mathbb{E}[X], \tag{3.26}$$

where X is some random variable. In all proofs we perform the three expectations in order, from the innermost to the outermost. Similar to the main part of this chapter we use $\rho = 1 - \omega(2-\omega)\lambda_{\min}^{+}$.

The following remark on random variables is also used in our proofs.

Remark 10. *Let x and y be random vectors and let σ positive constant. If we assume $\mathbb{E}[\|x\|_{\mathbf{B}}^2 \mid y] \leq \sigma^2$ then by using the variance inequality (check Table 3.3) we obtain $\mathbb{E}[\|x\|_{\mathbf{B}} \mid y] \leq \sigma$. In our setting if we assume $\mathbb{E}[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \leq \sigma_k^2$ where ϵ^k is the inexactness error and x^k is the current iterate then by the variance inequality it holds that $\mathbb{E}[\|\epsilon^k\|_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \leq \sigma_k$.*

Useful inequalities			
Inequalities (Full names)	Abbreviations	Formula	Assumptions
Jensen Inequality	*Jensen*	$f[\mathbb{E}(x)] \leq \mathbb{E}[f(x)]$	f is convex
Conditioned Jensen	*C.J.*	$f(\mathbb{E}[x \mid s]) \leq \mathbb{E}[f(x) \mid s]$	f is convex
Cauchy-Swartz (B-norm)	*C.S.*	$\lvert\langle a, b\rangle_{\mathbf{B}}\rvert \leq \|a\|_{\mathbf{B}}\|b\|_{\mathbf{B}}$	$a, b \in \mathbb{R}^n$
Variance Inequality	*V.I*	$(\mathbb{E}[X])^2 \leq \mathbb{E}[X^2]$	X random variable

Table 3.3: Popular inequalities with abbreviations and formulas.

3.8.1 Proof of Theorem 20

Proof. First we decompose:

$$\begin{aligned} \|x^{k+1} - x^*\|_{\mathbf{B}}^2 &= \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*) + \epsilon^k\|_{\mathbf{B}}^2 \\ &= \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}}^2 + \|\epsilon^k\|_{\mathbf{B}}^2 \\ &\quad +2\left\langle(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*), \epsilon^k\right\rangle. \end{aligned} \tag{3.27}$$

Applying the innermost expectation of (3.26) to (3.27), we get:

$$\begin{aligned} \mathbb{E}\left[\|x^{k+1} - x^*\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k\right] &= \underbrace{\mathbb{E}\left[\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k\right]}_{T1} + \underbrace{\mathbb{E}\left[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k\right]}_{T2} \\ &\quad +2\underbrace{\mathbb{E}\left[\left\langle(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*), \epsilon^k\right\rangle_{\mathbf{B}} \mid x^k, \mathbf{S}_k\right]}_{T3}. \end{aligned} \tag{3.28}$$

We now analyze the three expression T1,T2,T3 separately.

Note that an upper bound for the expression T2 can be directly obtained from the assumption

$$T2 = \mathbb{E}[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \leq \sigma_k^2. \tag{3.29}$$

The first expression can be written as:

$$\begin{aligned} T1 = \mathbb{E}[\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] &= \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}}^2 \\ &\overset{(1.42)}{=} \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega(2-\omega) f_{\mathbf{S}_k}(x^k) \end{aligned} \tag{3.30}$$

For expression T3:

$$\begin{aligned} \mathbb{E}\left[\left\langle (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*), \epsilon^k \right\rangle_{\mathbf{B}} \mid x^k, \mathbf{S}_k\right] &= \left\langle (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*), \mathbb{E}[\epsilon^k \mid x^k, \mathbf{S}_k] \right\rangle_{\mathbf{B}} \\ &\overset{\text{C.S.}}{\leq} \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} \|\mathbb{E}[\epsilon^k \mid x^k, \mathbf{S}_k]\|_{\mathbf{B}} \\ &\overset{\text{C.J.}}{\leq} \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} \mathbb{E}[\|\epsilon^k\|_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \\ &\overset{(*)}{\leq} \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} \sigma_k, \end{aligned} \tag{3.31}$$

where in the inequality $(*)$ we use Remark 10 and (3.5).

By substituting the bounds (3.29), (3.30), and (3.31) into (3.28) we obtain:

$$\begin{aligned} \mathbb{E}\left[\|x^{k+1} - x^*\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k\right] \leq\ & \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega(2-\omega) f_{\mathbf{S}_k}(x^k) + \sigma_k^2 \\ & + 2\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} \sigma_k. \end{aligned} \tag{3.32}$$

We now take the middle expectation (see (3.26)) and apply it to inequality (3.32):

$$\begin{aligned} \mathbb{E}\left[\mathbb{E}[\|x^{k+1} - x^*\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \mid x^k\right] \leq\ & \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega(2-\omega) f(x^k) + \sigma_k^2 \\ & + 2\mathbb{E}[\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} \mid x^k] \sigma_k. \end{aligned} \tag{3.33}$$

Now let us find a bound on the quantity $\mathbb{E}\left[\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} \mid x^k\right]$. Note that from (1.43) and (1.42) we have that $\mathbb{E}\left[\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}}^2 \mid x^k\right] \leq \rho\|x^k - x^*\|_{\mathbf{B}}^2$. By using Remark 10 in the last inequality we obtain:

$$\mathbb{E}\left[\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} \mid x^k\right] = \sqrt{\rho}\|x^k - x^*\|_{\mathbf{B}}. \tag{3.34}$$

By substituting (3.34) in (3.33):

$$\begin{aligned} \mathbb{E}[\mathbb{E}[\|x^{k+1} - x^*\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \mid x^k] &\leq \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega(2-\omega) f(x^k) + \sigma_k^2 \\ &\quad + 2\sigma_k\sqrt{\rho}\|x^k - x^*\|_{\mathbf{B}} \\ &\overset{(1.43)}{\leq} \rho\|x^k - x^*\|_{\mathbf{B}}^2 + \sigma_k^2 + 2\sigma_k\sqrt{\rho}\|x^k - x^*\|_{\mathbf{B}} \end{aligned} \tag{3.35}$$

We take the final expectation (outermost expectation in the tower rule (3.26)) on the above expression to find:

$$\begin{aligned} \mathbb{E}[\|x^{k+1} - x^*\|_{\mathbf{B}}^2] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[\|x^{k+1} - x^*\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \mid x^k]] \\ &\leq \rho\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] + \sigma_k^2 + 2\sigma_k\sqrt{\rho}\ \mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}] \\ &\overset{V.I}{\leq} \rho\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] + \sigma_k^2 + 2\sigma_k\sqrt{\rho}\sqrt{\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2]} \end{aligned} \tag{3.36}$$

Using $r_k = \mathbb{E}\left[\|x^k - x^*\|_{\mathbf{B}}^2\right]$ equation (3.36) takes the form:

$$r_{k+1} \leq \rho r_k + \sigma_k^2 + 2\sigma_k\sqrt{\rho}\sqrt{r_k} = \left(\sqrt{\rho r_k} + \sigma_k\right)^2$$

If we further substitute $p_k = \sqrt{r_k}$ and $\ell = \sqrt{\rho}$ the recurrence simplifies to:

$$p_{k+1} \leq \quad \ell p_k + \sigma_k$$

By unrolling the final inequality:

$$p_k \leq \ell^k r_0 + (\ell^0 \sigma_{k-1} + \ell \sigma_{k-2} + \cdots + \ell^{k-1}\sigma_0) = \ell^k p_0 + \sum_{i=0}^{k-1} \ell^{k-1-i}\sigma_i.$$

Hence,

$$\sqrt{\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2]} \leq \rho^{k/2}\|x^0 - x^*\|_{\mathbf{B}} + \sum_{i=0}^{k-1} \rho^{\frac{k-1-i}{2}}\sigma_i.$$

The result is obtained by using V.I in the last expression. □

3.8.2 Proof of Corollary 21

By denoting $r_k = \mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}]$ in (3.9) we obtain:

$$r_k \leq \rho^{k/2} r_0 + \rho^{1/2}\sigma \sum_{i=0}^{k-1} \rho^{k-1-i} = \rho^{k/2} r_0 + \rho^{1/2}\sigma \sum_{i=0}^{k-1} \rho^i = \rho^{k/2} r_0 + \rho^{1/2}\sigma \frac{1-\rho^k}{1-\rho}.$$

Since $1 - \rho^k \leq 1$ the result is obtained.

3.8.3 Proof of Theorem 22

In order to prove Theorem 22 we need to follow similar steps to the proof of Theorem 20. The main differences of the two proofs appear at the points that we need to upper bound the norm of the inexactness error ($\|\epsilon^k\|^2$). In particular instead of using the general sequence $\sigma_k^2 \in \mathbb{R}$ we utilize the bound $q^2\|x^k - x^*\|_{\mathbf{B}}^2$ from Assumption 1*b*. Thus, it is sufficient to focus at the parts of the proof that these bound is used.

Similar to the proof of Theorem 20 we first decompose to obtain the equation (3.28). There, the expression T1 can be upper bounded from (3.30) but now using the Assumption 1*b* the expression T2 and T3 can be upper bounded as follows:

$$T2 = \mathbb{E}[\|\epsilon^k\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \leq q^2\|x^k - x^*\|_{\mathbf{B}}^2. \tag{3.37}$$

$$\begin{aligned} T3 \quad &= \quad \mathbb{E}[\left\langle (\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*), \epsilon^k \right\rangle_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \\ &\overset{\text{Remark 10 and (3.31)}}{\leq} \quad \|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} q\|x^k - x^*\| \end{aligned} \tag{3.38}$$

As a result by substituting the bounds (3.30), (3.37), and (3.38) into (3.28) we obtain:

$$\begin{aligned} \mathbb{E}[\|x^{k+1} - x^*\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \quad &\overset{(3.28)}{\leq} \quad \|x^k - x^*\|_{\mathbf{B}}^2 - 2\omega(2-\omega) f_{\mathbf{S}_k}(x^k) + q^2\|x^k - x^*\|_{\mathbf{B}}^2 \\ &\quad +2\|(\mathbf{I} - \omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k - x^*)\|_{\mathbf{B}} q\|x^k - x^*\|_{\mathbf{B}}. \end{aligned} \tag{3.39}$$

By following the same steps to the proof of Theorem 20 the equation (3.35) takes the form:

$$\begin{aligned} \mathbb{E}[\mathbb{E}[\|x^{k+1} - x^*\|_{\mathbf{B}}^2 \mid x^k, \mathbf{S}_k] \mid x^k] \quad &\leq \quad \rho\|x^k - x^*\|_{\mathbf{B}}^2 + q^2\|x^k - x^*\|_{\mathbf{B}}^2 \\ &\quad + \quad 2q\|x^k - x^*\|_{\mathbf{B}}\sqrt{\rho}\|x^k - x^*\|_{\mathbf{B}} \\ &= \quad \left(\rho + 2q\sqrt{\rho} + q^2\right)\|x^k - x^*\|_{\mathbf{B}}^2. \\ &= \quad \left(\sqrt{\rho} + q\right)^2\|x^k - x^*\|_{\mathbf{B}}^2 \end{aligned} \tag{3.40}$$

We take the final expectation (outermost expectation in the tower rule (3.26)) on the above

expression to find:

$$\begin{aligned}\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}}] &= \mathbb{E}\left[\mathbb{E}[\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \mid x^k]\right] \\ &\leq \left(\sqrt{\rho}+q\right)^2 \mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}]. \end{aligned} \tag{3.41}$$

The final result follows by unrolling the recurrence.

3.8.4 Proof of Theorem 23

Proof. Similar to the previous two proofs by decomposing the update rule and using the innermost expectation of (3.26) we obtain equation (3.28). An upper bound of expression T1 is again given by inequality (3.30). For the expression T2 depending the assumption that we have on the norm of the inexactness error different upper bounds can be used. In particular,

(i) If Assumption 1 holds then: $T2 = \mathbb{E}[\|\epsilon^k\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \leq \sigma_k^2$.

(ii) If Assumption 1b holds then: $T2 = \mathbb{E}[\|\epsilon^k\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \leq \sigma_k^2 = q^2\|x^k-x^*\|^2_{\mathbf{B}}$.

(iii) If Assumption 1c holds then: $T2 = \mathbb{E}[\|\epsilon^k\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \leq \sigma_k^2 = 2q^2 f_{S_k}(x^k)$.

The main difference from the previous proofs, is that due to the Assumption 2 and tower property (3.26) the expression T3 will eventually be equal to zero. More specifically, we have that:

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k-x^*),\epsilon^k\right\rangle_{\mathbf{B}} \mid x^k, \mathbf{S}_k\right] \mid x^k\right]\right] &= \mathbb{E}\left[\left\langle(\mathbf{I}-\omega\mathbf{B}^{-1}\mathbf{Z}_k)(x^k-x^*),\epsilon^k\right\rangle_{\mathbf{B}}\right] \\ &= T3 = 0, \end{aligned} \tag{3.42}$$

Thus, in this case equation (3.32) takes the form:

$$\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \leq \|x^k-x^*\|^2_{\mathbf{B}} - 2\omega(2-\omega)f_{\mathbf{S}_k}(x^k) + \sigma_k^2. \tag{3.43}$$

Using the above expression depending the assumption that we have we obtain the following results:

(i) By taking the middle expectation (see (3.26)) and apply it to the above inequality:

$$\begin{aligned}\mathbb{E}\left[\mathbb{E}\left[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k\right] \mid x^k\right] &\leq \|x^k-x^*\|^2_{\mathbf{B}} - 2\omega(2-\omega)f(x^k) + \mathbb{E}[\sigma_k^2 \mid x^k] \\ &\overset{(1.43)}{\leq} \rho\|x^k-x^*\|^2_{\mathbf{B}} + \mathbb{E}[\sigma_k^2 \mid x^k] \end{aligned} \tag{3.44}$$

We take the final expectation (outermost expectation in the tower rule (3.26)) on the above expression to find:

$$\begin{aligned}\mathbb{E}\left[\|x^{k+1}-x^*\|^2_{\mathbf{B}}\right] &= \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k\right] \mid x^k\right]\right] \\ &\leq \rho\mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}] + \mathbb{E}[\mathbb{E}[\sigma_k^2 \mid x^k]] \\ &= \rho\mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}] + \mathbb{E}[\sigma_k^2] \\ &= \rho\mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}] + \bar{\sigma}_k^2 \end{aligned} \tag{3.45}$$

Using $r_k = \mathbb{E}\left[\|x^k-x^*\|^2_{\mathbf{B}}\right]$ the last inequality takes the form $r_{k+1} \leq \rho r_k + \bar{\sigma}_k^2$. By unrolling the last expression: $r_k \leq \rho^k r_0 + (\rho^0\bar{\sigma}_{k-1}^2 + \rho\bar{\sigma}_{k-2}^2 + \cdots + \rho^{k-1}\bar{\sigma}_0^2) = \rho^k r_0 + \sum_{i=0}^{k-1}\rho^{k-1-i}\bar{\sigma}_i^2$. Hence,

$$\mathbb{E}[\|x^k-x^*\|^2_{\mathbf{B}}] \leq \rho^k\|x^0-x^*\|^2_{\mathbf{B}} + \sum_{i=0}^{k-1}\rho^{k-1-i}\bar{\sigma}_i^2.$$

(ii) For the case (ii) inequality (3.43) takes the form:

$$\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \leq \|x^k-x^*\|^2_{\mathbf{B}} - 2\omega(2-\omega)f_{\mathbf{S}_k}(x^k) + q^2\|x^k-x^*\|^2_{\mathbf{B}},$$

and by taking the middle expectation (see (3.26)) we obtain:

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \mid x^k] &\leq \|x^k - x^*\|^2_{\mathbf{B}} - 2\omega(2-\omega)f(x^k) + q^2\|x^k - x^*\|^2_{\mathbf{B}} \\ &\overset{(1.43)}{\leq} \rho\|x^k - x^*\|^2_{\mathbf{B}} + q^2\|x^k - x^*\|^2_{\mathbf{B}} \\ &= (\rho + q^2)\|x^k - x^*\|^2_{\mathbf{B}}. \end{aligned} \tag{3.46}$$

By taking the final expectation of the tower rule (3.26) and apply it to the above inequality:

$$\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}}] \leq (\rho + q^2)\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}]. \tag{3.47}$$

and the result is obtain by unrolling the last expression.

(iii) For the case (iii) inequality (3.43) takes the form:

$$\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \leq \|x^k - x^*\|^2_{\mathbf{B}} - 2(\omega(2-\omega) - q^2)f_{\mathbf{S}_k}(x^k), \tag{3.48}$$

and by taking the middle expectation (see (3.26)) we obtain:

$$\begin{aligned}\mathbb{E}[\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}} \mid x^k, \mathbf{S}_k] \mid x^k] &\leq \|x^k - x^*\|^2_{\mathbf{B}} - 2(\omega(2-\omega) - q^2)f(x^k) \\ &\overset{(1.37)}{\leq} \|x^k - x^*\|^2_{\mathbf{B}} - (\omega(2-\omega) - q^2)\lambda^+_{\min}\|x^k - x^*\|^2_{\mathbf{B}} \\ &= (1 - (\omega(2-\omega) - q^2)\lambda^+_{\min})\|x^k - x^*\|^2_{\mathbf{B}}. \end{aligned} \tag{3.49}$$

By taking the final expectation of the tower rule (3.26) to the above inequality:

$$\mathbb{E}[\|x^{k+1}-x^*\|^2_{\mathbf{B}}] \leq (1 - (\omega(2-\omega) - q^2)\lambda^+_{\min})\mathbb{E}[\|x^k - x^*\|^2_{\mathbf{B}}]. \tag{3.50}$$

and the result is obtain by unrolling the last expression.

□

Chapter 4

Revisiting Randomized Gossip Algorithms

4.1 Introduction

Average consensus is a fundamental problem in distributed computing and multi-agent systems. It comes up in many real world applications such as coordination of autonomous agents, estimation, rumour spreading in social networks, PageRank and distributed data fusion on ad-hoc networks and decentralized optimization. Due to its great importance there is much classical [190, 35] and recent [202, 201, 16] work on the design of efficient algorithms/protocols for solving it.

In the average consensus (AC) problem we are given an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E}$. Each node $i \in \mathcal{V}$ "knows" a private value $c_i \in \mathbb{R}$. The goal of AC is for every node to compute the average of these private values, $\bar{c} := \frac{1}{n}\sum_i c_i$, in a decentralized fashion. That is, the exchange of information can only occur between connected nodes (neighbors).

Among the most attractive protocols for solving the average consensus problem are gossip algorithms. The development and design of gossip algorithms was studied extensively in the last decade. The seminal 2006 paper of Boyd et al. [16] motivated a fury of subsequent research and gossip algorithms now appear in many applications, including distributed data fusion in sensor networks [202], load balancing [29] and clock synchronization [56]. For a survey of selected relevant work prior to 2010, we refer the reader to the work of Dimakis et al. [42]. For more recent results on randomized gossip algorithms we suggest [217, 106, 148, 109, 131, 6]. See also [43, 7, 149].

4.1.1 Main contributions

In this chapter, we connect two areas of research which until now have remained remarkably disjoint in the literature: randomized iterative (projection) methods for solving linear systems and randomized gossip protocols for solving the average consensus. This connection enables us to make contributions by borrowing from each body of literature to the other and using it we propose a new framework for the design and analysis of novel efficient randomized gossip protocols.

The main contributions of our work include:

- **RandNLA.** We show how classical randomized iterative methods for solving linear systems can be interpreted as gossip algorithms when applied to special systems encoding the underlying network and explain in detail their decentralized nature. Through our general framework we recover a comprehensive array of well-known gossip protocols as special cases. In addition our approach allows for the development of novel block and dual variants of all of these methods. From a numerical analysis viewpoint our work is the first that explores in depth, the decentralized nature of randomized iterative methods

for solving linear systems and proposes them as efficient methods for solving the average consensus problem (and its weighted variant).

- **Weighted AC.** The methods presented in this chapter solve the more general *weighted* average consensus (Weighted AC) problem (Section 4.3.1) popular in the area of distributed cooperative spectrum sensing networks. The proposed protocols are the first randomized gossip algorithms that directly solve this problem with finite-time convergence rate analysis. In particular, we prove linear convergence of the proposed protocols and explain how we can obtain further acceleration using momentum. To the best of our knowledge, the existing decentralized protocols that solve the weighted average consensus problem show convergence but without convergence analysis.

- **Acceleration.** We present novel and provably *accelerated* randomized gossip protocols. In each step, of the proposed algorithms, all nodes of the network update their values using their own information but only a subset of them exchange messages. The protocols are inspired by the recently proposed accelerated variants of randomized Kaczmarz-type methods and use momentum terms on top of the sketch and project update rule (gossip communication) to obtain better theoretical and practical performance. To the best of our knowledge, our accelerated protocols are the first randomized gossip algorithms that converge to a consensus with a provably accelerated linear rate without making any further assumptions on the structure of the network. Achieving an accelerated linear rate in this setting using randomized gossip protocols was an open problem.

- **Duality.** We reveal a hidden duality of randomized gossip algorithms, with the dual iterative process maintaining variables attached to the edges of the network. We show how the randomized coordinate descent and randomized Newton methods work as edge-based dual randomized gossip algorithms.

- **Experiments.** We corroborate our theoretical results with extensive experimental testing on typical wireless network topologies. We numerically verify the linear convergence of the our protocols for solving the weighted AC problem. We explain the benefit of using block variants in the gossip protocols where more than two nodes update their values in each iteration. We explore the performance of the proposed provably accelerated gossip protocols and show that they significantly outperform the standard pairwise gossip algorithm and existing fast pairwise gossip protocols with momentum. An experiment showing the importance of over-relaxation in the gossip setting is also presented.

We believe that this work could potentially open up new avenues of research in the area of decentralized gossip protocols.

4.1.2 Structure of the chapter

This chapter is organized as follows. Section 4.2 introduces the necessary background on basic randomized iterative methods for linear systems that will be used for the development of randomized gossip protocols. Related work on the literature of linear system solvers, randomized gossip algorithms for averaging and gossip algorithms for consensus optimization is presented. In Section 4.3 the more general weighted average consensus problem is described and the connections between the two areas of research (randomized projection methods for linear systems and gossip algorithms) is established. In particular we explain how methods for solving linear systems can be interpreted as gossip algorithms when applied to special systems encoding the underlying network and elaborate in detail their distributed nature. Novel block gossip variants are also presented. In Section 4.4 we describe and analyze fast and provably accelerated randomized gossip algorithms. In each step of these protocols all nodes of the network update their values but only a subset of them exchange their private values. Section 4.5 describes dual randomized gossip algorithms that operate with values that are associated to the edges of the network and Section 4.6 highlights further connections between methods for solving linear systems and gossip algorithms. Numerical evaluation of the new gossip protocols is presented in Section 4.7. Finally, concluding remarks are given in Section 4.8.

4.1.3 Notation

For convenience, a table of the most frequently used notation of this chapter is included in Section A.2. In particular, with boldface upper-case letters denote matrices; $\mathbf{I}$ is the identity matrix. By $\|\cdot\|$ and $\|\cdot\|_F$ we denote the Euclidean norm and the Frobenius norm, respectively. For a positive integer number n, we write $[n] := \{1, 2, \ldots, n\}$. By $\mathcal{L}$ we denote the solution set of the linear system $\mathbf{A}x = b$, where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$.

Vector $x^k = (x_1^k, \ldots, x_n^k) \in \mathbb{R}^n$ represents the vector with the private values of the n nodes of the network at the k^{th} iteration while with x_i^k we denote the value of node $i \in [n]$ at the k^{th} iteration. $\mathcal{N}_i \subseteq \mathcal{V}$ denotes the set of nodes that are neighbors of node $i \in \mathcal{V}$. By $\alpha(\mathcal{G})$ we denote the algebraic connectivity of graph $\mathcal{G}$. Throughout the chapter, x^* is the projection of x^0 onto $\mathcal{L}$ in the $\mathbf{B}$-norm. We write $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$.

The complexity of all gossip protocols presented in this chapter is described by the spectrum of matrix

$$\mathbf{W} = \mathbf{B}^{-1/2}\mathbf{A}^\top \mathbb{E}[\mathbf{H}]\mathbf{A}\mathbf{B}^{-1/2} \overset{(1.11)}{=} \mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}, \tag{4.1}$$

where the expectation is taken over $\mathbf{S} \sim \mathcal{D}$. With $\lambda_{\min}^+$ and $\lambda_{\max}$ we indicate the smallest nonzero and the largest eigenvalue of matrix $\mathbf{W}$, respectively. Recall that this is exactly the same matrix used in the previous chapters of this thesis.

Finally, with $\mathbf{Q} \in \mathbb{R}^{|\mathcal{E}|\times n}$ we define the incidence matrix and with $\mathbf{L} \in \mathbb{R}^{n\times n}$ the Laplacian matrix of the network. Note that it holds that $\mathbf{L} = \mathbf{Q}^\top\mathbf{Q}$. Further, with $\mathbf{D}$ we denote the degree matrix of the graph. That is, $\mathbf{D} = \mathbf{Diag}(d_1, d_2, \ldots, d_n) \in \mathbb{R}^{n\times n}$ where d_i is the degree of node $i \in \mathcal{V}$.

4.2 Background - Technical Preliminaries

As we have already mentioned in this thesis, solving linear systems is a central problem in numerical linear algebra and plays an important role in computer science, control theory, scientific computing, optimization, computer vision, machine learning, and many other fields. With the advent of the age of big data, practitioners are looking for ways to solve linear systems of unprecedented sizes. In this large scale setting, randomized iterative methods are preferred mainly because of their cheap per iteration cost and because they can easily scale to extreme dimensions.

4.2.1 Randomized iterative methods for linear systems

Recall that in the introduction of this thesis we presented the sketch and project method (1.25) and we explained how this algorithm is identical to SGD (1.17), SN (1.18) and SPP (1.19) for solving the stochastic quadratic optimization problem (1.6). For the benefit of the reader and for the easier comparison to the gossip algorithms, a formal presentation of the Sketch and Project method for solving a consistent linear system $\mathbf{A}x = b$ is presented in Algorithm 8.

Algorithm 8 Sketch and Project Method [168]

1: **Parameters:** Distribution $\mathcal{D}$ from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$; momentum parameter β.
2: **Initialize:** $x^0, x^1 \in \mathbb{R}^n$
3: **for** $k = 0, 1, 2, \ldots$ **do**
4: Draw a fresh $\mathbf{S}_k \sim \mathcal{D}$
5: Set $x^{k+1} = x^k - \omega\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k)^\dagger\mathbf{S}_k^\top(\mathbf{A}x^k - b)$
6: **end for**
7: **Output:** The last iterate x^k

In this chapter, we are mostly interested in two special cases of the sketch and project framework— the randomized Kaczmarz (RK) method and its block variant, the randomized block Kaczmarz (RBK) method. In addition, in the following sections we present novel scaled

and accelerated variants of these two selected cases and interpret their gossip nature. In particular, we focus on explaining how these methods can solve the average consensus problem and its more general version, the weighted average consensus (subsection 4.3.1).

Let $e_i \in \mathbb{R}^m$ be the i^{th} unit coordinate vector in $\mathbb{R}^m$ and let $\mathbf{I}_{:C}$ be column submatrix of the $m \times m$ identity matrix with columns indexed by $C \subseteq [m]$. Then RK and RBK methods can be obtained as special cases of Algorithm 8 as follows:

- RK: Let $\mathbf{B} = \mathbf{I}$ and $\mathbf{S}_k = e_i$, where $i \in [m]$ is chosen independently at each iteration, with probability $p_i > 0$. In this setup the update rule of Algorithm 8 simplifies to

$$x^{k+1} = x^k - \omega \frac{\mathbf{A}_{i:}x^k - b_i}{\|\mathbf{A}_{i:}\|^2}\mathbf{A}_{i:}^\top. \tag{4.2}$$

- RBK: Let $\mathbf{B} = \mathbf{I}$ and $\mathbf{S} = \mathbf{I}_{:C}$, where set $C \subseteq [m]$ is chosen independently at each iteration, with probability $p_C \geq 0$. In this setup the update rule of Algorithm 8 simplifies to

$$x^{k+1} = x^k - \omega \mathbf{A}_{C:}^\top (\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger (\mathbf{A}_{C:}x^k - b_C). \tag{4.3}$$

As we explained in Chapter 1, the sketch and project method, converges linearly to one particular solution of the linear system: the projection (on $\mathbf{B}$-norm) of the initial iterate x^0 onto the solution set of the linear system, $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Therefore, the method solve the best approximation problem (1.22).

The convergence performance of the Sketch and Project method (Algorithm 8) for solving the best approximation problem is described by the following theorem[1].

Theorem 30. *Let assume exactness and let $\{x^k\}_{k=0}^\infty$ be the iterates produced by the sketch and project method (Algorithm 8) with step-size $\omega \in (0,2)$. Set, $x^* = \Pi_{\mathcal{L},\mathbf{B}}(x^0)$. Then,*

$$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \rho^k \|x^0 - x^*\|_{\mathbf{B}}^2, \tag{4.4}$$

where

$$\rho := 1 - \omega(2-\omega)\lambda_{\min}^+ \in [0,1]. \tag{4.5}$$

In other words, using standard arguments, from Theorem 30 we observe that for a given $\epsilon > 0$ we have that:

$$k \geq \frac{1}{1-\rho}\log\left(\frac{1}{\epsilon}\right) \quad \Rightarrow \quad \mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \epsilon \|x^0 - x^*\|_{\mathbf{B}}^2$$

4.2.2 Other related work

Gossip algorithms for average consensus The problem of average consensus has been extensively studied in the automatic control and signal processing literature for the past two decades [42], and was first introduced for decentralized processing in the seminal work [190]. A clear connection between the rate of convergence and spectral characteristics of the underlying network topology over which message passing occurs was first established in [16] for pairwise randomized gossip algorithms.

Motivated by network topologies with salient properties of wireless networks (e.g., nodes can communicate directly only with other nearby nodes), several methods were proposed to accelerate the convergence of gossip algorithms. For instance, [8] proposed averaging among a set of nodes forming a path in the network (this protocol can be seen as special case of our block variants in Section 4.3.4). Broadcast gossip algorithms have also been analyzed [7] where the nodes communicate with more than one of their neighbors by broadcasting their values.

While the gossip algorithms studied in [16, 8, 7] are all first-order (the update of x^{k+1} only depends on x^k), a faster randomized pairwise gossip protocol was proposed in [19] which suggested to incorporate additional memory to accelerate convergence. The first analysis of

[1] For the proof of Theorem 30 check Theorem 3 in Section 1.5 and recall that SGD and the sketch and project method are identical in this setting.

this protocol was later proposed in [106] under strong condition. It is worth to mention that in the setting of deterministic gossip algorithms theoretical guarantees for accelerated convergence were obtained in [150, 94]. In Section 4.4 we propose fast and provably accelerated randomized gossip algorithms with memory and compare them in more detail with the fast randomized algorithm proposed in [19, 106].

Gossip algorithms for multiagent consensus optimization. In the past decade there has been substantial interest in consensus-based mulitiagent optimization methods that use gossip updates in their update rule [131, 209, 179]. In multiagent consensus optimization setting , n agents or nodes, cooperate to solve an optimization problem. In particular, a local objective function $f_i : \mathbb{R}^d \to \mathbb{R}$ is associated with each node $i \in [n]$ and the goal is for all nodes to solve the optimization problem

$$\min_{x\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{4.6}$$

by communicate only with their neighbors. In this setting gossip algorithms works in two steps by first executing some local computation followed by communication over the network [131]. Note that the average consensus problem with c_i as node i initial value can be case as a special case of the optimization problem (4.6) when the function values are $f_i(x) = (x - c_i)^2$.

Recently there has been an increasing interest in applying mulitagent optimization methods to solve convex and non-convex optimization problems arising in machine learning [189, 105, 4, 5, 25, 95, 83]. In this setting most consensus-based optimization methods make use of standard, first-order gossip, such as those described in [16], and incorporating momentum into their updates to improve their practical performance.

4.3 Sketch and Project Methods as Gossip Algorithms

In this section we show how by carefully choosing the linear system in the constraints of the best approximation problem (1.22) and the combination of the parameters of the Sketch and Project method (Algorithm 8) we can design efficient randomized gossip algorithms. We show that the proposed protocols can actually solve the weighted average consensus problem, a more general version of the average consensus problem described in Section 4.1. In particular we focus, on a scaled variant of the RK method (4.2) and on the RBK (4.3) and understand the convergence rates of these methods in the consensus setting, their distributed nature and how they are connected with existing gossip protocols.

4.3.1 Weighted average consensus

In the *weighted average consensus* (Weighted AC) problem we are given an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E}$. Each node $i \in \mathcal{V}$ holds a private value $c_i \in \mathbb{R}$ and its weight w_i. The goal of this problem is for every node to compute the weighted average of the private values,

$$\bar{c} := \frac{\sum_{i=1}^{n} w_i c_i}{\sum_{i=1}^{n} w_i},$$

in a distributed fashion. That is, the exchange of information can only occur between connected nodes (neighbors).

Note that in the special case when the weights of all nodes are the same ($w_i = r$ for all $i \in [n]$) the weighted average consensus is reduced to the standard average consensus problem. However, there are more special cases that could be interesting. For instance the weights can represent the degree of the nodes ($w_i = d_i$) or they can denote a probability vector and satisfy $\sum_i^n w_i = 1$ with $w_i > 0$.

It can be easily shown that the weighted average consensus problem can be expressed as

optimization problem as follows:

$$\min_{x=(x_1,\dots,x_n)\in\mathbb{R}^n} \frac{1}{2}\|x-c\|_{\mathbf{B}}^2 \quad \text{subject to} \quad x_1=x_2=\cdots=x_n \tag{4.7}$$

where matrix $\mathbf{B}=\mathbf{Diag}(w_1,w_2,\dots,w_n)$ is a diagonal positive definite matrix (that is $w_i>0$ for all $i\in[n]$) and $c=(c_1,\dots,c_n)^\top$ the vector with the initial values c_i of all nodes $i\in\mathcal{V}$. The optimal solution of this problem is $x_i^*=\frac{\sum_{i=1}^n w_i c_i}{\sum_{i=1}^n w_i}$ for all $i\in[n]$ which is exactly the solution of the weighted average consensus.

As we have explained, the standard average consensus problem can be cast as a special case of weighted average consensus. However, in the situation when the nodes have access to global information related to the network, such as the size of the network (number of nodes $n=|\mathcal{V}|$) and the sum of the weights $\sum_{i=1}^n w_i$, then any algorithm that solves the standard average consensus can be used to solve the weighted average consensus problem with the initial private values of the nodes changed from c_i to $\frac{nw_ic_i}{\sum_{i=1}^n w_i}$.

The weighted AC problem is popular in the area of distributed cooperative spectrum sensing networks [84, 151, 212, 213]. In this setting, one of the goals is to develop decentralized protocols for solving the cooperative sensing problem in cognitive radio systems. The weights in this case represent a ratio related to the channel conditions of each node/agent [84]. The development of methods for solving the weighted AC problem is an active area of research (check [84] for a recent comparison of existing algorithms). However, to the best of our knowledge, existing analysis for the proposed algorithms focuses on showing convergence and not on providing convergence rates. Our framework allows us to obtain novel randomized gossip algorithms for solving the weighted AC problem. In addition, we provide a tight analysis of their convergence rates. In particular, we show convergence with a linear rate. See Section 4.7.1 for an experiment confirming linear convergence of one of our proposed protocols on typical wireless network topologies.

4.3.2 Gossip algorithms through sketch and project framework

We propose that randomized gossip algorithms should be viewed as special case of the Sketch and Project update to a particular problem of the form (1.22). In particular, we let $c=(c_1,\dots,c_n)$ be the initial values stored at the nodes of $\mathcal{G}$, and choose $\mathbf{A}$ and b so that the constraint $\mathbf{A}x=b$ is equivalent to the requirement that $x_i=x_j$ (the value stored at node i is equal to the value stored at node j) for all $(i,j)\in\mathcal{E}$.

Definition 31. *We say that $\mathbf{A}x=b$ is an "average consensus (AC) system" when $\mathbf{A}x=b$ iff $x_i=x_j$ for all $(i,j)\in\mathcal{E}$.*

It is easy to see that $\mathbf{A}x=b$ is an AC system precisely when $b=0$ and the nullspace of $\mathbf{A}$ is $\{t1_n : t\in\mathbb{R}\}$, where 1_n is the vector of all ones in $\mathbb{R}^n$. Hence, $\mathbf{A}$ has rank $n-1$. Moreover in the case that $x^0=c$, it is easy to see that for any AC system, the solution of (1.22) necessarily is $x^*=\bar{c}\cdot 1_n$ — this is why we singled out AC systems. In this sense, *any* algorithm for solving (1.22) will "find" the (weighted) average $\bar{c}$. However, in order to obtain a distributed algorithm we need to make sure that only "local" (with respect to $\mathcal{G}$) exchange of information is allowed.

Choices of AC systems. It can be shown that many linear systems satisfy the above definition.

For example, we can choose:

1. $b=0$ and $\mathbf{A}=\mathbf{Q}\in\mathbb{R}^{|\mathcal{E}|\times n}$ to be the incidence matrix of $\mathcal{G}$. That is, $\mathbf{Q}\in\mathbb{R}^{|\mathcal{E}|\times n}$ such that $\mathbf{Q}x=0$ directly encodes the constraints $x_i=x_j$ for $(i,j)\in\mathcal{E}$. That is, row $e=(i,j)\in\mathcal{E}$ of matrix $\mathbf{Q}$ contains value 1 in column i, value -1 in column j (we use an arbitrary but fixed order of nodes defining each edge in order to fix $\mathbf{Q}$) and zeros elsewhere.

2. A different choice is to pick $b=0$ and $\mathbf{A}=\mathbf{L}=\mathbf{Q}^\top\mathbf{Q}$, where $\mathbf{L}$ is the Laplacian matrix of network $\mathcal{G}$.

Depending on what AC system is used, the sketch and project methods can have different interpretations as gossip protocols.

In this work we mainly focus on the above two AC systems but we highlight that other choices are possible[2]. In Section 4.4.2 for the provably accelerated gossip protocols we also use a normalized variant ($\|\mathbf{A}_{i:}\|^2 = 1$) of the Incidence matrix.

Standard form and mass preservation

Assume that $\mathbf{A}x = b$ is an AC system. Note that since $b = 0$, the update rule of Algorithm 8 simplifies to:

$$x^{k+1} = \left[\mathbf{I} - \omega\mathbf{A}^\top\mathbf{H}_k\mathbf{A}\right] x^k = \left[\mathbf{I} - \omega\mathbf{Z}_k\right] x^k. \tag{4.8}$$

This is the standard form in which randomized gossip algorithms are written. What is new here is that the iteration matrix $\mathbf{I} - \omega\mathbf{Z}_k$ has a specific structure which guarantees convergence to x^* under very weak assumptions (see Theorem 30). Note that if $x^0 = c$, i.e., the starting primal iterate is the vector of private values (as should be expected from any gossip algorithm), then the iterates of (4.8) enjoy a mass preservation property (the proof follows the fact that $\mathbf{A}1_n = 0$):

Theorem 32 (Mass preservation)**.** *If $\mathbf{A}x = b$ is an AC system, then the iterates produced by (4.8) satisfy: $\frac{1}{n}\sum_{i=1}^n x_i^k = \bar{c}$, for all $k \geq 0$.*

Proof. Let fix $k \geq 0$ then,

$$\frac{1}{n}1_n^\top x^{k+1} = \frac{1}{n}1_n^\top(\mathbf{I} - \omega\mathbf{A}^\top\mathbf{H}_k\mathbf{A})x^k = \frac{1}{n}1_n^\top\mathbf{I}x^k - \frac{1}{n}1_n^\top\omega\mathbf{A}^\top\mathbf{H}_k\mathbf{A}x^k \overset{\mathbf{A}1_n=0}{=} \frac{1}{n}1_n^\top x^k.$$

□

ε-Averaging time

Let $z^k := \|x^k - x^*\|$. The typical measure of convergence speed employed in the randomized gossip literature, called ε-averaging time and here denoted by $T_{ave}(\varepsilon)$, represents the smallest time k for which x^k gets within εz^0 from x^*, with probability greater than $1-\varepsilon$, uniformly over all starting values $x^0 = c$. More formally, we define

$$T_{ave}(\varepsilon) := \sup_{c\in\mathbb{R}^n} \inf\left\{k \;:\; \mathbb{P}\left(z^k > \varepsilon z^0\right) \leq \varepsilon\right\}.$$

This definition differs slightly from the standard one in that we use z^0 instead of $\|c\|$.

Inequality (4.4), together with Markov inequality, can be used to give a bound on $K(\varepsilon)$, formalized next:

Theorem 33. *Assume $\mathbf{A}x = b$ is an AC system. Let $x^0 = c$ and $\mathbf{B}$ be positive definite diagonal matrix. Assume exactness. Then for any $0 < \varepsilon < 1$ we have*

$$T_{ave}(\epsilon) \leq 3\frac{\log(1/\varepsilon)}{\log(1/\rho)} \leq 3\frac{\log(1/\epsilon)}{1-\rho},$$

where ρ is defined in (4.5).

Proof. See Section 4.9.1. □

Note that under the assumptions of the above theorem, $\mathbf{W} = \mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}$ only has a single zero eigenvalue, and hence $\lambda_{\min}^+(\mathbf{W})$ is the second smallest eigenvalue of $\mathbf{W}$. Thus, ρ is the second largest eigenvalue of $\mathbf{I} - \mathbf{W}$. The bound on $K(\varepsilon)$ appearing in Thm 33 is often written with ρ replaced by $\lambda_2(\mathbf{I} - \mathbf{W})$ [16].

In the rest of this section we show how two special cases of the sketch and project framework, the randomized Kaczmarz (RK) and its block variant, randomized block Kaczmatz (RBK) work as gossip algorithms for the two AC systems described above.

[2]Novel gossip algorithms can be proposed by using different AC systems to formulate the average consensus problem. For example one possibility is using the random walk normalized Laplacian $\mathbf{L}^{rw} = \mathbf{D}^{-1}\mathbf{L}$. For the case of degree-regular networks the symmetric normalized Laplacian matrix $\mathbf{L}^{sym} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ can also being used.

4.3.3 Randomized Kaczmarz method as gossip algorithm

As we described before the sketch and project update rule of Algorithm 8 has several parameters that should be chosen in advance by the user. These are the stepsize ω (relaxation parameter), the positive definite matrix $\mathbf{B}$ and the distribution $\mathcal{D}$ of the random matrices $\mathbf{S}$.

In this section we focus on one particular special case of the sketch and project framework, a scaled/weighted variant of the randomized Kaczmarz method (RK) presented in (4.2), and we show how this method works as gossip algorithm when applied to special systems encoding the underlying network. In particular, the linear systems that we solve are the two AC systems described in the previous section where the matrix is either the incidence matrix $\mathbf{Q}$ or the Laplacian matrix $\mathbf{L}$ of the network.

As we described in (4.2) the standard RK method can be cast as special case of Algorithm 8 by choosing $\mathbf{B} = \mathbf{I}$ and $\mathbf{S} = e_i$. In this section, we focus on a small modification of this algorithm and we choose the positive definite matrix $\mathbf{B}$ to be $\mathbf{B} = \mathbf{Diag}(w_1, w_2, \ldots, w_n)$, the diagonal matrix of the weights presented in the weighted average consensus problem.

Scaled RK: Let us have a general consistent linear system $\mathbf{A}x = b$ with $\mathbf{A} \in \mathbb{R}^{m\times n}$. Let us also choose $\mathbf{B} = \mathbf{Diag}(w_1, w_2, \ldots, w_n)$ and $\mathbf{S}_k = e_i$, where $i \in [m]$ is chosen in each iteration independently, with probability $p_i > 0$. In this setup the update rule of Algorithm 8 simplifies to

$$x^{k+1} = x^k - \omega \frac{e_i^\top (\mathbf{A}x^k - b)}{e_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top e_i}\mathbf{B}^{-1}\mathbf{A}^\top e_i = x^k - \omega \frac{\mathbf{A}_{i:}x^k - b_i}{\|\mathbf{B}^{-1/2}\mathbf{A}_{i:}^\top\|_2^2}\mathbf{B}^{-1}\mathbf{A}_{i:}^\top. \tag{4.9}$$

This small modification of RK allow us to solve the more general weighted average consensus presented in Section 4.3.1 (and at the same time the standard average consensus problem if $\mathbf{B} = r\mathbf{I}$ where $r \in \mathbb{R}$). To the best of our knowledge, even if this variant is special case of the general Sketch and project update, was never precisely presented before in any setting.

AC system with incidence matrix Q

Let us represent the constraints of problem (4.7) as linear system with matrix $\mathbf{A} = \mathbf{Q} \in \mathbb{R}^{|\mathcal{E}|\times n}$ be the Incidence matrix of the graph and right had side $b = 0$. Lets also assume that the random matrices $\mathbf{S} \sim \mathcal{D}$ are unit coordinate vectors in $\mathbb{R}^m = \mathbb{R}^{|\mathcal{E}|}$.

Let $e = (i, j) \in \mathcal{E}$ then from the definition of matrix $\mathbf{Q}$ we have that $\mathbf{Q}_{e:}^\top = f_i - f_j$ where f_i, f_j are unit coordinate vectors in $\mathbb{R}^n$. In addition, from the definition the diagonal positive definite matrix $\mathbf{B}$ we have that

$$\|\mathbf{B}^{-1/2}\mathbf{Q}_{e:}^\top\|^2 = \|\mathbf{B}^{-1/2}(f_i - f_j)\|^2 = \frac{1}{w_1} + \frac{1}{w_j}. \tag{4.10}$$

Thus in this case the update rule (4.9) simplifies:

$$\begin{aligned} x^{k+1} &\overset{b=0,\mathbf{A}=\mathbf{Q},(4.9)}{=} x^k - \omega \frac{\mathbf{Q}_{e:}x^k}{\|\mathbf{B}^{-1/2}\mathbf{Q}_{e:}^\top\|^2}\mathbf{B}^{-1}\mathbf{Q}_{e:}^\top \\ &\overset{(4.10)}{=} x^k - \omega \frac{\mathbf{Q}_{e:}x^k}{\frac{1}{w_1} + \frac{1}{w_j}}\mathbf{B}^{-1}\mathbf{Q}_{e:}^\top \\ &= x^k - \frac{\omega(x_i^k - x_j^k)}{\frac{1}{w_i} + \frac{1}{w_j}}\left(\frac{1}{w_i}f_i - \frac{1}{w_j}f_j\right). \end{aligned} \tag{4.11}$$

From (4.11) it can be easily seen that only the values of coordinates i and j update their values. These coordinates correspond to the private values x_i^k and x_j^k of the nodes of the selected edge $e = (i, j)$. In particular the values of x_i^k and x_j^k are updated as follows:

$$x_i^{k+1} = \left(1 - \omega\frac{w_j}{w_j + w_i}\right)x_i^k + \omega\frac{w_j}{w_j + w_i}x_j^k \quad \text{and} \quad x_j^{k+1} = \omega\frac{w_i}{w_j + w_i}x_i^k + \left(1 - \omega\frac{w_i}{w_j + w_i}\right)x_j^k. \tag{4.12}$$

Remark 11. *In the special case that* $\mathbf{B} = r\mathbf{I}$ *where* $r \in \mathbb{R}$ *(we solve the standard average consensus problem) the update of the two nodes is simplified to*

$$x_i^{k+1} = \left(1 - \frac{\omega}{2}\right) x_i^k + \frac{\omega}{2} x_j^k \quad \text{and} \quad x_j^{k+1} = \frac{\omega}{2} x_i^k + \left(1 - \frac{\omega}{2}\right) x_j^k.$$

If we further select $\omega = 1$ *then this becomes:*

$$x_i^{k+1} = x_j^{k+1} = \frac{x_i^k + x_j^k}{2}, \tag{4.13}$$

which is the update of the standard pairwise randomized gossip algorithm first presented and analyzed in [16].

AC system with Laplacian matrix L

The AC system takes the form $\mathbf{L}x = 0$, where matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the Laplacian matrix of the network. In this case, each row of the matrix corresponds to a node. Using the definition of the Laplacian, we have that $\mathbf{L}_{i:}^\top = d_i f_i - \sum_{j \in \mathcal{N}_i} f_j$, where f_i, f_j are unit coordinate vectors in $\mathbb{R}^n$ and d_i is the degree of node $i \in \mathcal{V}$.

Thus, by letting $\mathbf{B} = \mathbf{Diag}(w_1, w_2, \ldots, w_n)$ to be the diagonal matrix of the weights we obtain:

$$\|\mathbf{B}^{-1/2}\mathbf{L}_{i:}^\top\|^2 = \left\|\mathbf{B}^{-1/2}(d_i f_i - \sum_{j \in \mathcal{N}_i} f_j)\right\|^2 = \frac{d_i^2}{w_i} + \sum_{j \in \mathcal{N}_i} \frac{1}{w_j}. \tag{4.14}$$

In this case, the update rule (4.9) simplifies to:

$$\begin{aligned} x^{k+1} &\overset{b=0,\mathbf{A}=\mathbf{L},(4.9)}{=} x^k - \omega \frac{\mathbf{L}_{i:}x^k}{\|\mathbf{B}^{-1/2}\mathbf{L}_{i:}^\top\|_2^2}\mathbf{B}^{-1}\mathbf{L}_{i:}^\top \\ &\overset{(4.14)}{=} x^k - \omega \frac{\mathbf{L}_{i:}x^k}{\frac{d_i^2}{w_i} + \sum_{j \in \mathcal{N}_i} \frac{1}{w_j}}\mathbf{B}^{-1}\mathbf{L}_{i:}^\top \\ &= x^k - \frac{\omega\left(d_i x_i^k - \sum_{j \in \mathcal{N}_i} x_j^k\right)}{\frac{d_i^2}{w_i} + \sum_{j \in \mathcal{N}_i} \frac{1}{w_j}} \left(\frac{d_i}{w_i} f_i - \sum_{j \in \mathcal{N}_i} \frac{1}{w_j} f_j\right). \end{aligned} \tag{4.15}$$

From (4.15), it is clear that only coordinates $\{i\} \cup \mathcal{N}_i$ update their values. All the other coordinates remain unchanged. In particular, the value of the selected node i (coordinate i) is updated as follows:

$$x_i^{k+1} = x_i^k - \frac{\omega\left(d_i x_i^k - \sum_{j \in \mathcal{N}_i} x_j^k\right)}{\frac{d_i^2}{w_i} + \sum_{j \in \mathcal{N}_i} \frac{1}{w_j}} \frac{d_i}{w_i}, \tag{4.16}$$

while the values of its neighbors $j \in \mathcal{N}_i$ are updated as:

$$x_j^{k+1} = x_j^k + \frac{\omega\left(d_i x_i^k - \sum_{\ell \in \mathcal{N}_i} x_\ell^k\right)}{\frac{d_i^2}{w_i} + \sum_{\ell \in \mathcal{N}_i} \frac{1}{w_\ell}} \frac{1}{w_j}. \tag{4.17}$$

Remark 12. *Let* $\omega = 1$ *and* $\mathbf{B} = r\mathbf{I}$ *where* $r \in \mathbb{R}$ *then the selected nodes update their values as follows:*

$$x_i^{k+1} = \frac{\sum_{\ell \in \{i \cup \mathcal{N}_i\}} x_\ell^k}{d_i + 1} \quad \text{and} \quad x_j^{k+1} = x_j^k + \frac{(d_i x_i^k - \sum_{\ell \in \mathcal{N}_i} x_\ell^k)}{d_i^2 + d_i}. \tag{4.18}$$

That is, the selected node i *updates its value to the average of its neighbors and itself, while all*

the nodes $j \in \mathcal{N}_i$ update their values using the current value of node i and all nodes in $\mathcal{N}_i$.

In a wireless network, to implement such an update, node i would first broadcast its current value to all of its neighbors. Then it would need to receive values from each neighbor to compute the sums over $\mathcal{N}_i$, after which node i would broadcast the sum to all neighbors (since there may be two neighbors $j_1, j_2 \in \mathcal{N}_i$ for which $(j_1, j_2) \notin \mathcal{E}$). In a wired network, using standard concepts from the MPI library, such an update rule could be implemented efficiently by defining a process group consisting of $\{i\} \cup \mathcal{N}_i$, and performing one `Broadcast` in this group from i (containing x_i) followed by an `AllReduce` to sum x_ℓ over $\ell \in \mathcal{N}_i$. Note that the terms involving diagonal entries of $\mathbf{B}$ and the degrees d_i could be sent once, cached, and reused throughout the algorithm execution to reduce communication overhead.

Details on complexity results

Recall that the convergence rate of the sketch and project method (Algorithm 8) is equivalent to:

$$\rho := 1 - \omega(2-\omega)\lambda_{\min}^{+}(\mathbf{W}),$$

where $\omega \in (0,2)$ and $\mathbf{W} = \mathbf{B}^{-1/2}\mathbf{A}^\top \mathbb{E}[\mathbf{H}]\mathbf{A}\mathbf{B}^{-1/2}$ (from Theorem 30). In this subsection we explain how the convergence rate of the scaled RK method (4.9) is modified for different choices of the main parameters of the method.

Let us choose $\omega = 1$ (no over-relaxation). In this case, the rate is simplified to $\rho = 1 - \lambda_{\min}^{+}$.

Note that the different ways of modeling the problem (AC system) and the selection of the main parameters (weight matrix $\mathbf{B}$ and distribution $\mathcal{D}$) determine the convergence rate of the method through the spectrum of matrix $\mathbf{W}$.

Recall that in the k^{th} iterate of the scaled RK method (4.9) a random vector $\mathbf{S}_k = e_i$ is chosen with probability $p_i > 0$. For convenience, let us choose[3]:

$$p_i = \frac{\|\mathbf{B}^{-1/2}\mathbf{A}_{i:}^\top\|^2}{\|\mathbf{B}^{-1/2}\mathbf{A}^\top\|_F^2}. \tag{4.19}$$

Then we have that:

$$\begin{aligned}
\mathbb{E}[\mathbf{H}] &= \mathbb{E}[\mathbf{S}(\mathbf{S}^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S})^\dagger\mathbf{S}^\top] \\
&= \sum_{i=1}^m p_i \frac{e_i e_i^\top}{e_i^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top e_i} = \sum_{i=1}^m p_i \frac{e_i e_i^\top}{\|\mathbf{A}_{i:}^\top\|_{\mathbf{B}^{-1}}^2} = \sum_{i=1}^m p_i \frac{e_i e_i^\top}{\|\mathbf{B}^{-1/2}\mathbf{A}_{i:}^\top\|^2} \\
&\overset{(4.19)}{=} \sum_{i=1}^m \frac{e_i e_i^\top}{\|\mathbf{B}^{-1/2}\mathbf{A}^\top\|_F^2} = \frac{1}{\|\mathbf{B}^{-1/2}\mathbf{A}^\top\|_F^2}\mathbf{I},
\end{aligned} \tag{4.20}$$

and

$$\mathbf{W} \overset{(4.1),(4.20)}{=} \frac{\mathbf{B}^{-1/2}\mathbf{A}^\top\mathbf{A}\mathbf{B}^{-1/2}}{\|\mathbf{B}^{-1/2}\mathbf{A}^\top\|_F^2}. \tag{4.21}$$

Incidence Matrix: Let us choose the AC system to be the one with the incidence matrix $\mathbf{A} = \mathbf{Q}$. Then $\|\mathbf{B}^{-1/2}\mathbf{Q}^\top\|_F^2 = \sum_{i=1}^n \frac{d_i}{b_i}$ and we obtain

$$\mathbf{W} \overset{\mathbf{A}=\mathbf{Q},(4.21)}{=} \frac{\mathbf{B}^{-1/2}\mathbf{L}\mathbf{B}^{-1/2}}{\|\mathbf{B}^{-1/2}\mathbf{Q}^\top\|_F^2} = \frac{\mathbf{B}^{-1/2}\mathbf{L}\mathbf{B}^{-1/2}}{\sum_{i=1}^n \frac{d_i}{\mathbf{B}_{ii}}}.$$

[3]Similar probabilities have been chosen in [73] for the convergence of the standard RK method ($\mathbf{B} = \mathbf{I}$). The distribution $\mathcal{D}$ of the matrices $\mathbf{S}$ used in equation (4.19) is common in the area of randomized iterative methods for linear systems and is used to simplify the analysis and the expressions of the convergence rates. For more choices of distributions we refer the interested reader to [73]. It is worth to mention that the probability distribution that optimizes the convergence rate of the RK and other projection methods can be expressed as the solution to a convex semidefinite program [73, 30].

If we further have $\mathbf{B} = \mathbf{D}$, then $\mathbf{W} = \frac{\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}}{n}$ and the convergence rate simplifies to:

$$\rho = 1 - \frac{\lambda_{\min}^{+}\left(\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}\right)}{n} = 1 - \frac{\lambda_{\min}^{+}\left(\mathbf{L}^{sym}\right)}{n}.$$

If $\mathbf{B} = r\mathbf{I}$ where $r \in \mathbb{R}$ (solve the standard average consensus problem), then $\mathbf{W} = \frac{\mathbf{L}}{\|\mathbf{Q}\|_F^2} = \frac{\mathbf{L}}{\sum_{i=1}^n d_i} = \frac{\mathbf{L}}{2m}$ and the convergence rate simplifies to

$$\rho = 1 - \frac{\lambda_{\min}^{+}(\mathbf{L})}{2m} = 1 - \frac{\alpha(\mathcal{G})}{2m}, \tag{4.22}$$

which is exactly the same convergence rate of the pairwise gossip algorithm presented in [16]. This was expected, since the gossip protocol in this case works exactly the same as the one proposed in [16], see equation (4.13).

Laplacian Matrix: If we choose to formulate the AC system using the Laplacian matrix $\mathbf{L}$, that is $\mathbf{A} = \mathbf{L}$, then $\|\mathbf{B}^{-1/2}\mathbf{L}^\top\|_F^2 = \sum_{i=1}^n \frac{d_i(d_i+1)}{\mathbf{B}_{ii}}$ and we have:

$$\mathbf{W} \overset{\mathbf{A}=\mathbf{L},(4.21)}{=} \frac{\mathbf{B}^{-1/2}\mathbf{L}^\top\mathbf{L}\mathbf{B}^{-1/2}}{\sum_{i=1}^n \frac{d_i(d_i+1)}{\mathbf{B}_{ii}}}.$$

If $\mathbf{B} = \mathbf{D}$, then the convergence rate simplifies to:

$$\rho = 1 - \frac{\lambda_{\min}^{+}(\mathbf{D}^{-1/2}\mathbf{L}^\top\mathbf{L}\mathbf{D}^{-1/2})}{\sum_{i=1}^n (d_i+1)} = 1 - \frac{\lambda_{\min}^{+}\left(\mathbf{D}^{-1/2}\mathbf{L}^2\mathbf{D}^{-1/2}\right)}{n + \sum_{i=1}^n d_i} \overset{\sum_{i=1}^n d_i = 2m}{=} 1 - \frac{\lambda_{\min}^{+}\left(\mathbf{D}^{-1/2}\mathbf{L}^2\mathbf{D}^{-1/2}\right)}{n+2m}.$$

If $\mathbf{B} = r\mathbf{I}$, where $r \in \mathbb{R}$, then $\mathbf{W} = \frac{\mathbf{L}^2}{\|\mathbf{L}\|_F^2} = \frac{\mathbf{L}^2}{\sum_{i=1}^n d_i(d_i+1)}$ and the convergence rate simplifies to

$$\rho = 1 - \frac{\lambda_{\min}^{+}(\mathbf{L}^2)}{\sum_{i=1}^n d_i(d_i+1)} = 1 - \frac{\alpha(\mathcal{G})^2}{\sum_{i=1}^n d_i(d_i+1)}.$$

4.3.4 Block gossip algorithms

Up to this point we focused on the basic connections between the convergence analysis of the sketch and project methods and the literature of randomized gossip algorithms. We show how specific variants of the randomized Kaczmarz method (RK) can be interpreted as gossip algorithms for solving the weighted and standard average consensus problems.

In this part we extend the previously described methods to their block variants related to randomized block Kaczmarz (RBK) method (4.3). In particular, in each step of Algorithm 8, the random matrix $\mathbf{S}$ is selected to be a random column submatrix of the $m \times m$ identity matrix corresponding to columns indexed by a random subset $C \subseteq [m]$. That is, $\mathbf{S} = \mathbf{I}_{:C}$, where a set $C \subseteq [m]$ is chosen in each iteration independently, with probability $p_C \geq 0$ (see equation (4.3)). Note that in the special case that set C is a singleton with probability 1 the algorithm is simply the randomized Kaczmarz method of the previous section.

To keep things simple, we assume that $\mathbf{B} = \mathbf{I}$ (standard average consensus, without weights) and choose the stepsize $\omega = 1$. In the next section, we will describe gossip algorithms with heavy ball momentum and explain in detail how the gossip interpretation of RBK change in the more general case of $\omega \in (0, 2)$.

Similar to the previous subsections, we formulate the consensus problem using either $\mathbf{A} = \mathbf{Q}$ or $\mathbf{A} = \mathbf{L}$ as the matrix in the AC system. In this setup, the iterative process of Algorithm 8 has the form:

$$x^{k+1} \overset{(4.3),(4.8)}{=} x^k - \mathbf{A}^\top\mathbf{I}_{:C}(\mathbf{I}_{:C}^\top\mathbf{A}\mathbf{A}^\top\mathbf{I}_{:C})^\dagger\mathbf{I}_{:C}^\top\mathbf{A}x^k = x^k - \mathbf{A}_{C:}^\top(\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger\mathbf{A}_{C:}x^k, \tag{4.23}$$

which, as explained in the introduction, can be equivalently written as:

$$x^{k+1} = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \{\|x - x^k\|^2 \ : \ \mathbf{I}_{:C}^\top \mathbf{A} x = 0\}. \tag{4.24}$$

Essentially in each step of this method the next iterate is evaluated to be the projection of the current iterate x^k onto the solution set of a row subsystem of $\mathbf{A}x = 0$.

AC system with Incidence Matrix: In the case that $\mathbf{A} = \mathbf{Q}$ the selected rows correspond to a random subset $C \subseteq \mathcal{E}$ of selected edges. While (4.23) may seem to be a complicated algebraic (resp. variational) characterization of the method, due to our choice of $\mathbf{A} = \mathbf{Q}$ we have the following result which gives a natural interpretation of RBK as a gossip algorithm (see also Figure 4.1).

Theorem 34 (RBK as Gossip algorithm: RBKG)**.** *Consider the AC system with the constraints being expressed using the Incidence matrix* $\mathbf{Q}$*. Then each iteration of RBK (Algorithm* (4.23)*) works as gossip algorithm as follows:*

1. *Select a random set of edges* $C \subseteq \mathcal{E}$*,*
2. *Form subgraph* $\mathcal{G}_k$ *of* $\mathcal{G}$ *from the selected edges*
3. *For each connected component of* $\mathcal{G}_k$*, replace node values with their average.*

Proof. See Section 4.9.2. □

Using the convergence result of general Theorem 30 and the form of matrix $\mathbf{W}$ (recall that in this case we assume $\mathbf{B} = \mathbf{I}$, $\mathbf{S} = \mathbf{I}_{:C} \sim \mathcal{D}$ and $\omega = 1$), we obtain the following complexity for the algorithm:

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \left[1 - \lambda_{\min}^+ \left(\mathbb{E}\left[\mathbf{Q}_{C:}^\top (\mathbf{Q}_{C:}\mathbf{Q}_{C:}^\top)^\dagger \mathbf{Q}_{C:}\right]\right)\right]^k \|x^0 - x^*\|^2. \tag{4.25}$$

For more details on the above convergence rate of randomized block Kaczmarz method with meaningfully bounds on the rate in a more general setting we suggest the papers [134, 135].

There is a very closed relationship between the gossip interpretation of RBK explained in Theorem 34 and several existing randomized gossip algorithms that in each step update the values of more than two nodes. For example the *path averaging* algorithm porposed in [8] is a special case of RBK, when set C is restricted to correspond to a path of vertices. That is, in path averaging, in each iteration a path of nodes is selected and the nodes that belong to it update their values to their exact average. A different example is the recently proposed clique gossiping [110] where the network is already divided into cliques and through a random procedure a clique is activated and the nodes of it update their values to their exact average. In [16] a synchronous variant of gossip algorithm is presented where in each step multiple node pairs communicate exactly at the same time with the restriction that these simultaneously active node pairs are disjoint.

It is easy to see that all of the above algorithms can be cast as special cases of RBK if the distribution $\mathcal{D}$ of the random matrices is chosen carefully to be over random matrices $\mathbf{S}$ (column sub-matrices of Identity) that update specific set of edges in each iteration. As a result our general convergence analysis can recover the complexity results proposed in the above works.

Finally, as we mentioned, in the special case in which set C is always a singleton, Algorithm (4.23) reduces to the standard randomized Kaczmarz method. This means that only a random edge is selected in each iteration and the nodes incident with this edge replace their local values with their average. This is the pairwise gossip algorithm of Boyd er al. [16] presented in equation (4.13). Theorem 34 extends this interpretation to the case of the RBK method.

AC system with Laplacian Matrix: For this choice of AC system the update is more complicated. To simplify the way that the block variant work as gossip we make an extra assumption. We assume that the selected rows of the constraint $\mathbf{I}_{:C}^\top \mathbf{L} x = 0$ in update (4.24) have no-zero elements at different coordinates. This allows to have a direct extension of the

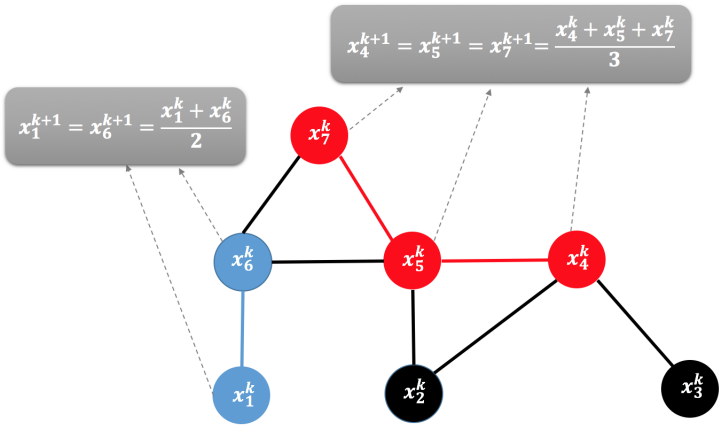


Figure 4.1: Example of how the RBK method works as gossip algorithm in case of AC system with Incidence matrix. In the presented network 3 edges are randomly selected and a subgraph of two connected components (blue and red) is formed. Then the nodes of each connected component update their private values to their average.

serial variant presented in Remark 12. Thus, in this setup, the RBK update rule (4.23) works as gossip algorithm as follows:

1. $|C|$ nodes are activated (with restriction that the nodes are not neighbors and they do not share common neighbors)

2. For each node $i \in C$ we have the following update:

$$x_i^{k+1} = \frac{\sum_{\ell \in \{i \cup \mathcal{N}_i\}} x_\ell^k}{d_i + 1} \quad \text{and} \quad x_j^{k+1} = x_j^k + \frac{\left(d_i x_i^k - \sum_{\ell \in \mathcal{N}_i} x_\ell^k\right)}{d_i^2 + d_i}. \tag{4.26}$$

The above update rule can be seen as a parallel variant of update (4.18). Similar to the convergence in the case of Incidence matrix, the RBK for solving the AC system with a Laplacian matrix converges to x^* with the following rate (using result of Theorem 30):

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \left[1 - \lambda_{\min}^+ \left(\mathbb{E}\left[\mathbf{L}_{C:}^\top (\mathbf{L}_{C:}\mathbf{L}_{C:}^\top)^\dagger \mathbf{L}_{C:}\right]\right)\right]^k \|x^0 - x^*\|^2.$$

4.4 Faster and Provably Accelerated Randomized Gossip Algorithms

The main goal in the design of gossip protocols is for the computation and communication to be done as quickly and efficiently as possible. In this section, our focus is precisely this. We design randomized gossip protocols which converge to consensus fast with provable accelerated linear rates. To the best of our knowledge, the proposed protocols are the first randomized gossip algorithms that converge to consensus with an accelerated linear rate.

In particular, we present novel protocols for solving the average consensus problem where in each step all nodes of the network update their values but only a subset of them exchange their private values. The protocols are inspired from the recently developed accelerated variants of randomized Kaczmarz-type methods for solving consistent linear systems where the addition of momentum terms on top of the sketch and project update rule provides better theoretical and practical performance.

In the area of optimization algorithms, there are two popular ways to accelerate an algorithm using momentum. The first one is using the Polyak's heavy ball momentum [156] and the second one is using the theoretically much better understood momentum introduced by Nesterov [138, 140]. Both momentum approaches have been recently proposed and analyzed to improve the performance of randomized iterative methods for solving linear systems.

To simplify the presentation, the accelerated algorithms and their convergence rates are presented for solving the standard average consensus problem ($\mathbf{B} = \mathbf{I}$). Using a similar approach as in the previous section, the update rules and the convergence rates can be easily modified to solve the more general weighted average consensus problem. For the protocols in this section we use the incidence matrix $\mathbf{A} = \mathbf{Q}$ or its normalized variant to formulate the AC system.

4.4.1 Gossip algorithms with heavy ball momentum

In Chapter 2 of this thesis we have analyzed heavy ball momentum variants of several algorithms for solving the stochastic optimization problem (1.6) and as we explained the best approximation problem (1.22). In this section we revisit Algorithm 1 of Chapter 2 and we focus on its sketch and project viewpoint. In particular, we explain how it works as gossip algorithm when is applied to the AC system with the incidence matrix.

Sketch and project with heavy ball momentum

The sketch and project method with heavy ball momentum is formally presented in the following algorithm.

Algorithm 9 Sketch and Project with Heavy Ball Momentum

Parameters: Distribution $\mathcal{D}$ from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$; momentum parameter β.
Initialize: $x^0, x^1 \in \mathbb{R}^n$
for $k = 1, 2, \ldots$ **do**
  Draw a fresh $\mathbf{S}_k \sim \mathcal{D}$
  Set

$$x^{k+1} = x^k - \omega \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x^k - b) + \beta(x^k - x^{k-1}). \tag{4.27}$$

end for
Output: The last iterate x^k

Using, $\mathbf{B} = \mathbf{I}$ and the same choice of distribution $\mathcal{D}$ as in equations (4.2) and (4.3) we can now obtain momentum variants of the RK and RBK as special case of the above algorithm as follows:

- RK with momentum (mRK):

$$x^{k+1} = x^k - \omega \frac{\mathbf{A}_{i:}x^k - b_i}{\|\mathbf{A}_{i:}\|^2}\mathbf{A}_{i:}^\top + \beta(x^k - x^{k-1}). \tag{4.28}$$

- RBK with momentum (mRBK):

$$x^{k+1} = x^k - \omega \mathbf{A}_{C:}^\top (\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger (\mathbf{A}_{C:}x^k - b_C) + \beta(x^k - x^{k-1}). \tag{4.29}$$

For more details on the convergence analysis of Algorithm 9 see Section 2.3 and recall that in our setting the sketch and project update rule is identical to the SGD (Chapter 1). As a result Algorithm 9 is identical to Algorithm 1 (mSGD/mSN/mSPP).

Having presented Algorithm 9, let us now describe its behavior as a randomized gossip protocol when applied to the AC system $\mathbf{A}x = 0$ with $\mathbf{A} = \mathbf{Q} \in |\mathcal{E}| \times n$ (incidence matrix of the network).

Note that since $b = 0$ (from the AC system definition), the update rule (4.27) of Algorithm 9 is simplified to (and by having $\mathbf{B} = \mathbf{I}$):

$$x^{k+1} = \left[\mathbf{I} - \omega \mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top \mathbf{A}\right] x^k + \beta(x^k - x^{k-1}). \tag{4.30}$$

In the rest of this section we focus on two special cases of (4.30): RK with heavy ball momentum (equation (4.28) with $b_i = 0$) and RBK with heavy ball momentum (equation (4.29) with $b_C = 0$).

Algorithm 10 mRK: Randomized Kaczmarz with momentum as a gossip algorithm

Parameters: Distribution $\mathcal{D}$ from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$; heavy ball/momentum parameter β.
Initialize: $x^0, x^1 \in \mathbb{R}^n$
for $k = 1, 2, \ldots$ **do**
    Pick an edge $e = (i, j)$ following the distribution $\mathcal{D}$
    The values of the nodes are updated as follows:
- Node i: $x_i^{k+1} = \frac{2-\omega}{2} x_i^k + \frac{\omega}{2} x_j^k + \beta(x_i^k - x_i^{k-1})$
- Node j: $x_j^{k+1} = \frac{2-\omega}{2} x_j^k + \frac{\omega}{2} x_i^k + \beta(x_j^k - x_j^{k-1})$
- Any other node ℓ: $x_\ell^{k+1} = x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1})$

end for
Output: The last iterate x^k

Randomized Kaczmarz gossip with heavy ball momentum

As we have seen in previous section when the standard RK is applied to solve the AC system $\mathbf{Q}x = 0$, one can recover the famous pairwise gossip algorithm [16]. Algorithm 10 describes how a relaxed variant of randomized Kaczmarz with heavy ball momentum ($0 < \omega < 2$ and $0 \leq \beta < 1$) behaves as a gossip algorithm. See also Figure (4.2) for a graphical illustration of the method.

Remark 13. *In the special case that $\beta = 0$ (zero momentum) only the two nodes of edge $e = (i, j)$ update their values. In this case the two selected nodes do not update their values to their exact average but to a convex combination that depends on the stepsize $\omega \in (0, 2)$. To obtain the pairwise gossip algorithm of [16], one should further choose $\omega = 1$.*

Distributed Nature of the Algorithm: Here we highlight a few ways to implement mRK in a distributed fashion.

- *Pairwise broadcast gossip:* In this protocol each node $i \in \mathcal{V}$ of the network $\mathcal{G}$ has a clock that ticks at the times of a rate 1 Poisson process. The inter-tick times are exponentially distributed, independent across nodes, and independent across time. This is equivalent to a global clock ticking at a rate n Poisson process which wakes up an edge of the network at random. In particular, in this implementation mRK works as follows: In the k^{th} iteration (time slot) the clock of node i ticks and node i randomly contact one of its neighbors and simultaneously broadcast a signal to inform the nodes of the whole network that is updating (this signal does not contain any private information of node i). The two nodes (i, j) share their information and update their private values following the update rule of Algorithm 10 while all the other nodes update their values using their own information. In each iteration only one pair of nodes exchange their private values.

- *Synchronous pairwise gossip:* In this protocol a single global clock is available to all nodes. The time is assumed to be slotted commonly across nodes and in each time slot only a pair of nodes of the network is randomly activated and exchange their information following the update rule of Algorithm 10. The remaining not activated nodes update their values using their own last two private values. Note that this implementation of mRK comes with the disadvantage that it requires a central entity which in each step requires to choose the activated pair of nodes[4].

- *Asynchronous pairwise gossip with common counter:* Note that the update rule of the selected pair of nodes (i, j) in Algorithm 10 can be rewritten as follows:

$$x_i^{k+1} = x_i^k + \beta(x_i^k - x_i^{k-1}) + \frac{\omega}{2}(x_j^k - x_i^k),$$

[4]We speculate that a completely distributed synchronous gossip algorithm that finds pair of nodes in a distributed manner without any additional computational burden can be design following the same procedure proposed in Section III.C of [16].

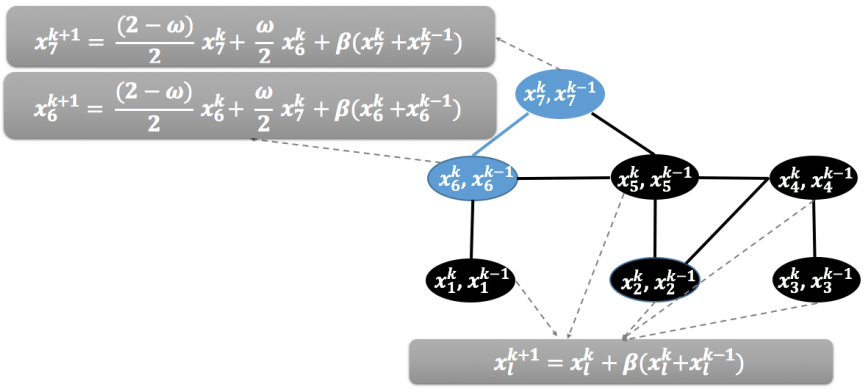


Figure 4.2: Example of how mRK works as gossip algorithm. In the presented network the edge that connects nodes 6 and 7 is randomly selected. The pair of nodes exchange their information and update their values following the update rule of the Algorithm 10 while the rest of the nodes, $\ell \in [5]$, update their values using only their own previous private values.

$$x_j^{k+1} = x_j^k + \beta(x_j^k - x_j^{k-1}) + \frac{\omega}{2}(x_i^k - x_j^k).$$

In particular observe that the first part of the above expressions $x_i^k + \beta(x_i^k - x_i^{k-1})$ (for the case of node i) is exactly the same with the update rule of the non activate nodes at k^{th} iterate (check step 5 of Algorithm 10) . Thus, if we assume that all nodes share a common counter that keeps track of the current iteration count and that each node $i \in \mathcal{V}$ remembers the iteration counter k_i of when it was last activated, then step 5 of Algorithm 10 takes the form:

- $x_i^{k+1} = i_k \left[x_i^k + \beta(x_i^k - x_i^{k-1})\right] + \frac{\omega}{2}(x_j^k - x_i^k)$,
- $x_j^{k+1} = j_k \left[x_j^k + \beta(x_j^k - x_j^{k-1})\right] + \frac{\omega}{2}(x_i^k - x_j^k)$,
- $k_i = k_j = k + 1$,
- Any other node ℓ: $x_\ell^{k+1} = x_\ell^k$,

where $i_k = k - k_i$ ($j_k = k - k_j$) denotes the number of iterations between the current iterate and the last time that the i^{th} (j^{th}) node is activated. In this implementation only a pair of nodes communicate and update their values in each iteration (thus the justification of asynchronous), however it requires the nodes to share a common counter that keeps track the current iteration count in order to be able to compute the value of $i_k = k - k_i$.

Connections with existing fast randomized gossip algorithms

In the randomized gossip literature there is one particular method closely related to our approach. It was first proposed in [19] and its analysis under strong conditions was presented in [106]. In this work local memory is exploited by installing shift registers at each agent. In particular we are interested in the case of two registers where the first stores the agent's current value and the second the agent's value before the latest update. The algorithm can be described as follows. Suppose that edge $e = (i, j)$ is chosen at time k. Then,

- Node i: $x_i^{k+1} = \omega(\frac{x_i^k + x_j^k}{2}) + (1 - \omega)x_i^{k-1}$,
- Node j: $x_i^{k+1} = \omega(\frac{x_i^k + x_j^k}{2}) + (1 - \omega)x_j^{k-1}$,
- Any other node ℓ: $x_\ell^{k+1} = x_\ell^k$,

where $\omega \in [1,2)$. The method was analyzed in [106] under a strong assumption on the probabilities of choosing the pair of nodes, that as the authors mentioned, is unrealistic in practical scenarios, and for networks like the random geometric graphs. At this point we should highlight that the results presented in Chapter 2 hold for essentially any distribution $\mathcal{D}$ [5] and as a result in the proposed gossip variants with heavy ball momentum such problem cannot occur.

Note that, in the special case that we choose $\beta = \omega - 1$ in the update rule of Algorithm 10 is simplified to:

- Node i: $x_i^{k+1} = \omega(\frac{x_i^k + x_j^k}{2}) + (1-\omega)x_i^{k-1}$,
- Node j: $x_i^{k+1} = \omega(\frac{x_i^k + x_j^k}{2}) + (1-\omega)x_j^{k-1}$,
- Any other node ℓ: $x_\ell^{k+1} = \omega x_\ell^k + (1-\omega)x_\ell^{k-1}$.

Recall that in order to apply Theorem 8, we need to assume that $0 < \omega < 2$ and $\beta = \omega - 1 \geq 0$ which also means that $\omega \in [1,2)$. Thus for $\omega \in [1,2)$ and momentum parameter $\beta = \omega - 1$ it is easy to see that our approach is very similar to the shift-register algorithm. Both methods update the selected pair of nodes in the same way. However, in Algorithm 10 the not selected nodes of the network do not remain idle but instead update their values using their own previous information.

By defining the momentum matrix $\mathbf{M} = \mathbf{Diag}(\beta_1, \beta_2, \ldots, \beta_n)$, the above closely related algorithms can be expressed, in vector form, as:

$$x^{k+1} = x^k - \frac{\omega}{2}(x_i^k - x_j^k)(e_i - e_j) + \mathbf{M}(x^k - x^{k-1}). \tag{4.31}$$

In particular, in mRK every diagonal element of matrix $\mathbf{M}$ is equal to $\omega - 1$, while in the algorithm of [19, 106] all the diagonal elements are zeros except the two values that correspond to nodes i and j that are equal to $\beta_i = \beta_j = \omega - 1$.

Remark 14. *The shift register algorithm of [106] and Algorithm 10 of this work can be seen as the two limit cases of the update rule* (4.31)*. As we mentioned, the shift register method [106] uses only two non-zero diagonal elements in* $\mathbf{M}$*, while our method has a full diagonal. We believe that further methods can be developed in the future by exploring the cases where more than two but not all elements of the diagonal matrix* $\mathbf{M}$ *are non-zero. It might be possible to obtain better convergence if one carefully chooses these values based on the network topology. We leave this as an open problem for future research.*

Randomized block Kaczmarz gossip with heavy ball momentum

Recall that Theorem 34 explains how RBK (with no momentum and no relaxation) can be interpreted as a gossip algorithm. In this subsection by using this result we explain how relaxed RBK with momentum works. Note that the update rule of RBK with momentum can be rewritten as follows:

$$x^{k+1} \overset{(4.30),(4.29)}{=} \omega\left(\mathbf{I} - \mathbf{A}_{C:}^\top(\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger\mathbf{A}_{C:}\right)x^k + (1-\omega)x^k + \beta(x^k - x^{k-1}), \tag{4.32}$$

and recall that $x^{k+1} = \left(\mathbf{I} - \mathbf{A}_{C:}^\top(\mathbf{A}_{C:}\mathbf{A}_{C:}^\top)^\dagger\mathbf{A}_{C:}\right)x^k$ is the update rule of the standard RBK (4.23).

Thus, in analogy to the standard RBK, in the k^{th} step, a random set of edges is selected and $q \leq n$ connected components are formed as a result. This includes the connected components that belong to both sub-graph $\mathcal{G}_k$ and also the singleton connected components (nodes outside the $\mathcal{G}_k$). Let us define the set of the nodes that belong in the $r \in [q]$ connected component at the k^{th} step $\mathcal{V}_r^k$, such that $\mathcal{V} = \cup_{r\in[q]}\mathcal{V}_r^k$ and $|\mathcal{V}| = \sum_{r=1}^q |\mathcal{V}_r^k|$ for any $k > 0$.

Using the update rule (4.32), Algorithm 11 shows how mRBK is updating the private values of the nodes of the network (see also Figure 4.3 for the graphical interpretation).

Note that in the update rule of mRBK the nodes that are not attached to a selected edge (do not belong in the sub-graph $\mathcal{G}_k$) update their values via $x_\ell^{k+1} = x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1})$. By

[5] The only restriction is the exactness condition to be satisfied. See Theorem 8.

Algorithm 11 mRBK: Randomized Block Kaczmarz Gossip with momentum

Parameters: Distribution $\mathcal{D}$ from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$; heavy ball/momentum parameter β.
Initialize: $x^0, x^1 \in \mathbb{R}^n$
for $k = 1, 2, \ldots$ **do**
    Select a random set of edges $\mathcal{S} \subseteq \mathcal{E}$
    Form subgraph $\mathcal{G}_k$ of $\mathcal{G}$ from the selected edges
    Node values are updated as follows:

- For each connected component $\mathcal{V}_r^k$ of $\mathcal{G}_k$, replace the values of its nodes with:

$$x_i^{k+1} = \omega \frac{\sum_{j \in \mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} + (1-\omega)x_i^k + \beta(x_i^k - x_i^{k-1}). \tag{4.33}$$

- Any other node ℓ: $x_\ell^{k+1} = x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1})$

end for
Output: The last iterate x^k

considering these nodes as singleton connected components their update rule is exactly the same with the nodes of sub-graph $\mathcal{G}_k$. This is easy to see as follows:

$$\begin{aligned} x_\ell^{k+1} &\overset{(4.33)}{=} \omega \frac{\sum_{j \in \mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} + (1-\omega)x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1}) \\ &\overset{|\mathcal{V}_r^k|=1}{=} \omega x_\ell^k + (1-\omega)x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1}) \\ &= x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1}). \end{aligned} \tag{4.34}$$

Remark 15. *In the special case that only one edge is selected in each iteration ($\mathbf{S}_k \in \mathbb{R}^{m \times 1}$) the update rule of mRBK is simplified to the update rule of mRK. In this case the sub-graph $\mathcal{G}_k$ is the pair of the two selected edges.*

Remark 16. *In previous section we explained how several existing gossip protocols for solving the average consensus problem are special cases of the RBK (Theorem 34). For example two gossip algorithms that can be cast as special cases of the standard RBK are the path averaging proposed in [8] and the clique gossiping [110]. In path averaging, in each iteration a path of nodes is selected and its nodes update their values to their exact average ($\omega = 1$). In clique gossiping, the network is already divided into cliques and through a random procedure a clique is activated and the nodes of it update their values to their exact average ($\omega = 1$). Since mRBK contains the standard RBK as a special case (when $\beta = 0$), we expect that these special protocols can also be accelerated with the addition of momentum parameter $\beta \in (0, 1)$.*

Mass preservation

One of the key properties of some of the most efficient randomized gossip algorithms is mass preservation. That is, the sum (and as a result the average) of the private values of the nodes remains fixed during the iterative procedure ($\sum_{i=1}^n x_i^k = \sum_{i=1}^n x_i^0, \quad \forall k \geq 1$). The original pairwise gossip algorithm proposed in [16] satisfied the mass preservation property, while exisiting fast gossip algorithms [19, 106] preserving a scaled sum. In this subsection we show that mRK and mRBK gossip protocols presented above satisfy the mass preservation property. In particular, we prove mass preservation for the case of the block randomized gossip protocol (Algorithm 11) with momentum. This is sufficient since the randomized Kaczmarz gossip with momentum (mRK), Algorithm 10 can be cast as special case.

Theorem 35. *Assume that $x^0 = x^1 = c$. That is, the two registers of each node have the same initial value. Then for the Algorithms 10 and 11 we have $\sum_{i=1}^n x_i^k = \sum_{i=1}^n c_i$ for any $k \geq 0$ and as a result, $\frac{1}{n}\sum_{i=1}^n x_i^k = \bar{c}$.*

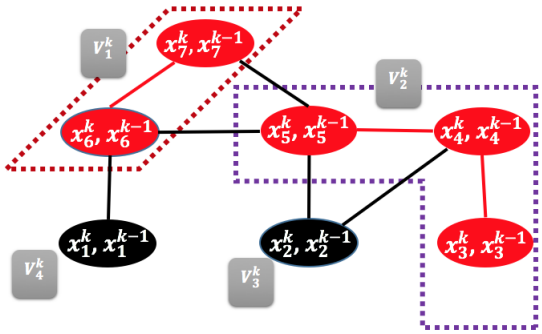


Figure 4.3: Example of how the mRBK method works as gossip algorithm. In the presented network the red edges are randomly chosen in the k^{th} iteration, and they form subgraph $\mathcal{G}_k$ and four connected component. In this figure V_1^k and V_2^k are the two connected components that belong in the subgraph $\mathcal{G}_k$ while V_3^k and V_4^k are the singleton connected components. Then the nodes update their values by communicate with the other nodes of their connected component using the update rule (4.33). For example the node number 5 that belongs in the connected component V_2^k will update its value using the values of node 4 and 3 that also belong in the same component as follows: $x_5^{k+1} = \omega \frac{x_3^k + x_4^k + x_5^k}{3} + (1-\omega)x_5^k + \beta(x_5^k - x_5^{k-1})$.

Proof. We prove the result for the more general Algorithm 11. Assume that in the k^{th} step of the method q connected components are formed. Let the set of the nodes of each connected component be $\mathcal{V}_r^k$ so that $\mathcal{V} = \cup_{r=\{1,2,\dots q\}} \mathcal{V}_r^k$ and $|\mathcal{V}| = \sum_{r=1}^q |\mathcal{V}_r^k|$ for any $k > 0$. Thus:

$$\sum_{i=1}^n x_i^{k+1} = \sum_{i\in\mathcal{V}_1^k} x_i^{k+1} + \cdots + \sum_{i\in\mathcal{V}_q^k} x_i^{k+1}. \tag{4.35}$$

Let us first focus, without loss of generality, on connected component $r \in [q]$ and simplify the expression for the sum of its nodes:

$$\begin{aligned}
\sum_{i\in\mathcal{V}_r^k} x_i^{k+1} &\overset{(4.33)}{=} \sum_{i\in\mathcal{V}_r^k} \omega \frac{\sum_{j\in\mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} + (1-\omega)\sum_{i\in\mathcal{V}_r^k} x_i^k + \beta \sum_{i\in\mathcal{V}_r^k} (x_i^k - x_i^{k-1}) \\
&= |\mathcal{V}_r^k| \frac{\omega \sum_{j\in\mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} + (1-\omega)\sum_{i\in\mathcal{V}_r^k} x_i^k + \beta \sum_{i\in\mathcal{V}_r^k} (x_i^k - x_i^{k-1}) \\
&= (1+\beta)\sum_{i\in\mathcal{V}_r^k} x_i^k - \beta \sum_{i\in\mathcal{V}_r^k} x_i^{k-1}.
\end{aligned} \tag{4.36}$$

By substituting this for all $r \in [q]$ into the right hand side of (4.35) and from the fact that $\mathcal{V} = \cup_{r\in[q]} \mathcal{V}_r^k$, we obtain:

$$\sum_{i=1}^n x_i^{k+1} = (1+\beta)\sum_{i=1}^n x_i^k - \beta \sum_{i=1}^n x_i^{k-1}.$$

Since $x^0 = x^1$, we have $\sum_{i=1}^n x_i^0 = \sum_{i=1}^n x_i^1$, and as a result $\sum_{i=1}^n x_i^k = \sum_{i=1}^n x_i^0$ for all $k \geq 0$. □

4.4.2 Provably accelerated randomized gossip algorithms

In this subsection we focus on one specific case of the Sketch and Project framework, the RK method (4.2). We present two accelerated variants of RK where the Nesterov's momentum is used, for solving consistent linear systems and we describe their theoretical convergence results. Based on these methods we propose two provably accelerated gossip protocols, along with some remarks on their implementation.

Accelerated Kaczmarz methods using Nesterov's momentum

There are two different but very similar ways to provably accelerate the randomized Kaczmarz method using Nesterov's acceleration. The first paper that proves *asymptotic convergence* with an accelerated linear rate is [107]. The proof technique is similar to the framework developed by Nesterov in [139] for the acceleration of coordinate descent methods. In [192, 72] a modified version for the selection of the parameters was proposed and a *non-asymptotic* accelerated linear rate was established. In Algorithm 12, pseudocode of the Accelerated Kaczmarz method (AccRK) is presented where both variants can be cast as special cases, by choosing the parameters with the correct way.

Algorithm 12 Accelerated Randomized Kaczmarz Method (AccRK)

Data: Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$; vector $b \in \mathbb{R}^m$
Choose $x^0 \in \mathbb{R}^n$ and set $v^0 = x^0$
Parameters: Evaluate the sequences of the scalars $\alpha_k, \beta_k, \gamma_k$ following one of two possible options.
for $k = 0, 1, 2, \ldots, K$ **do**
    $y^k = \alpha_k v^k + (1-\alpha_k)x^k$
    Draw a fresh sample $i_k \in [m]$ with equal probability
    $x^{k+1} = y^k - \frac{\mathbf{A}_{i_k:}y^k - b_{i_k}}{\|\mathbf{A}_{i_k:}\|^2}\mathbf{A}_{i_k:}^\top$.
    $v^{k+1} = \beta_k v^k + (1-\beta_k)y^k - \gamma_k \frac{\mathbf{A}_{i_k:}y^k - b_{i_k}}{\|\mathbf{A}_{i_k:}\|^2}\mathbf{A}_{i_k:}^\top$.
end for

There are two options for selecting the parameters of the AccRK for solving consistent linear systems with normalized matrices, which we describe next.

1. From [107]: Choose $\lambda \in [0, \lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})]$ and set $\gamma_{-1} = 0$. Generate the sequence $\{\gamma_k : k = 0, 1, \ldots, K+1\}$ by choosing γ_k to be the largest root of

$$\gamma_k^2 - \frac{\gamma_k}{m} = (1 - \frac{\gamma_k}{\lambda}m)\gamma_{k-1}^2,$$

   and generate the sequences $\{\alpha_k : k = 0, 1, \ldots, K+1\}$ and $\{\beta_k : k = 0, 1, \ldots, K+1\}$ by setting

$$\alpha_k = \frac{m - \gamma_k\lambda}{\gamma_k(m^2-\lambda)}, \quad \beta_k = 1 - \frac{\gamma_k\lambda}{m}.$$

2. From [72]: Let

$$\nu = \max_{u\in \text{Range}(\mathbf{A}^\top)} \frac{u^\top \left[\sum_{i=1}^m \mathbf{A}_{i:}^\top \mathbf{A}_{i:}(\mathbf{A}^\top\mathbf{A})^\dagger \mathbf{A}_{i:}^\top \mathbf{A}_{i:}\right] u}{u^\top \frac{\mathbf{A}^\top\mathbf{A}}{m} u}. \tag{4.37}$$

   Choose the three sequences to be fixed constants as follows: $\beta_k = \beta = 1 - \sqrt{\frac{\lambda_{\min}^+(\mathbf{W})}{\nu}}$, $\gamma_k = \gamma = \sqrt{\frac{1}{\lambda_{\min}^+(\mathbf{W})\nu}}$, $\alpha_k = \alpha = \frac{1}{1+\gamma\nu} \in (0,1)$ where $\mathbf{W} = \frac{\mathbf{A}^\top\mathbf{A}}{m}$.

Theoretical guarantees of AccRK

The two variants (Option 1 and Option 2) of AccRK are closely related, however their convergence analyses are different. Below we present the theoretical guarantees of the two options as presented in [107] and [72].

Theorem 36 ([107])**.** *Let $\{x^k\}_{k=0}^\infty$ be the sequence of random iterates produced by Algorithm 12 with the Option 1 for the parameters. Let $\mathbf{A}$ be normalized matrix and let $\lambda \in [0, \lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})]$. Set $\sigma_1 = 1 + \frac{\sqrt{\lambda}}{2m}$ and $\sigma_2 = 1 - \frac{\sqrt{\lambda}}{2m}$. Then for any $k \geq 0$ we have that:*

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \frac{4\lambda}{(\sigma_1^k - \sigma_2^k)^2}\|x^0 - x^*\|^2_{(\mathbf{A}^\top\mathbf{A})^\dagger}.$$

Corollary 37 ([107]). *Note that as $k \to \infty$, we have that $\sigma_2^k \to 0$. This means that the decrease of the right hand side is governed mainly by the behavior of the term σ_1 in the denominator and as a result the method converge* asymptotically *with a decrease factor per iteration: $\sigma_1^{-2} = (1 + \frac{\sqrt{\lambda}}{2m})^{-2} \approx 1 - \frac{\sqrt{\lambda}}{m}$. That is, as $k \to \infty$:*

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \left(1 - \sqrt{\lambda}/m\right)^k 4\lambda \|x^0 - x^*\|^2_{(\mathbf{A}^\top \mathbf{A})^\dagger}$$

Thus, by choosing $\lambda = \lambda_{\min}^+$ and for the case that $\lambda_{\min}^+$ is small, Algorithm 12 will have significantly faster convergence rate than RK. Note that the above convergence results hold only for normalized matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$, that is matrices that have $\|\mathbf{A}_{i:}\| = 1$ for any $i \in m$.

Using Corollary 37, Algorithm 12 with the first choice of the parameters converges linearly with rate $\left(1 - \sqrt{\lambda}/m\right)$. That is, it requires $O\left(m/\sqrt{\lambda}\log(1/\epsilon)\right)$ iterations to obtain accuracy $\mathbb{E}[\|x^k - x^*\|^2] \leq \epsilon 4\lambda \|x^0 - x^*\|^2_{(\mathbf{A}^\top \mathbf{A})^\dagger}$.

Theorem 38 ([72]). *Let $\mathbf{W} = \frac{\mathbf{A}^\top \mathbf{A}}{m}$ and let assume exactness*[6]*. Let $\{x^k, y^k, v^k\}$ be the iterates of Algorithm 12 with the Option 2 for the parameters. Then*

$$\Psi^k \leq \left(1 - \sqrt{\lambda_{\min}^+(\mathbf{W})/\nu}\right)^k \Psi^0,$$

where $\Psi^k = \mathbb{E}\left[\|v^k - x^*\|^2_{\mathbf{W}^\dagger} + \frac{1}{\mu}\|x^k - x^*\|^2\right]$.

The above result implies that Algorithm 12 converges linearly with rate $1 - \sqrt{\lambda_{\min}^+(\mathbf{W})/\nu}$, which translates to a total of $\mathcal{O}\left(\sqrt{\nu/\lambda_{\min}^+(\mathbf{W})}\log(1/\epsilon)\right)$ iterations to bring the quantity Ψ^k below $\epsilon > 0$. It can be shown that $1 \leq \nu \leq 1/\lambda_{\min}^+(\mathbf{W})$, (Lemma 2 in [72]) where ν is as defined in (4.37). Thus, $\sqrt{\frac{1}{\lambda_{\min}^+(\mathbf{W})}} \leq \sqrt{\frac{\nu}{\lambda_{\min}^+(\mathbf{W})}} \leq \frac{1}{\lambda_{\min}^+(\mathbf{W})}$, which means that the rate of AccRK (Option 2) is always better than that of the RK with unit stepsize which is equal to $\mathcal{O}\left(\frac{1}{\lambda_{\min}^+(\mathbf{W})}\log(1/\epsilon)\right)$ (see Theorem 30).

In [72], Theorem 38 has been proposed for solving more general consistent linear systems (the matrix $\mathbf{A}$ of the system is not assumed to be normalized). In this case $\mathbf{W} = \mathbb{E}[\mathbf{Z}]$ and the parameter ν is slightly more complicated than the one of equation (4.37). We refer the interested reader to [72] for more details.

Comparison of the convergence rates: Before describe the distributed nature of the AccRK and explain how it can be interpreted as a gossip algorithm, let us compare the convergence rates of the two options of the parameters for the case of general normalized consistent linear systems ($\|\mathbf{A}_{i:}\| = 1$ for any $i \in [m]$).

Using Theorems 36 and 38, it is clear that the iteration complexity of AccRK is

$$O\left(\frac{m}{\sqrt{\lambda}}\log(1/\epsilon)\right) \overset{\lambda=\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}{=} O\left(\frac{m}{\sqrt{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}}\log(1/\epsilon)\right), \tag{4.38}$$

and

$$\mathcal{O}\left(\sqrt{\frac{\nu m}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}}\log(1/\epsilon)\right), \tag{4.39}$$

for the Option 1 and Option 2 for the parameters, respectively.

In the following derivation we compare the iteration complexity of the two methods.

[6]Note that in this setting $\mathbf{B} = \mathbf{I}$, which means that $\mathbf{W} = \mathbb{E}[\mathbf{Z}]$, and the exactness assumption takes the form $\text{Null}(\mathbf{W}) = \text{Null}(\mathbf{A})$.

Lemma 39. *Let matrices $\mathbf{C} \in \mathbb{R}^{n\times n}$ and $\mathbf{C}_i \in \mathbb{R}^{n\times n}$ where $i \in [m]$ be positive semidefinite, and satisfying $\sum_{i=1}^m \mathbf{C}_i = \mathbf{C}$. Then*

$$\sum_{i=1}^m \mathbf{C}_i \mathbf{C}^\dagger \mathbf{C}_i \preceq \mathbf{C}.$$

Proof. From the definition of the matrices it holds that $\mathbf{C}_i \preceq \mathbf{C}$ for any $i \in [m]$. Using the properties of Moore-Penrose pseudoinverse, this implies that

$$\mathbf{C}_i^\dagger \succeq \mathbf{C}^\dagger. \tag{4.40}$$

Therefore

$$\mathbf{C}_i = \mathbf{C}_i \mathbf{C}_i^\dagger \mathbf{C}_i \overset{(4.40)}{\succeq} \mathbf{C}_i \mathbf{C}^\dagger \mathbf{C}_i. \tag{4.41}$$

From the definition of the matrices by taking the sum over all $i \in [m]$ we obtain:

$$\mathbf{C} = \sum_{i=1}^m \mathbf{C}_i \overset{(4.41)}{\succeq} \sum_{i=1}^m \mathbf{C}_i \mathbf{C}^\dagger \mathbf{C}_i,$$

which completes the proof. □

Let us now choose $\mathbf{C}_i = \mathbf{A}_{i:}^\top \mathbf{A}_{i:}$ and $\mathbf{C} = \mathbf{A}^\top \mathbf{A}$. Note that from their definition the matrices are positive semidefinite and satisfy $\sum_{i=1}^m \mathbf{A}_{i:}^\top \mathbf{A}_{i:} = \mathbf{A}^\top \mathbf{A}$. Using Lemma 39 it is clear that:

$$\sum_{i=1}^m \mathbf{A}_{i:}^\top \mathbf{A}_{i:} (\mathbf{A}^\top \mathbf{A})^\dagger \mathbf{A}_{i:}^\top \mathbf{A}_{i:} \preceq \mathbf{A}^\top \mathbf{A},$$

or in other words, for any vector $v \notin \text{Null}(\mathbf{A})$ we set the inequality

$$\frac{v^\top \left[\sum_{i=1}^m \mathbf{A}_{i:}^\top \mathbf{A}_{i:} (\mathbf{A}^\top \mathbf{A})^\dagger \mathbf{A}_{i:}^\top \mathbf{A}_{i:}\right] v}{v^\top [\mathbf{A}^\top \mathbf{A}] v} \leq 1.$$

Multiplying both sides by m, we set:

$$\frac{v^\top \left[\sum_{i=1}^m \mathbf{A}_{i:}^\top \mathbf{A}_{i:} (\mathbf{A}^\top \mathbf{A})^\dagger \mathbf{A}_{i:}^\top \mathbf{A}_{i:}\right] v}{v^\top [\frac{\mathbf{A}^\top \mathbf{A}}{m}] v} \leq m.$$

Using the above derivation, it is clear from the definition of the parameter ν (4.37), that $\nu \leq m$. By combining our finding with the bounds already obtained in [72] for the parameter ν, we have that:

$$1 \leq \nu \leq \min\left\{m, \frac{1}{\lambda_{\min}^+(\mathbf{W})}\right\}. \tag{4.42}$$

Thus, by comparing the two iteration complexities of equations (4.38) and (4.39) it is clear that Option 2 for the parameters [72] is always faster in theory than Option 1 [107]. To the best of our knowledge, such comparison of the two choices of the parameters for the AccRK was never presented before.

Accelerated randomized gossip algorithms

Having presented the complexity analysis guarantees of AccRK for solving consistent linear systems with normalized matrices, let us now explain how the two options of AccRK behave as gossip algorithms when they are used to solve the linear system $\mathbf{A}x = 0$ where $\mathbf{A} \in \mathbb{R}^{|\mathcal{E}|\times n}$ is the normalized incidence matrix of the network. That is, each row $e = (i, j)$ of $\mathbf{A}$ can be represented as $(\mathbf{A}_{e:})^\top = \frac{1}{\sqrt{2}}(e_i - e_j)$ where e_i (resp.e_j) is the i^{th} (resp. j^{th}) unit coordinate vector in $\mathbb{R}^n$.

By using this particular linear system, the expression $\frac{\mathbf{A}_{i:}y^k - b_i}{\|\mathbf{A}_{i:}\|^2}\mathbf{A}_{i:}^\top$ that appears in steps 7

and 8 of AccRK takes the following form when the row $e = (i, j) \in \mathcal{E}$ is sampled:

$$\frac{\mathbf{A}_{e:}y^k - b_i}{\|\mathbf{A}_{e:}\|^2}\mathbf{A}_{e:}^\top \overset{b=0}{=} \frac{\mathbf{A}_{e:}y^k}{\|\mathbf{A}_{e:}\|^2}\mathbf{A}_{e:}^\top \overset{\text{form of A}}{=} \frac{y_i^k - y_j^k}{2}(e_i - e_j).$$

Recall that with $\mathbf{L}$ we denote the Laplacian matrix of the network. For solving the above AC system (see Definition 31), the standard RK requires $\mathcal{O}\left(\left(\frac{2m}{\lambda_{\min}^+(\mathbf{L})}\right)\log(1/\epsilon)\right)$ iterations to achieve expected accuracy $\epsilon > 0$. To understand the acceleration in the gossip framework this should be compared to the

$$\mathcal{O}\left(m\sqrt{\frac{2}{\lambda_{\min}^+(\mathbf{L})}}\log(1/\epsilon)\right)$$

of AccRK (Option 1) and the

$$\mathcal{O}\left(\sqrt{\frac{2m\nu}{\lambda_{\min}^+(\mathbf{L})}}\log(1/\epsilon)\right)$$

of AccRK (Option 2).

Algorithm 13 describes in a single framework how the two variants of AccRK of Section 4.4.2 behave as gossip algorithms when are used to solve the above linear system. Note that each node $\ell \in \mathcal{V}$ of the network has two local registers to save the quantities v_ℓ^k and x_ℓ^k. In each step using these two values every node $\ell \in \mathcal{V}$ of the network (activated or not) computes the quantity $y_\ell^k = \alpha_k v_\ell^k + (1 - \alpha_k)x_\ell^k$. Then in the k^{th} iteration the activated nodes i and j of the randomly selected edge $e = (i, j)$ exchange their values y_i^k and y_j^k and update the values of x_i^k, x_j^k and v_i^k, v_j^k as shown in Algorithm 13. The rest of the nodes use only their own y_ℓ^k to update the values of v_i^k and x_i^k without communicate with any other node.

The parameter $\lambda_{\min}^+(\mathbf{L})$ can be estimated by all nodes in a decentralized manner using the method described in [24]. In order to implement this algorithm, we assume that all nodes have synchronized clocks and that they know the rate at which gossip updates are performed, so that inactive nodes also update their local values. This may not be feasible in all applications, but when it is possible (e.g., if nodes are equipped with inexpensive GPS receivers, or have reliable clocks) then they can benefit from the significant speedup achieved.

Algorithm 13 Accelerated Randomized Gossip Algorithm (AccGossip)

Data: Matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ (normalized incidence matrix); vector $b = 0 \in \mathbb{R}^m$
Choose $x^0 \in \mathbb{R}^n$ and set $v^0 = x^0$
Parameters: Evaluate the sequences of the scalars $\alpha_k, \beta_k, \gamma_k$ following one of two possible options.
for $k = 0, 1, 2, \ldots, K$ **do**
  Each node $\ell \in \mathcal{V}$ evaluate $y_\ell^k = \alpha_k v_\ell^k + (1 - \alpha_k)x_\ell^k$.
  Pick an edge $e = (i, j)$ uniformly at random.
  Then the nodes update their values as follows:

- The selected node i and node j:

$$x_i^{k+1} = x_j^{k+1} = (y_i^k + y_j^k)/2$$

$$v_i^{k+1} = \beta_k v_i^k + (1 - \beta_k)y_i^k - \gamma_k(y_i^k - y_j^k)/2$$

$$v_j^{k+1} = \beta_k v_j^k + (1 - \beta_k)y_j^k - \gamma_k(y_j^k - y_i^k)/2$$

- Any other node $\ell \in \mathcal{V}$:

$$x_\ell^{k+1} = y_\ell^k \quad , \quad v_\ell^{k+1} = \beta_k v_\ell^k + (1 - \beta_k)y_\ell^k$$

end for

4.5 Dual Randomized Gossip Algorithms

An important tool in optimization literature is duality. In our setting, instead of solving the original minimization problem (primal problem) one may try to develop dual in nature methods that have as a goal to directly solve the dual maximization problem. Then the primal solution can be recovered through the use of optimality conditions and the development of an affine mapping between the two spaces (primal and dual).

In this section, using existing dual methods and the connection already established between the two areas of research (methods for linear systems and gossip algorithms), we present a different viewpoint that allows the development of novel dual randomized gossip algorithms.

Without loss of generality we focus on the case of $\mathbf{B} = \mathbf{I}$ (no weighted average consensus). For simplicity, we formulate the AC system as the one with the incidence matrix of the network ($\mathbf{A} = \mathbf{Q}$) and focus on presenting the distributed nature of dual randomized gossip algorithms with no momentum. While we focus only on no-momentum protocols, we note that accelerated variants of the dual methods could be easily obtained using tools from Section 4.4.

4.5.1 Dual problem and SDSA

As we have already presented in Section 1.4, the Lagrangian dual of the best approximation problem (1.22) is the (bounded) unconstrained concave quadratic maximization problem:

$$\max_{y\in\mathbb{R}^m} D(y) := (b - \mathbf{A}x^0)^\top y - \frac{1}{2}\|\mathbf{A}^\top y\|^2_{\mathbf{B}^{-1}}. \tag{4.43}$$

A direct method for solving the dual problem is Stochastic Dual Subspace Accent (SDSA), a randomized iterative algorithm first proposed in [74], which updates the dual vectors y^k as follows:

$$y^{k+1} = y^k + \omega\mathbf{S}_k\left(\mathbf{S}_k^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_k\right)^\dagger\mathbf{S}_k^\top\left(b - \mathbf{A}(x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k)\right). \tag{4.44}$$

In Section 1.4 we showed that the iterates $\{x^k\}_{k\geq 0}$ of the sketch and project method (Algorithm 8) can be arised as affine images of the iterates $\{y^k\}_{k\geq 0}$ of the dual method (4.44) through the mapping:

$$x^k = \phi(y^k) = x^0 + \mathbf{B}^{-1}\mathbf{A}^\top y^k, \tag{4.45}$$

and we provided a proof for the linear convergence of SDSA (see Theorem 6). Recall that SDSA and the sketch and project method (Algorithm 8) converge to a solution of the dual problem and primal problem, respectively, with exactly the same convergence rate.

Let us choose $\mathbf{B} = \mathbf{I}$. In the special case that the random matrix $\mathbf{S}_k$ is chosen randomly from the set of unit coordinate/basis vectors in $\mathbb{R}^m$, the dual method (4.44) is the randomized coordinate descent [104, 166], and the corresponding primal method is RK (4.2). More generally, if $\mathbf{S}_k$ is a random column submatrix of the $m \times m$ identity matrix, the dual method is the randomized Newton method [162], and the corresponding primal method is RBK (4.3). Next we shall describe the more general block case in more detail.

4.5.2 Randomized Newton method as a dual gossip algorithm

In this subsection we bring a new insight into the randomized gossip framework by presenting how the dual iterative process that is associated to RBK method solves the AC problem with $\mathbf{A} = \mathbf{Q}$ (incidence matrix). Recall that the right hand side of the linear system is $b = 0$. For simplicity, we focus on the case of $\mathbf{B} = \mathbf{I}$ and $\omega = 1$.

Under this setting ($\mathbf{A} = \mathbf{Q}$, $\mathbf{B} = \mathbf{I}$ and $\omega = 1$) the dual iterative process (4.44) takes the form:

$$y^{k+1} = y^k - \mathbf{I}_{C:}(\mathbf{I}_{C:}^\top\mathbf{Q}\mathbf{Q}^\top\mathbf{I}_{C:})^\dagger\mathbf{Q}(x^0 + \mathbf{Q}^\top y^k), \tag{4.46}$$

and from Theorem 6 converges to a solution of the dual problem as follows:

$$\mathbb{E}\left[D(y^*) - D(y^k)\right] \leq \left[1 - \lambda^+_{\min}\left(\mathbb{E}\left[\mathbf{Q}_{C:}^\top(\mathbf{Q}_{C:}\mathbf{Q}_{C:}^\top)^\dagger\mathbf{Q}_{C:}\right]\right)\right]^k\left[D(y^*) - D(y^0)\right].$$

Note that the convergence rate is exactly the same with the rate of the RBK under the same assumptions (see (4.25)).

This algorithm is a randomized variant of the Newton method applied to the problem of maximizing the quadratic function $D(y)$ defined in (4.43). Indeed, in each iteration we perform the update $y^{k+1} = y^k + \mathbf{I}_{C:}\lambda^k$, where λ^k is chosen greedily so that $D(y^{k+1})$ is maximized. In doing so, we invert a random principal submatrix of the Hessian of D, whence the name.

Randomized Newton Method (RNM) was first proposed by Qu et al. [162]. RNM was first analyzed as an algorithm for minimizing *smooth strongly convex functions*. In [74] it was also extended to the case of a *smooth but weakly convex quadratics*. This method was not previously associated with any gossip algorithm.

The most important distinction of RNM compared to existing gossip algorithms is that it operates with values that are associated to the *edges* of the network. To the best of our knowledge, it is the first *randomized dual gossip method*. In particular, instead of iterating over values stored at the nodes, RNM uses these values to update "dual weights" $y^k \in \mathbb{R}^m$ that correspond to the edges $\mathcal{E}$ of the network. However, deterministic dual distributed averaging algorithms were proposed before [164, 64]. Edge-based methods have also been proposed before; in particular in [195] an asynchronous distributed ADMM algorithm presented for solving the more general consensus optimization problem with convex functions.

Natural Interpretation. In iteration k, *RNM* (Algorithm (4.46)) executes the following steps: 1) Select a random set of edges $\mathcal{S}_k \subseteq \mathcal{E}$, 2) Form a subgraph $\mathcal{G}_k$ of $\mathcal{G}$ from the selected edges, 3) The values of the edges in each connected component of $\mathcal{G}_k$ are updated: their new values are a linear combination of the private values of the nodes belonging to the connected component and of the adjacent edges of their connected components. (see also example of Figure 4.4).

Dual Variables as Advice. The weights y^k of the edges have a natural interpretation as *advice* that each selected node receives from the network in order to update its value (to one that will eventually converge to the desired average).

Consider RNM performing the k^{th} iteration and let $\mathcal{V}_r$ denote the set of nodes of the selected connected component that node i belongs to. Then, from Theorem 34 we know that $x_i^{k+1} = \sum_{i\in\mathcal{V}_r} x_i^k/|\mathcal{V}_r|$. Hence, by using (4.45), we obtain the following identity:

$$(\mathbf{A}^\top y^{k+1})_i = \tfrac{1}{|\mathcal{V}_r|}\textstyle\sum_{i\in\mathcal{V}_r}(c_i + (\mathbf{A}^\top y^k)_i) - c_i. \tag{4.47}$$

Thus in each step $(\mathbf{A}^\top y^{k+1})_i$ represents the term (advice) that must be added to the initial value c_i of node i in order to update its value to the average of the values of the nodes of the connected component i belongs to.

Importance of the dual perspective: It was shown in [162] that when RNM (and as a result, RBK, through the affine mapping (4.45)) is viewed as a family of methods indexed by the size $\tau = |\mathcal{S}|$ (we choose $\mathcal{S}$ of fixed size in the experiments), then $\tau \to 1/(1-\rho)$, where ρ is defined in (4.5), decreases *superlinearly* fast in τ. That is, as τ increases by some factor, the iteration complexity drops by a factor that is at least as large. Through preliminary numerical experiments in Section 4.7.2 we experimentally show that this is true for the case of AC systems as well.

4.6 Further Connections Between Methods for Solving Linear Systems and Gossip Algorithms

In this section we highlight some further interesting connections between linear systems solvers and gossip protocols for average consensus:

- **Eavesdrop gossip as special case of Kaczmarz-Motzkin method.** In [193] greedy gossip with eavesdropping (GGE), a novel randomized gossip algorithm for distributed computation of the average consensus problem was proposed and analyzed. In particular it was shown that that greedy updates of GGE lead to rapid convergence. In this protocol, the greedy updates are made possible by exploiting the broadcast nature of wireless communications. During the operation of GGE, when a node decides to gossip, instead of

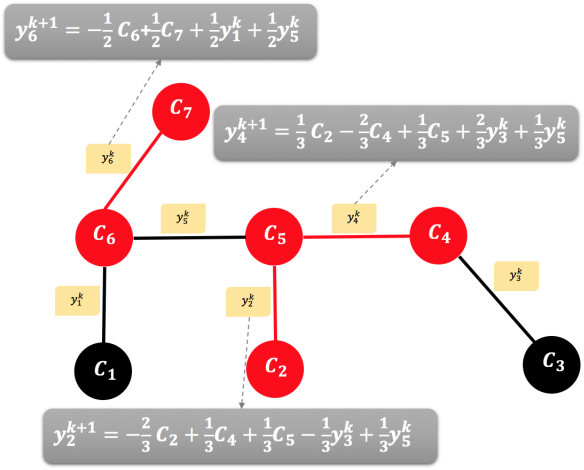


Figure 4.4: Example of how the RNM method works as gossip algorithm. In this specific case 3 edges are selected and form a sub-graph with two connected components. Then the values at the edges update their values using the private values of the nodes belonging to their connected component and the values associate to the adjacent edges of their connected components.

choosing one of its neighbors at random, it makes a greedy selection, choosing the node which has the value most different from its own. In particular the method behaves as follows:

At the k^{th} iteration of GGE, a node i_k is chosen uniformly at random from $[n]$. Then, i_k identifies a neighboring node $j_k \in \mathcal{N}_i$ satisfying:

$$j_k \in \max_{j \in \mathcal{N}_i} \left\{ \frac{1}{2}(x_i^k - x_j^k)^2 \right\}$$

which means that the selected node i_k identifies a neighbor that currently has the most different value from its own. This choice is possible because each node $i \in \mathcal{V}$ maintains not only its own local variable x_i^k, but also a copy of the current values at its neighbors x_j^k for $j \in \mathcal{N}_i$. In the case that node i_k has multiple neighbors whose values are all equally (and maximally) different from its current value, it chooses one of these neighbors at random. Then node i_k and j_k update their values to:

$$x_i^{k+1} = x_j^{k+1} = \frac{1}{2}(x_i^k + x_j^k).$$

In the area of randomized methods for solving large linear system there is one particular method, the Kaczmarz-Motzkin algorithm [32, 78] that can work as gossip algorithm with the same update as the GGE when is use to solve the homogeneous linear system with matrix the Incidence matrix of the network.

Update rule of Kaczmarz-Motzkin algorithm (KMA) [32, 78]:

1. Choose sample of d_k constraints, P_k, uniformly at random from among the rows of matrix $\mathbf{A}$.
2. From among these d_k constraints, choose $t_k = \text{argmax}_{i \in P_k} \mathbf{A}_{i:} x^k - b_i$.
3. Update the value: $x^{k+1} = x^k - \frac{\mathbf{A}_{t_k:} x^k - b_i}{\|\mathbf{A}_{t_k:}\|^2} \mathbf{A}_{t_k:}^\top$.

It is easy to verify that when the Kaczmarz-Motzkin algorithm is used for solving the AC system with $\mathbf{A} = \mathbf{Q}$ (incidence matrix) and in each step of the method the chosen constraints d_k of the linear system correspond to edges attached to one node it behaves exactly like the GGE. From numerical analysis viewpoint an easy way to choose the constraints d_k that are compatible to the desired edges is in each iteration to find the

indexes of the non-zeros of a uniformly at random selected column (node) and then select the rows corresponding to these indexes.

Therefore, since GGE [193] is a special case of the KMA (when the later applied to special AC system with Incidence matrix) it means that we can obtain the convergence rate of GGE by simply use the tight conergence analysis presented in [32, 78] [7]. In [193] it was mentioned that analyzing the convergence behavior of GGE is non-trivial and not an easy task. By establishing the above connection the convergence rates of GGE can be easily obtained as special case of the theorems presented in [32].

In Section 4.4 we presented provably accelerated variants of the pairwise gossip algorithm and of its block variant. Following the same approach one can easily develop accelerated variants of the GGE using the recently proposed analysis for the accelerated Kaczmarz-Motzkin algorithm presented in [126].

- **Inexact Sketch and Project Methods:**

  In Chapter 3, several inexact variants of the sketch and project method (8) have been proposed. As we have already mentioned the sketch and project method is a two step procedure algorithm where first the sketched system is formulated and then the last iterate x^k is *exactly* projected into the solution set of the sketched system. In Chapter 3, we replace the exact projection with an inexact variant and we suggest to run a different algorithm (this can be the sketch and project method itself) in the sketched system to obtain an approximate solution. It was shown that in terms of time the inexact updates can be faster than their exact variants.

  In the setting of randomized gossip algorithms for the AC system with Incidence matrix ($\mathbf{A} = \mathbf{Q}$) , $\mathbf{B} = \mathbf{I}$ and $\omega = 1$ a variant of the inexact sketch and project method will work as follows (similar to the update proved in Theorem 34):

  1. Select a random set of edges $C \subseteq \mathcal{E}$.
  2. Form subgraph $\mathcal{G}_k$ of $\mathcal{G}$ from the selected edges.
  3. Run the pairwise gossip algorithm of [16] (or any variant of the sketch and project method) on the subgraph $\mathcal{G}_k$ until an accuracy ϵ is achieved (reach a neighborhood of the exact average).

- **Non-randomized gossip algorithms as special cases of Kaczmarz methods:**

  In the gossip algorithms literature there are efficient protocols that are not randomized[127, 82, 108, 208]. Typically, in these algorithms the pairwise exchanges between nodes it happens in a deterministic, such as predefined cyclic, order. For example, T-periodic gossiping is a protocol which stipulates that each node must interact with each of its neighbours exactly once every T time units. It was shown that under suitable connectivity assumptions of the network $\mathcal{G}$, the T-periodic gossip sequence will converge at a rate determined by the magnitude of the second largest eigenvalue of the stochastic matrix determined by the sequence of pairwise exchanges which occurs over a period. It has been shown that if the underlying graph is a tree, the mentioned eigenvalue is constant for all possible T-periodic gossip protocols.

  In this work we focus only on randomized gossip protocols. However we speculate that the above non-randomized gossip algorithms would be able to express as special cases of popular non-randomized projection methods for solving linear systems [159, 144, 45]. Establishing connections like that is an interesting future direction of research and can possibly lead to the development of novel block and accelerated variants of many non-randomized gossip algorithms, similar to the protocols we present in Sections 4.3 and 4.4.

[7] Note that the convergence theorems of [32, 78] use $d_k = d$. However, with a small modification in the original proof the theorem can capture the case of different d_k.

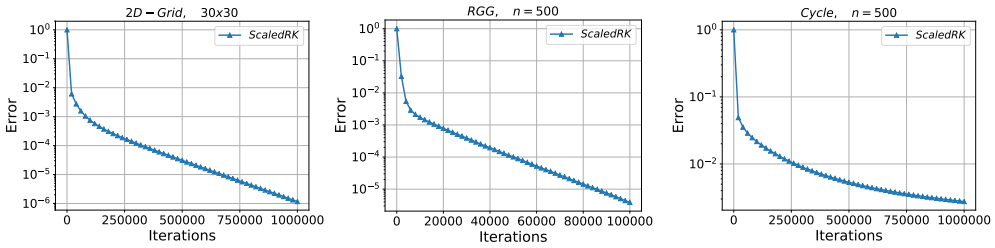


Figure 4.5: Performance of ScaledRK in a 2-dimension grid, random geometric graph (RGG) and a cycle graph for solving the weighted average consensus problem. The weight matrix is chosen to be $\mathbf{B} = \mathbf{D}$, the degree matrix of the network. The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

4.7 Numerical Evaluation

In this section, we empirically validate our theoretical results and evaluate the performance of the proposed randomized gossip algorithms. The section is divided into four main parts, in each of which we highlight a different aspect of our contributions.

In the first experiment, we numerically verify the linear convergence of the Scaled RK algorithm (see equation (4.9)) for solving the weighted average consensus problem presented in Section 4.3.1. In the second part, we explain the benefit of using block variants in the gossip protocols where more than two nodes update their values in each iteration (protocols presented in Section 4.3.4). In the third part, we explore the performance of the faster and provably accelerated gossip algorithms proposed in Section 4.4. In the last experiment, we numerically show that relaxed variants of the pairwise randomized gossip algorithm converge faster than the standard randomized pairwise gossip with unit stepsize (no relaxation). This gives a specific setting where the phenomenon of over-relaxation of iterative methods for solving linear systems is beneficial.

In the comparison of all gossip algorithms we use the relative error measure $\|x^k - x^*\|^2_{\mathbf{B}} / \|x^0 - x^*\|^2_{\mathbf{B}}$ where $x^0 = c \in \mathbb{R}^n$ is the starting vector of the values of the nodes and matrix $\mathbf{B}$ is the positive definite diagonal matrix with weights in its diagonal (recall that in the case of standard average consensus this can be simply $\mathbf{B} = \mathbf{I}$). Depending on the experiment, we choose the values of the starting vector $c \in \mathbb{R}^n$ to follow either a Gaussian distribution or uniform distribution or to be integer values such that $c_i = i \in \mathbb{R}$. In the plots, the horizontal axis represents the number of iterations except in the figures of subsection 4.7.2, where the horizontal axis represents the block size.

In our implementations we use three popular graph topologies from the area of wireless sensor networks. These are the cycle (ring graph), the 2-dimension grid and the random geometric graph (RGG) with radius $r = \sqrt{\log(n)/n}$. In all experiments we formulate the average consensus problem (or its weighted variant) using the incidence matrix. That is, $\mathbf{A} = \mathbf{Q}$ is used as the AC system. Code was written in Julia 0.6.3.

4.7.1 Convergence on weighted average consensus

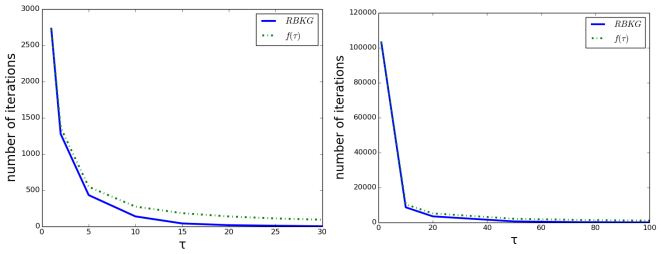
As we explained in Section 4.3, the sketch and project method (Algorithm 8) can solve the more general weighted AC problem. In this first experiment we numerically verify the linear convergence of the Scaled RK algorithm (4.9) for solving this problem in the case of $\mathbf{B} = \mathbf{D}$. That is, the matrix $\mathbf{B}$ of the weights is the degree matrix $\mathbf{D}$ of the graph ($\mathbf{B}_{ii} = d_i$, $\forall i \in [n]$). In this setting the exact update rule of the method is given in equation (4.12), where in order to have convergence to the weighted average the chosen nodes are required to share not only their private values but also their weight $\mathbf{B}_{ii}$ (in our experiment this is equal to the degree of the node d_i). In this experiment the starting vector of values $x^0 = c \in \mathbb{R}^n$ is a Gaussian vector. The linear convergence of the algorithm is clear in Figure 4.5.

4.7.2 Benefit of block variants

We devote this experiment to evaluate the performance of the randomized block gossip algorithms presented in Sections 4.3.4 and 4.5. In particular, we would like to highlight the benefit of using larger block size in the update rule of randomized Kaczmarz method and as a result through our established connection of the randomized pairwise gossip algorithm [16] (see equation (4.13)).

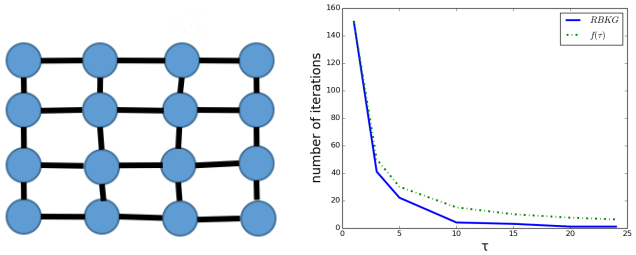
Recall that in Section 4.5 we show that both RBK and RNM converge to the solution of the primal and dual problems respectively with the same rate and that their iterates are related via a simple affine transform (4.45). In addition note that an interesting feature of the RNM [162], is that when the method viewed as algorithm indexed by the size $\tau = |C|$, it enjoys superlinear speedup in τ. That is, as τ (block size) increases by some factor, the iteration complexity drops by a factor that is at least as large (see Section 4.5.2). Since RBK and RNM share the same rates this property naturally holds for RBK as well.

We show that for a connected network $\mathcal{G}$, the complexity improves superlinearly in $\tau = |C|$, where C is chosen as a subset of $\mathcal{E}$ of size τ, uniformly at random (recall the in the update rule of RBK the random matrix is $\mathbf{S} = \mathbf{I}_{:C}$). Similar to the rest of this section in comparing the number of iterations for different values of τ, we use the relative error $\varepsilon = \|x^k - x^*\|^2 / \|x^0 - x^*\|^2$. We let $x_i^0 = c_i = i$ for each node $i \in \mathcal{V}$ (vector of integers). We run RBK until the relative error becomes smaller than 0.01. The blue solid line in the figures denotes the actual number of iterations (after running the code) needed in order to achieve $\varepsilon \leq 10^{-2}$ for different values of τ. The green dotted line represents the function $f(\tau) := \frac{\ell}{\tau}$, where ℓ is the number of iterations of RBK with $\tau = 1$ (i.e., the pairwise gossip algorithm). The green line depicts linear speedup; the fact that the blue line (obtained through experiments) is below the green line points to superlinear speedup. In this experiment we use the Cycle graph with $n = 30$ and $n = 100$ nodes (Figure 4.6) and the 4×4 two dimension grid graph (Figure 4.7). Note that, when $|C| = m$ the convergence rate of the method becomes $\rho = 0$ and as a result it converges in one step.



(a) Cycle, $n = 30$ (b) Cycle, $n = 100$

Figure 4.6: Superlinear speedup of RBK on cycle graphs.



(a) 2D-Grid, 4×4 (b) Speedup in τ

Figure 4.7: Superlinear speedup of RBK on a 4×4 two dimension grid graph.

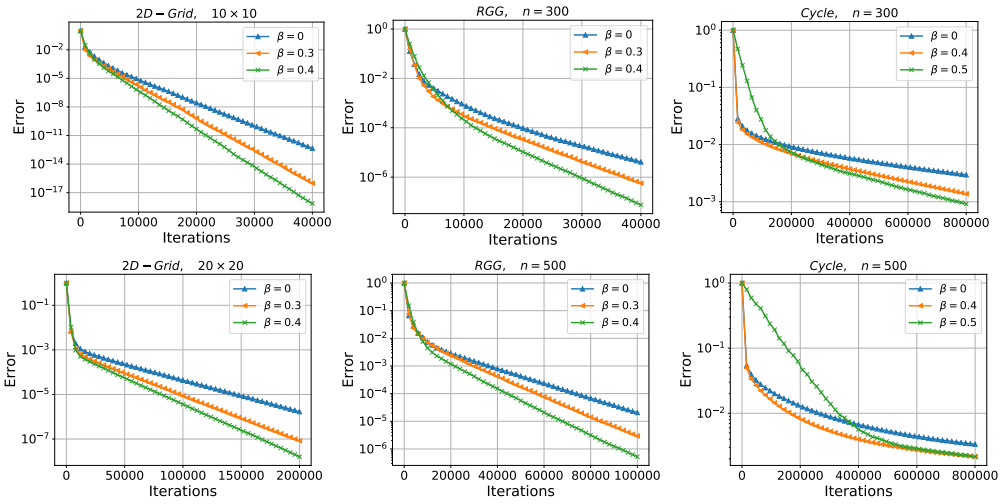


Figure 4.8: Performance of mRK for fixed step-size $\omega = 1$ and several momentum parameters β in a 2-dimension grid, random geometric graph (RGG) and a cycle graph. The choice $\beta = 0$ corresponds to the randomized pairwise gossip algorithm proposed in [16]. The starting vector $x^0 = c \in \mathbb{R}^n$ is a Gaussian vector. The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

4.7.3 Accelerated gossip algorithms

We devote this subsection to experimentally evaluate the performance of the proposed accelerated gossip algorithms: mRK (Algorithm 10), mRBK (Algorithm 11) and AccGossip with the two options of the parameters (Algorithm 13). In particular we perform four experiments. In the first two we focus on the performance of the mRK and how the choice of stepsize (relaxation parameter) ω and heavy ball momentum parameter β affect the performance of the method. In the next experiment we show that the addition of heavy ball momentum can be also beneficial for the performance of the block variant mRBK. In the last experiment we compare the standard pairwise gossip algorithm (baseline method) from [16], the mRK and the AccGossip and show that the probably accelerated gossip algorithm, AccGossip outperforms the other algorithms and converge as predicted from the theory with an accelerated linear rate.

Impact of momentum parameter on mRK

As we have already presented in the standard pairwise gossip algorithm (equation (4.13)) the two selected nodes that exchange information update their values to their exact average while all the other nodes remain idle. In our framework this update can be cast as special case of mRK when $\beta = 0$ and $\omega = 1$.

In this experiment we keep the stepsize fixed and equal to $\omega = 1$ which means that the pair of the chosen nodes update their values to their exact average and we show that by choosing a suitable momentum parameter $\beta \in (0, 1)$ we can obtain faster convergence to the consensus for all networks under study. The momentum parameter β is chosen following the suggestions made in Chapter 2 for solving general consistent linear systems. See Figure 4.8 for more details. It is worth to point out that for all networks under study the addition of a heavy ball momentum term is beneficial in the performance of the method.

Comparison of mRK and shift-Register algorithm [106]

In this experiment we compare mRK with the shift register gossip algorithm (pairwise momentum method, abbreviation: Pmom) analyzed in [106]. We choose the parameters ω and β of mRK in such a way in order to satisfy the connection established in Section 4.4.1. That is, we choose $\beta = \omega - 1$ for any choice of $\omega \in (1, 2)$. Observe that in all plots of Figure 4.9 mRK outperforms the corresponding shift-register algorithm.

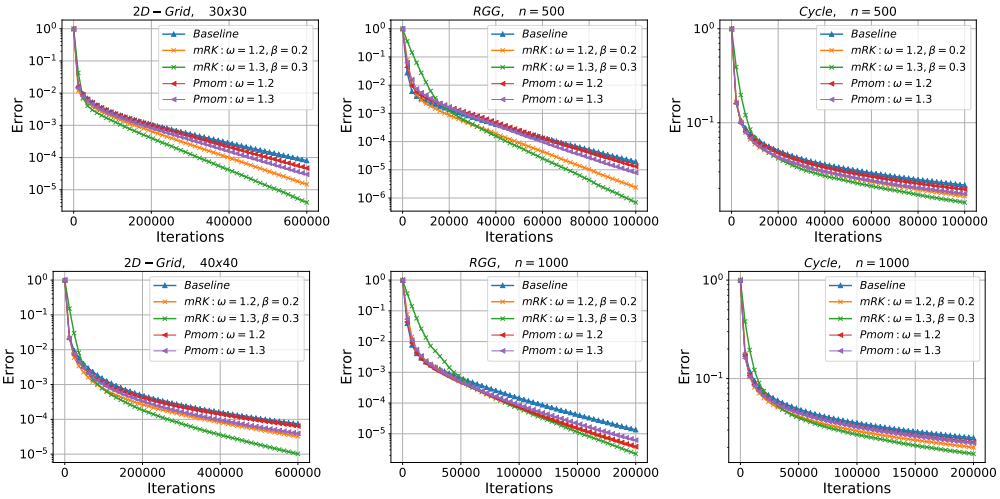


Figure 4.9: Comparison of mRK and the pairwise momentum method (Pmom), proposed in [106] (shift-register algorithm of Section 4.4.1). Following the connection between mRK and Pmom established in Section 4.4.1 the momentum parameter of mRK is chosen to be $\beta = \omega - 1$ and the stepsizes are selected to be either $\omega = 1.2$ or $\omega = 1.3$. The baseline method is the standard randomized pairwise gossip algorithm from [16]. The starting vector $x^0 = c \in \mathbb{R}^n$ is a Gaussian vector. The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

Impact of momentum parameter on mRBK

In this experiment our goal is to show that the addition of heavy ball momentum accelerates the RBK gossip algorithm presented in Section 4.3.4. Without loss of generality we choose the block size to be equal to $\tau = 5$. That is, the random matrix $\mathbf{S}_k \sim \mathcal{D}$ in the update rule of mRBK is a $m \times 5$ column submatrix of the indetity $m \times m$ matrix. Thus, in each iteration 5 edges of the network are chosen to form the subgraph $\mathcal{G}_k$ and the values of the nodes are updated according to Algorithm 11. Note that similar plots can be obtained for any choice of block size. We run all algorithms with fixed stepsize $\omega = 1$. From Figure 4.10, it is obvious that for all networks under study, choosing a suitable momentum parameter $\beta \in (0, 1)$ gives faster convergence than having no momentum, $\beta = 0$.

Performance of AccGossip

In the last experiment on faster gossip algorithms we evaluate the performance of the proposed provably accelerated gossip protocols of Section 4.4.2. In particular we compare the standard RK (pairwise gossip algorithm of [16]) the mRK (Algorithm 10) and the AccGossip (Algorithm 13) with the two options for the selection of the parameters presented in Section 4.4.2.

The starting vector of values $x^0 = c$ is taken to be a Gaussian vector. For the implementation of mRK we use the same parameters with the ones suggested in the stochastic heavy ball (SGB) setting in Chapter 2. For the AccRK (Option 1) we use $\lambda = \lambda_{\min}^+(\mathbf{A}^\top \mathbf{A})$ and for AccRK (Option 2) we select $\nu = m$[8]. From Figure 4.11 it is clear that for all networks under study the two randomized gossip protocols with Nesterov momentum are faster than both the pairwise gossip algorithm of [16] and the mRK/SHB (Algorithm 10). To the best of our knowledge Algorithm 13 (Option 1 and Option 2) is the first randomized gossip protocol that converges with provably accelerated linear rate and as we can see from our experiment its faster convergence is also obvious in practice.

[8]For the networks under study we have $m < \frac{1}{\lambda_{\min}^+(\mathbf{W})}$. Thus, by choosing $\nu = m$ we select the pessimistic upper bound of the parameter (4.42) and not its exact value (4.37). As we can see from the experiments, the performance is still accelerated and almost identical to the performance of AccRK (Option 1) for this choice of ν.

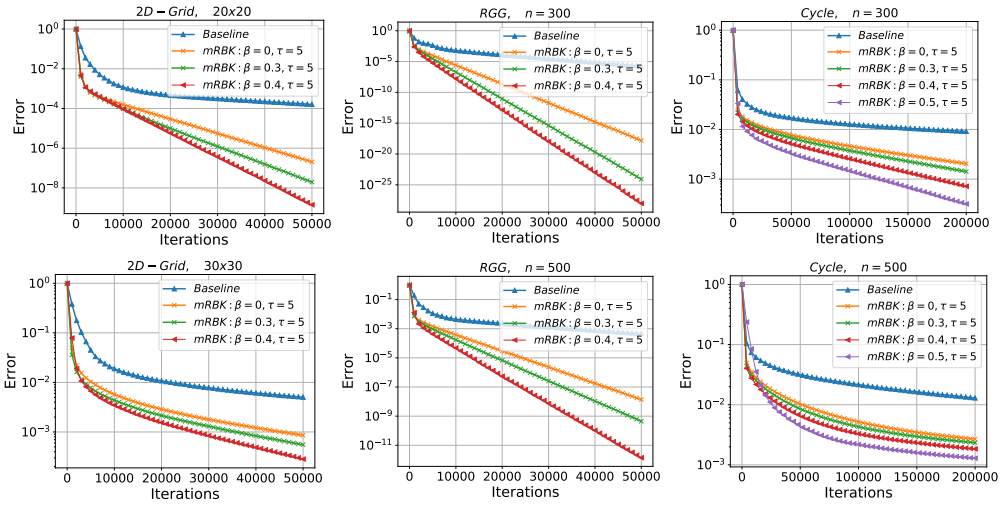


Figure 4.10: Comparison of mRBK with its no momentum variant RBK ($\beta = 0$). The stepsize for all methods is $\omega = 1$ and the block size is $\tau = 5$. The baseline method in the plots denotes the standard randomized pairwise gossip algorithm (block $\tau = 1$) and is plotted to highlight the benefits of having larger block sizes (at least in terms of iterations). The starting vector $x^0 = c \in \mathbb{R}^n$ is a Gaussian vector. The n in the title of each plot indicates the number of nodes. For the grid graph this is $n \times n$.

4.7.4 Relaxed randomized gossip without momentum

In the area of randomized iterative methods for linear systems it is know that over-relaxation (using of larger step-sizes) can be particularly helpful in practical scenarios. However, to the best of our knowledge there is not theoretical justification of why this is happening.

In our last experiment we explore the performance of relaxed randomized gossip algorithms ($\omega \neq 1$) without momentum and show that in this setting having larger stepsize can be particularly beneficial.

As we mentioned before (see Theorem 30) the sketch and project method (Algorithm 8) converges with linear rate when the step-size (relaxation parameter) of the method is $\omega \in (0, 2)$ and the best theoretical rate is achieved when $\omega = 1$. In this experiment we explore the performance of the standard pairwise gossip algorithm when the step-size of the update rule is chosen in $(1, 2)$. Since there is no theoretical proof of why over-relaxation can be helpful we perform the experiments using different starting values of the nodes. In particular we choose the values of vector $c \in R^n$ to follow (i) Gaussian distribution, (ii) Uniform Distribution and (iii) to be integers values such that $c_i = i \in R$. Our findings are presented in Figure 4.12. Note that for all networks under study and for all choices of starting values having larger stepsize, $\omega \in (1, 2)$ can lead to better performance. Interesting observation from Figure 4.12 is that the stepsizes $\omega = 1.8$ and $\omega = 1.9$ give the best performance (among the selected choices of stepsizes) for all networks and for all choices of starting vector $x^0 = c$.

4.8 Conclusion

In this chapter, we present a general framework for the analysis and design of randomized gossip algorithms. Using tools from numerical linear algebra and the area of randomized projection methods for solving linear systems we propose novel serial, block and accelerated gossip protocols for solving the average consensus and weighted average consensus problems.

We believe that this work could open up several future avenues for research. Using similar approach with the one presented in this manuscript, many popular projection methods can be interpreted as gossip algorithms when used to solve linear systems encoding the underlying network. This can lead to the development of novel distributed protocols for average consensus.

In addition, we speculate that the gossip protocols presented in this work can be extended to the more general setting of multi-agent consensus optimization where the goal is to minimize the average of convex or non-convex functions $\frac{1}{n}\sum_{i=1}^{n} f_i(x)$ in a decentralized way [131].

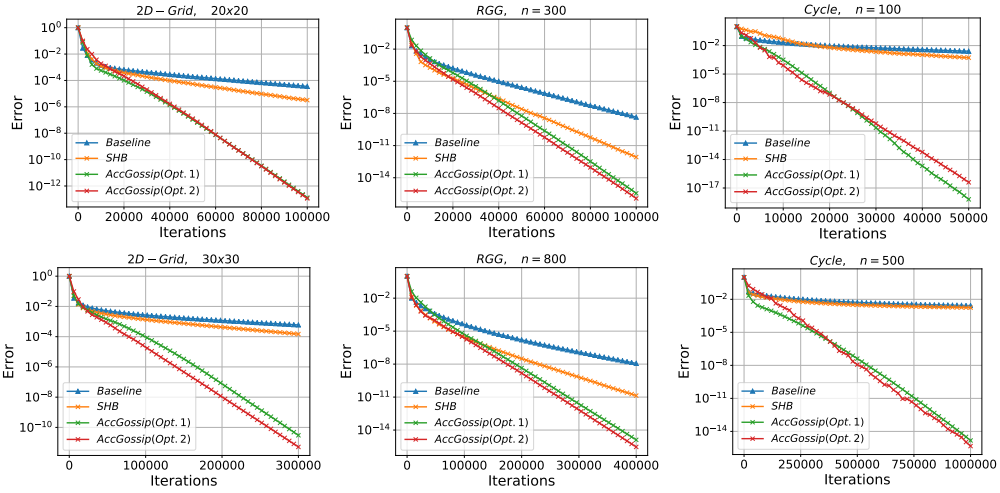


Figure 4.11: Performance of AccGossip (Option 1 and Option 2 for the parameters) in a 2-dimension grid, random geometric graph (RGG) and a cycle graph. The Baseline method corresponds to the randomized pairwise gossip algorithm proposed in [16] and the SHB represents the mRK (Algorithm 10) with the best choice of parameters as proposed in Chapter 2 ; The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

4.9 Missing Proofs

4.9.1 Proof of Theorem 33

Proof. Let $z^k := \|x^k - x^*\|$, $x^0 = c$ is the starting point and ρ is as defined in (4.5). From Theorem 30 we know that sketch and project method converges with

$$\mathbb{E}[\|x^k - x^*\|_{\mathbf{B}}^2] \leq \rho^k \|x^0 - x^*\|_{\mathbf{B}}^2, \tag{4.48}$$

where x^* is the solution of (1.22). Inequality (4.48), together with Markov inequality can be used to give the following bound

$$\mathbb{P}(z^k/z^0 \geq \varepsilon^2) \leq \frac{\mathbb{E}(z^k/z^0)}{\varepsilon^2} \leq \frac{\rho^k}{\varepsilon^2}. \tag{4.49}$$

Therefore, as long as k is large enough so that $\rho^k \leq \varepsilon^3$, we have $\mathbb{P}\left(z^k/z^0 \geq \varepsilon^2\right) \leq \varepsilon$. That is, if

$$\rho^k \leq \varepsilon^3 \Leftrightarrow k \geq \frac{3\log\varepsilon}{\log\rho} \Leftrightarrow k \geq \frac{3\log(1/\varepsilon)}{\log(1/\rho)},$$

then:

$$\mathbb{P}\left(\frac{\|x^k - \bar{c}1\|}{\|x^0 - \bar{c}1\|} \geq \varepsilon\right) \leq \varepsilon.$$

Hence, an upper bound for value $T_{ave}(\epsilon)$ can be obtained as follows,

$$\begin{aligned} T_{ave}(\epsilon) &= \sup_{c\in\mathbb{R}^n} \inf\left\{k \,:\, \mathbb{P}\left(z^k > \varepsilon z^0\right) \leq \varepsilon\right\} \leq \sup_{c\in\mathbb{R}^n} \inf\left\{k \,:\, k \geq \frac{3\log(1/\varepsilon)}{\log(1/\rho)}\right\} \\ &= \sup_{c\in\mathbb{R}^n} \frac{3\log(1/\varepsilon)}{\log(1/\rho)} = \frac{3\log(1/\varepsilon)}{\log(1/\rho)} \leq \frac{3\log(1/\varepsilon)}{1-\rho}, \end{aligned} \tag{4.50}$$

where in last inequality we use $1/\log(1/\rho) \leq 1/1-\rho$ which is true because $\rho \in (0,1)$. □

4.9.2 Proof of Theorem 34

Proof. The following notation conventions are used in this proof. With q_k we indicate the number of connected components of subgraph $\mathcal{G}_k$, while with $\mathcal{V}_r$ we denote the set of nodes of

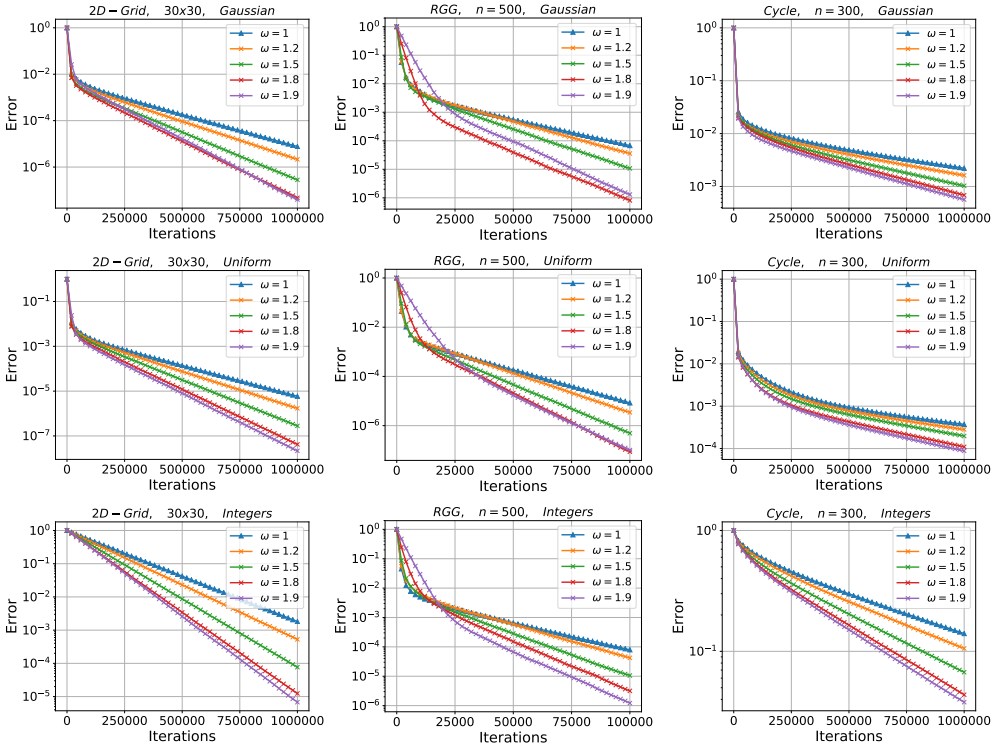


Figure 4.12: Performance of Relaxed randomized pairwise Gossip algorithm in a 2-dimension grid, random geometric graph (RGG) and a cycle graph. The case of $\omega = 1$ corresponds to the randomized pairwise gossip algorithm proposed in [16] ; The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$. The title of each plot indicates the vector of starting values that is used.

each connected component q_k ($r \in \{1, 2, \ldots, q_k\}$). Finally, $|\mathcal{V}_r|$ shows the cardinality of set $\mathcal{V}_r$. Notice that, if $\mathcal{V}$ is the set of all nodes of the graph then $\mathcal{V} = \bigcup_{r=\{1,2,\ldots q\}} \mathcal{V}_r$ and $|\mathcal{V}| = \sum_{r=1}^{q} |\mathcal{V}_r|$.

Note that from equation (4.24), the update of RBK for $\mathbf{A} = \mathbf{Q}$(Incidence matrix) can be expressed as follows:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \phi^k(x) := \|x - x^k\|^2 \\ \text{subject to} \quad & \mathbf{I}_{:C}^\top \mathbf{Q} x = 0 \end{aligned} \tag{4.51}$$

Notice that $\mathbf{I}_{:C}^\top \mathbf{Q}$ is a row submatrix of matrix $\mathbf{Q}$ with rows those that correspond to the random set $C \subseteq \mathcal{E}$ of the edges. From the expression of matrix $\mathbf{Q}$ we have that

$$(\mathbf{I}_{:C}^\top \mathbf{Q})_{e:}^\top = f_i - f_j, \quad \forall e = (i, j) \in C \subseteq \mathcal{E}.$$

Now, using this, it can be seen that the constraint $\mathbf{I}_{:C}^\top \mathbf{Q} x = 0$ of problem (1.22) is equivalent to q equations (number of connected components) where each one of them forces the values x_i^{k+1} of the nodes $i \in \mathcal{V}_r$ to be equal. That is, if we use z_r to represent the value of all nodes that belong in the connected component r then:

$$x_i^{k+1} = z_r \quad \forall i \in \mathcal{V}_r, \tag{4.52}$$

and the constrained optimization problem (1.22) can expressed as unconstrained as follows:

$$\underset{z}{\text{minimize}} \quad \phi^k(z) = \sum_{i \in \mathcal{V}_1} (z_1 - x_i^k)^2 + \ldots + \sum_{i \in \mathcal{V}_q} (z_q - x_i^k)^2, \tag{4.53}$$

where $z = (z_1, z_2, \ldots, z_q) \in \mathbb{R}^q$ is the vector of all values z_r when $r \in \{1, 2, \ldots, q\}$. Since our problem is unconstrained the minimum of equation (4.53) is obtained when $\nabla \phi^k(z) = 0$.

By evaluating partial derivatives of (4.53) we obtain:

$$\frac{\partial \phi^k(z)}{\partial z_r} = 0 \Longleftrightarrow \sum_{i \in \mathcal{V}_r} 2(z_r - x_i^k) = 0.$$

As a result,

$$z_r = \frac{\sum\limits_{i \in \mathcal{V}_r} x_i^k}{|\mathcal{V}_r|}, \quad \forall r \in \{1, 2, \ldots, q\}.$$

Thus from (4.52), the value of each node $i \in \mathcal{V}_r$ will be updated to

$$x_i^{k+1} = z_r = \frac{\sum\limits_{i \in \mathcal{V}_r} x_i^k}{|\mathcal{V}_r|}.$$

. □

Chapter 5

Privacy Preserving Randomized Gossip Algorithms

5.1 Introduction

In this chapter, similar to Chapter 4, we consider the average consensus (AC) problem. In particular, we focus on randomized gossip algorithms for solving the AC problem and propose techniques for protecting the information of the initial values c_i, as these may be sensitive. We develop and analyze three privacy preserving variants of the randomized pairwise gossip algorithm ("randomly pick an edge $(i,j) \in \mathcal{E}$ and then replace the values stored at vertices i and j by their average") first proposed in [16] for solving the average consensus problem. While we shall not formalize the notion of privacy preservation in this work, it will be intuitively clear that our methods indeed make it harder for nodes to infer information about the private values of other nodes, *which might be useful in practice.*

5.1.1 Background

The literature on decentralized protocols for solving the average consensus problem is vast and has long history [191, 190, 10, 93]. In particular, the algorithms for solving this problem can be divided into two broad categories: the average consensus algorithms [201] which work in a synchronous setting and the gossip algorithms [16, 176] which they consider ideal protocols for the asynchronous time model [16]. In the average consensus algorithms, all nodes of the network update their values simultaneously by communicating with a set of their neighbours and in each iteration of the algorithmic procedure the same update occurs. On the other hand, in gossip protocols only one edge of the whole network is selected at each iteration and only the nodes, that this edge connects, exchange their private information and update their values to their average.

In this chapter, we focus on modifying the basic algorithm of [16], which we refer to as "Standard Gossip" algorithm. In the following, we review some of the most important gossip protocols for solving the average consensus proposed in the last decade. While we do not address any privacy considerations for these protocols, they can serve as inspiration for further work. For a survey of relevant work, we refer the interested reader to [42, 147, 165, 131].

The *Geographic Gossip algorithm* was proposed in [43], in which the authors combine the gossip approach with a geographic routing towards a randomly chosen location with the main goal to improve the convergence rate of Standard Gossip algorithm. In each step, a node is activated, assuming that it is aware of its geographic location and some additional assumptions on the network topology, it chooses another node from the rest of the network (not necessarily one of its neighbours) and performs a pairwise averaging with this node. Later, using the same assumptions, this algorithm was extended into *Geographic Gossip Algorithm with Path Averaging* [8], in which connected sequences of nodes were chosen in each step and they averaged their values. More recently, in [57] and [58] authors propose a geographic and path averaging methods which converge to the average consensus without the assumption that nodes are aware

of their geographic location. Recall that, in Section 4.3.4 we show how the path averaging gossip algorithm can be seen as special case of the Randomized Block Kaczmarz method for solving consistent linear systems.

Another important randomized gossip algorithm is the *Broadcast Gossip algorithm*, first proposed in [7] and then extended in [55, 200, 90]. The idea of this algorithm is simple: In each step, a node in the network is activated uniformly at random, following the asynchronous time model, and broadcasts its value to its neighbours. The neighbours receive this value and update their own values. It was experimentally shown that this method converges faster than the pairwise and geographic randomized gossip algorithms.

Alternative gossip protocols are the so-called *non-randomized Gossip algorithms* [127, 82, 108, 208]. Typically, this class of algorithms executes the pairwise exchanges between nodes in a deterministic, such as predefined cyclic, order. T-periodic gossiping is a protocol which stipulates that each node must interact with each of its neighbours exactly once every T time units. Under suitable connectivity assumptions of the network $\mathcal{G}$, the T-periodic gossip sequence will converge at a rate determined by the magnitude of the second largest eigenvalue of the stochastic matrix determined by the sequence of pairwise exchanges which occurs over a period. It has been shown that if the underlying graph is a tree, the mentioned eigenvalue is constant for all possible T-periodic gossip protocols.

Accelerated Gossip algorithms have also been proposed for solving the average consensus problem. In this setting, the nodes of the network incorporate additional memory to accelerate convergence. In particular, the nodes update their value using an update rule that involves not only the current values of the sampled nodes but also their previous values. This idea is closely related to the shift register methods studied in numerical linear algebra for improving the convergence rate of linear system solvers. The works [19, 106] have shown theoretically and numerically, that under specific assumptions this idea can improve the performance of the Standard Gossip algorithm. For more details on these gossip protocols check also Section 4.4.1 of this thesis.

Randomized Kaczmarz-type Gossip algorithms. In Chapter 4 of this thesis we presented how popular randomized Kaczmarz-type methods for solving large linear systems can also solve the AC problem. We explained how these methods can be interpreted as randomized gossip algorithms when applied to special systems encoding the underlying network structure and present in detail their decentralized nature.

Asynchronous Time Model: In this chapter, we are interested in the asynchronous time model [16, 10]. More precisely, we assume that each node of our network has a clock which ticks at a rate of 1 Poisson process. This is equivalent of having available a global clock which ticks according to a rate n Poisson process and selects an edge of the network uniformly at random. In general, the synchronous setting (all nodes update the values of their nodes simultaneously using information from a set of their neighbours) is convenient for theoretical considerations but is not representative of some practical scenarios, such as the distributed nature of sensor networks. For more details on clock modeling we refer the reader to [16], as the contribution of this chapter is orthogonal to these considerations.

Privacy and Average Consensus: Finally, the introduction of notions of privacy within the AC problem is relatively recent in the literature, and the existing works consider two different ideas.

1. In [86], the concept of differential privacy [47] is used to protect the output value $\bar{c}$ computed by all nodes. In this work, an exponentially decaying Laplacian noise is added to the consensus computation. This notion of privacy refers to protection of the *final average*, and formal guarantees are provided.

2. A different approach with a more stricter goal is the design of privacy-preserving average consensus protocols that guarantee protection of the *initial values* c_i of the nodes [142, 123, 124]. In this setting each node should be unable to infer a lot about the initial values c_i of any other node. In the existing works, this is mainly achieved with the clever addition of noise through the iterative procedure that guarantees preservation of privacy

and at the same time converges to the exact average. We shall however mention, that none of these works address any specific notion of privacy (no clear measure of privacy is presented) and it is still not clear how the formal concept of differential privacy [47] can be applied in this setting.

It is worth to highlight that all of the above-mentioned privacy preserving average consensus papers propose protocols which work on the synchronous setting (all nodes update their values simultaneously). To the best of our knowledge our proposed protocols are the first that solve the AC problem and at the same time protect the initial values of the nodes using the asynchronous time model (by having gossip updates).

5.1.2 Main contributions

In this chapter, we present three different approaches for solving the Average Consensus problem while at the same time protecting the information about the initial values. To the best of our knowledge, this work is the first which combines the *gossip framework* with the privacy concept of protection of the initial values. It is important to stress that, we provide tools for protection of the initial values, but we do not address any specific notion of privacy or a threat model, nor how these quantitatively translate to any explicit measure. These would be highly application dependent, and we only provide theoretical convergence rates for the techniques we propose.

The methods we propose are all dual in nature. The dual setting of this chapter will be explained in detail in Section 5.2. Recall that in Chapters 1, 2 and 3 of this thesis, we have shown how duality and dual algorithms can be used for solving consistent linear systems. In addition, in Chapter 4, the dual viewpoint was extended to the concept of the average consensus problem and the first dual gossip algorithms were presented. As we have seen, the dual updates correspond to updates of the primal variables, via an affine mapping. Using this relationship of the primal and the dual spaces the convergence analysis of the dual methods can be easily obtained once the analysis of the primal methods is available (see for example the proof of Theorem 6 in the introduction of this thesis). In this chapter, one of our contributions is a novel dual analysis of randomized pairwise gossip (without the use of rates that obtain first through a primal analysis) which exactly recovers existing convergence rates for the primal iterates.

We now outline the three different techniques we propose in this chapter, which we refer to as "Binary Oracle", "ϵ-Gap Oracle" and "Controlled Noise Insertion". The first two are, to best of our knowledge, the first proposals of weakening the oracle used in the gossip framework. Privacy preservation is attained implicitly, as the nodes do not exchange the full information about their values. The last technique is inspired by the addition of noise proposed in [101, 142, 123, 124] for the synchronous setting. We extend this technique by providing explicit finite time convergence guarantees.

Binary Oracle. We propose to reduce the amount of information transmitted in each iteration to a single bit[1]. More precisely, when an edge is selected, each corresponding node will only receive information whether the value on the other node is smaller or larger. Instead of setting the value on the selected nodes to their average, each node increases or decreases its value by a pre-specified step.

ϵ-Gap Oracle. In this case, we have an oracle that returns one of three options and is parametrized by ϵ. If the difference in values of sampled nodes is larger than ϵ, an update similar to the one in Binary Oracle is taken. Otherwise, the values remain unchanged. An advantage compared to the Binary Oracle is that this approach will converge to a certain accuracy and stop there, determined by ϵ (Binary Oracle will oscillate around optimum for a fixed stepsize). However, in general, it will disclose more information about the initial values.

Controlled Noise Insertion. This approach is inspired by the works of [123, 124], and protects the initial values by inserting noise in the process. Broadly speaking, in each iteration, each of the sampled nodes first adds a noise to its current value, and an average is computed

[1]We do not refer to the size of the object being transmitted over the network, but the binary information that can be inferred from the exchange. In practice, this might be achieved using secure multiparty protocols [26], causing the overall network bandwidth to slightly increase compared to the usual implementation of standard gossip algorithm.

Main Results			
Randomized Gossip Methods	Convergence Rate	Success Measure	Thm
Standard Gossip [16]	$\left(1-\frac{\alpha(\mathcal{G})}{2m}\right)^k$	$\mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1}-x^k\|^2\right]$	42
New: Private Gossip with Binary Oracle	$1/\sqrt{k}$	$\min_{t\leq k}\mathbb{E}\left[\frac{1}{m}\sum_e\|x_i^t-x_j^t\|\right]$	44
New: Private Gossip with ϵ-Gap Oracle	$1/(k\epsilon^2)$	$\mathbb{E}\left[\frac{1}{k}\sum_{t=0}^{k-1}\Delta^t(\epsilon)\right]$	47
New: Private Gossip with Controlled Noise Insertion	$\left(1-\min\left(\frac{\alpha(\mathcal{G})}{2m},\frac{\gamma}{m}\right)\right)^k$	$\mathbb{E}\left[D(y^*)-D(y^k)\right]$	49

Table 5.1: Complexity results of all proposed privacy preserving randomized gossip algorithms.

afterward. Convergence is guaranteed due to the correlation in the noise across iterations. Each node remembers the noise it added last time it was sampled, and in the following iteration, the previously added noise is first subtracted, and a fresh noise of smaller magnitude is added. Empirically, the protection of initial values is provided by first injecting noise into the system, which propagates across the network, but is gradually withdrawn to ensure convergence to the true average.

Convergence Rates of our Methods: In Table 5.1, we present the summary of convergence guarantees for the above three techniques. By $\|\cdot\|$ we denote the standard Euclidean norm.

The two approaches which restrict the amount of information disclosed, Binary Oracle and ϵ-Gap Oracle, converge slower than the standard Gossip. In particular, these algorithms have sublinear convergence rate. At first sight, this should not be surprising, since we indeed use much less information. However, in Theorem 45, we show that if we had in a certain sense perfect global information, we could use it to construct a sequence of adaptive stepsizes, which would push the capability of the binary oracle to a linear convergence rate. However, this rate is still m-times slower than the standard rate of the binary gossip algorithm. We note, however, that having the global information at hand is an impractical assumption. Nevertheless, this result highlights that there is a potentially large scope for improvement, which we leave for future work.

The approach of Controlled Noise Insertion yields a linear convergence rate which is driven by the minimum of two factors. Without going into details, which of these is bigger depends on the speed by which the magnitude of the inserted noise decays. If the noise decays fast enough, we recover the convergence rate of the standard the gossip algorithm. In the case of slow decay, the convergence is driven by this decay. By $\alpha(\mathcal{G})$ we denote the *algebraic connectivity* of graph $\mathcal{G}$ [52]. The parameter γ controls the decay speed of the inserted noise, see Corollary 50.

Measures of Success: Note that the convergence of each randomized gossip algorithm in Table 5.1 naturally depends on a different measure of suboptimality. All of them converge to 0 as we approach the optimal solution. The details of these measures will be described later in the main body of this chapter. In particular a lemma that formally describes the key connections between these measures is presented in Section 5.3.1. For now lets us give a brief description of these results. The standard Gossip and Controlled Noise Insertion essentially depend on the same quantity, but we present the latter in terms of dual values as this is what our proofs are based on. Lemma 43 formally specifies this equivalence. The binary oracle depends on the average difference among directly connected nodes. The measure for the ϵ-Gap Oracle depends on quantities $\Delta^t(\epsilon)=\frac{1}{m}\left|\{(i,j)\in\mathcal{E}:|x_i^t-x_j^t|\geq\epsilon\}\right|$, which is the number of edges that the values of their connecting nodes differ by more than ϵ.

5.1.3 Structure of the chapter

The remainder of this chapter is organized as follows: Section 5.2 introduces the basic setup that is used through the chapter. A detailed explanation of the duality behind the randomized pairwise gossip algorithm is given. We also include a novel and insightful dual analysis of

this method as it will make it easier for the reader to parse later development. In Section 5.3 we present our three private gossip algorithms as well as the associated iteration complexity results. Section 5.4 is devoted to the numerical evaluation of our methods. Finally, conclusions are drawn in Section 5.5.

5.2 Dual Analysis of Randomized Pairwise Gossip

As we outlined in the introduction of this chapter, our approach for extending the (standard) randomized pairwise gossip algorithm to privacy preserving variants utilizes duality. The purpose of this section is to formalize this duality. In addition, we provide a novel and self-contained dual analysis of randomized pairwise gossip. While this is of an independent interest, we include the proofs as their understanding aids in the understanding of the more involved proofs of our private gossip algorithms developed in the remainder of the chapter.

The main problems under study are the best approximation problem (1.22) and its dual (1.26) that we have seen multiple times throughout the thesis. However, similar to Chapter 4 we focus on the more specific setting of the average consensus. To keep the chapter self-contained and for the benefit of the reader we present the definitions of these problems and we explain again how they are related to the average consensus problem.

5.2.1 Primal and dual problems

Consider solving the (primal) problem of projecting a given vector $c = x^0 \in \mathbb{R}^n$ onto the solution space of a linear system:

$$\min_{x\in\mathbb{R}^n} P(x) := \frac{1}{2}\|x - x^0\|^2 \quad \text{subject to} \quad \mathbf{A}x = b, \tag{5.1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$. Note that this the best approximation problem (1.22) with $\mathbf{B} = \mathbf{I}$ (Identity matrix). We assume the problem is feasible, i.e., that the system $\mathbf{A}x = b$ is consistent. With the above optimization problem we associate the dual problem

$$\max_{y\in\mathbb{R}^m} D(y) := (b - \mathbf{A}x^0)^\top y - \frac{1}{2}\|\mathbf{A}^\top y\|^2. \tag{5.2}$$

As we have explained in the previous chapters, the dual is an unconstrained concave (but not necessarily strongly concave) quadratic maximization problem. It can be seen that as soon as the system $\mathbf{A}x = b$ is feasible, the dual problem is bounded. Moreover, all bounded concave quadratics in $\mathbb{R}^m$ can be written in the as $D(y)$ for some matrix $\mathbf{A}$ and vectors b and x^0 (up to an additive constant).

With any dual vector y we associate the primal vector via an affine transformation: $\phi(y) = x^0 + \mathbf{A}^\top y$. It can be shown that if y^* is dual optimal, then $x^* = \phi(y^*)$ is primal optimal [74]. Hence, any dual algorithm producing a sequence of dual variables $y^t \to y^*$ gives rise to a corresponding primal algorithm producing the sequence $x^t := \phi(y^t) \to x^*$. We shall now consider one such dual algorithm.

5.2.2 Stochastic dual subspace ascent

Stochastic dual subspace ascent (SDSA) is a stochastic method for solving the dual problem (5.2). In Section 1.4 we have already described how by choosing appropriately the main parameters of SDSA we can recover many known algorithms as special cases. In this chapter we focus only on one special case of the general algorithm. For the more general update rule of SDSA check equations (1.29) and (1.30). In particular, following the notation of the rest of the thesis, we select $\omega = 1$ (stepsize of the method) and $\mathbf{B} = \mathbf{I}$ (positive definite matrix that defines the geometry of the space). If we further use the fact that AC linear systems (see Definition 31) have zero right hand side ($b = 0$), then the update rule of SDSA takes the form:

$$y^{t+1} = y^t - \mathbf{S}_t(\mathbf{S}_t^\top \mathbf{A}\mathbf{A}^\top \mathbf{S}_t)^\dagger \mathbf{S}_t^\top \mathbf{A}(x^0 + \mathbf{A}^\top y^t), \tag{5.3}$$

where $\mathbf{S}_t$ is a random matrix drawn independently at each iteration t from an arbitrary but fixed distribution $\mathcal{D}$, and $\dagger$ denotes the Moore-Penrose pseudoinverse.

The corresponding primal iterates are defined via:

$$x^t := \phi(y^t) = x^0 + \mathbf{A}^\top y^t. \tag{5.4}$$

The relevance of this all to average consensus follows through the observation that for a specific choice of matrix $\mathbf{A}$ and distribution $\mathcal{D}$, the primal method produced by combining (5.4) and (5.3) is equivalent to the (standard) randomized pairwise gossip method (see discussion in Chapter 4). In that case, SDSA is a dual variant of randomized pairwise gossip. In particular, in this chapter, we define $\mathcal{D}$ as follows: $\mathbf{S}_t$ is a unit basis vector in $\mathbb{R}^m$, chosen uniformly at random from the collection of all such unit basis vectors, denoted $\{f_e \mid e \in \mathcal{E}\}$. In this case, SDSA is the *randomized coordinate ascent method* applied to the dual problem.

5.2.3 Randomized gossip setup: choosing A

We wish $(\mathbf{A}, b)$ to be an average consensus (AC) system (see Definition 31). As we explained in Chapter 4 if $\mathbf{A}x = b$ is an AC system, then the solution of the primal problem (5.1) is necessarily $x^* = \bar{c} \cdot \mathbf{1}$, where $\bar{c} = \frac{1}{n}\sum_{i=1}^n x_i^0$ is the value that each node needs to compute in the standard average consensus problem ($x_i^* = \bar{c}$ for all $i \in \mathcal{V}$).

In the rest of this chapter we focus on a specific AC system; the one in which the matrix $\mathbf{A}$ is the incidence matrix of the graph $\mathcal{G}$. In particular, we let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be the matrix defined as follows. Row $e = (i, j) \in \mathcal{E}$ of $\mathbf{A}$ is given by $\mathbf{A}_{ei} = 1$, $\mathbf{A}_{ej} = -1$ and $\mathbf{A}_{el} = 0$ if $l \notin \{i, j\}$. Notice that the system $\mathbf{A}x = 0$ encodes the constraints $x_i = x_j$ for all $(i, j) \in \mathcal{E}$, as desired.

5.2.4 Randomized pairwise gossip

We provide both primal and dual form of the (standard) randomized pairwise gossip algorithm.

The primal form is standard and needs no lengthy commentary. At the beginning of the process, node i contains private information $c_i = x_i^0$. In each iteration we sample a pair of connected nodes $(i, j) \in \mathcal{E}$ uniformly at random, and update x_i and x_j to their average. We let the values at the remaining nodes intact.

Algorithm 14 (Primal form)

Input: Vector of private values $c \in \mathbb{R}^n$.
Initialize: Set $x^0 = c$.
1: **for** $t = 0, 1, \ldots, k-1$ **do**
2: Choose edge $e = (i, j) \in \mathcal{E}$ uniformly at random.
3: Update the primal variable:

$$x_l^{t+1} = \begin{cases} \frac{x_i^t + x_j^t}{2}, & l \in \{i, j\} \\ x_l^t, & l \notin \{i, j\}. \end{cases}$$

4: **end for**
5: **Return** x^k

The dual form of the standard randomized pairwise gossip method is a specific instance of SDSA, as described in (5.3), with $x^0 = c$ and $\mathbf{S}_t$ being a randomly chosen standard unit basis vector f_e in $\mathbb{R}^m$ (e is a randomly selected edge). It can be seen [74] that in that case, (5.3) takes the following form:

Algorithm 14 (Dual form)

Input: Vector of private values $c \in \mathbb{R}^n$.
Initialize: Set $y^0 = 0 \in \mathbb{R}^m$.
1: **for** $t = 0, 1, \ldots, k-1$ **do**
2: Choose edge $e = (i, j) \in \mathcal{E}$ uniformly at random.
3: Update the dual variable:

$$y^{t+1} = y^t + \lambda^t f_e \quad \text{where} \quad \lambda^t = \text{argmax}_{\lambda'} D(y^t + \lambda' f_e).$$

4: **end for**
5: **Return** y^k

The following lemma is useful for the analysis of all our methods. It describes the increase in the dual function value after an arbitrary change to a single dual variable e.

Lemma 40. *Define $z = y^t + \lambda f_e$, where $e = (i, j)$ and $\lambda \in \mathbb{R}$. Then*

$$D(z) - D(y^t) = -\lambda(x_i^t - x_j^t) - \lambda^2. \tag{5.5}$$

Proof. The claim follows by direct calculation:

$$\begin{aligned} D(y^t + \lambda f_e) - D(y^t) &= -(\mathbf{A}c)^\top (y^t + \lambda f_e) - \frac{1}{2}\|\mathbf{A}^\top (y^t + \lambda f_e)\|^2 + (\mathbf{A}c)^\top y^t + \frac{1}{2}\|\mathbf{A}^\top y^t\|^2 \\ &= -\lambda f_e^\top \mathbf{A} \underbrace{(c + \mathbf{A}^\top y^t)}_{x^t} - \frac{1}{2}\lambda^2 \underbrace{\|\mathbf{A}^\top f_e\|^2}_{=2} \quad = \quad -\lambda(x_i^t - x_j^t) - \lambda^2. \end{aligned}$$

□

The maximizer in λ of the expression in (5.5) leads to the exact line search formula $\lambda^t = (x_j^t - x_i^t)/2$ used in the dual form of the method.

5.2.5 Complexity results

With graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ we now associate a certain quantity, which we shall denote $\beta = \beta(\mathcal{G})$. It is the smallest nonnegative number β such that the following inequality[2] holds for all $x \in \mathbb{R}^n$:

$$\sum_{(i,j)} (x_j - x_i)^2 \leq \beta \sum_{(i,j)\in\mathcal{E}} (x_j - x_i)^2. \tag{5.6}$$

The Laplacian matrix of graph $\mathcal{G}$ is given by $\mathbf{L} = \mathbf{A}^\top \mathbf{A}$. Let $\lambda_1(\mathbf{L}) \geq \lambda_2(\mathbf{L}) \geq \cdots \geq \lambda_{n-1}(\mathbf{L}) \geq \lambda_n(\mathbf{L})$ be the eigenvalues of $\mathbf{L}$. The *algebraic connectivity* of $\mathcal{G}$ is the second smallest eigenvalue of $\mathbf{L}$:

$$\alpha(\mathcal{G}) = \lambda_{n-1}(\mathbf{L}). \tag{5.7}$$

We have $\lambda_n(\mathbf{L}) = 0$. Since we assume $\mathcal{G}$ to be connected, we have $\alpha(\mathcal{G}) > 0$. Thus, $\alpha(\mathcal{G})$ is the smallest nonzero eigenvalue of the Laplacian: $\alpha(\mathcal{G}) = \lambda_{\min}^+(\mathbf{L}) = \lambda_{\min}^+(\mathbf{A}^\top \mathbf{A})$. As the next result states, the quantities $\beta(\mathcal{G})$ and $\alpha(\mathcal{G})$ are inversely proportional.

Lemma 41. $\beta(\mathcal{G}) = \frac{n}{\alpha(\mathcal{G})}$.

Proof. See Section 5.6.1. □

The following theorem gives a complexity result for (standard) randomized gossip. Our analysis is dual in nature.

[2]We write $\sum_{(i,j)}$ to indicate sum over all *unordered* pairs of vertices. That is, we do not count (i, j) and (j, i) separately, only once. By $\sum_{(i,j)\in\mathcal{E}}$ we denote a sum over all edges of $\mathcal{G}$. On the other hand, by writing $\sum_i \sum_j$, we are summing over all (unordered) pairs of vertices twice.

Theorem 42. *Consider the randomized gossip algorithm (Algorithm 14) with uniform edge-selection probabilities: $p_e = 1/m$. Then:*

$$\mathbb{E}\left[D(y^*) - D(y^k)\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)^k [D(y^*) - D(y^0)].$$

Proof. See Section 5.6.2 □

Theorem 42 yields the complexity estimate $\mathcal{O}\left(\frac{2m}{\alpha(\mathcal{G})}\log(1/\epsilon)\right)$, which exactly matches the complexity result obtained from the primal analysis (see (4.22) in Chapter 4). Hence, the primal and dual analyses give the same rate.

Randomized coordinate descent methods were first analyzed in [104, 139, 166, 167]. For a recent treatment, see [160, 161]. Duality in randomized coordinate descent methods was studied in [178, 163]. Acceleration was studied in [102, 51, 2]. These methods extend to nonsmooth problems of various flavours [50, 22].

With all of this preparation, we are now ready to formulate and analyze our private gossip algorithms; we do so in Section 5.3.

5.3 Private Gossip Algorithms

In this section, we introduce three novel private gossip algorithms, complete with iteration complexity guarantees. In Section 5.3.1 the key relationships between the measures of success (see Table 5.1) of all proposed algorithms are presented. In Section 5.3.2 the privacy is protected via a binary communication protocol. In Section 5.3.3 we communicate more: besides binary information, we allow for the communication of a bound on the gap, introducing the ϵ-gap oracle. In Section 5.3.4 we introduce a privacy-protection mechanism based on a procedure we call *controlled noise insertion*.

5.3.1 Measures of success

We devote this subsection to present Lemma 43 that formally specifies the connections between the different measures of suboptimality of the privacy preserving algorithms, firstly presented in Table 5.1.

Lemma 43. *(Relationship between convergence measures) Suppose that x is primal variable corresponding to the dual variable y as defined in* (5.4)*. Dual suboptimality can be expressed as the following [74]:*

$$D(y^*) - D(y) = \frac{1}{2}\|\bar{c}\mathbf{1} - x\|^2. \tag{5.8}$$

Moreover, for any $x \in \mathbb{R}^n$ we have :

$$\frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j - x_i)^2 = \|\bar{c}\mathbf{1} - x\|^2 \tag{5.9}$$

$$\sum_{e=(i,j)\in\mathcal{E}} |x_i - x_j| \leq \sqrt{mn}\|\bar{c}\mathbf{1} - x\|, \tag{5.10}$$

$$\sum_{e=(i,j)\in\mathcal{E}} |x_i - x_j| \geq \sqrt{\alpha(\mathcal{G})}\|\bar{c}\mathbf{1} - x\|, \tag{5.11}$$

$$\sum_{e=(i,j)\in\mathcal{E}} |x_i - x_j| \geq \epsilon\left|\{(i,j)\in\mathcal{E} : |x_i - x_j| \geq \epsilon\}\right|. \tag{5.12}$$

Proof. See Section 5.6.3. □

5.3.2 Private gossip via binary oracle

We now present the gossip algorithm with Binary Oracle in detail and provide theoretical convergence guarantee. The information exchanged between sampled nodes is constrained to a

single bit, describing which of the nodes has the higher value. As mentioned earlier, we only present the conceptual idea, not how exactly would the oracle be implemented within a secure multiparty protocol between participating nodes [26].

We will first introduce the dual version of the algorithm.

Algorithm 15 (Dual form)

Input: Vector of private values $c \in \mathbb{R}^n$, sequence of positive stepsizes $\{\lambda^t\}_{t=0}^{\infty}$
Initialize: Set $y^0 = 0 \in \mathbb{R}^m$, $x^0 = c$.

1: **for** $t = 0, 1, \ldots, k-1$ **do**
2: Choose edge $e = (i, j) \in \mathcal{E}$ uniformly at random.
3: Update the dual variable:

$$y^{t+1} = \begin{cases} y^t + \lambda^t f_e, & x_i^t < x_j^t, \\ y^t - \lambda^t f_e, & x_i^t \geq x_j^t. \end{cases}$$

4: Set

$$\begin{aligned} x_i^{t+1} &= \begin{cases} x_i^t + \lambda^t, & x_i^t < x_j^t, \\ x_i^t - \lambda^t, & x_i^t \geq x_j^t. \end{cases} \\ x_j^{t+1} &= \begin{cases} x_j^t - \lambda^t, & x_i^t < x_j^t, \\ x_j^t + \lambda^t & x_i^t \geq x_j^t. \end{cases} \\ x_l^{t+1} &= x_l^t \qquad l \notin \{i, j\} \end{aligned}$$

5: **end for**
6: **Return** y^k

The update of primal variables above is equivalent to set x^{t+1} as primal point corresponding to dual iterate: $x^{t+1} = c + \mathbf{A}^\top y^{t+1} = x^t + \mathbf{A}^\top (y^{t+1} - y^t)$. In other words, the primal iterates $\{x^t\}$ associated with the dual iterates $\{y^t\}$ can be written in the form:

$$x^{t+1} = \begin{cases} x^t + \lambda^t \mathbf{A}_{e:}^\top, & x_i^t < x_j^t, \\ x^t - \lambda^t \mathbf{A}_{e:}^\top, & x_i^t \geq x_j^t. \end{cases}$$

It is easy to verify that due to the structure of $\mathbf{A}$, this is equivalent to the updates above.

Since the evolution of dual variables $\{y^k\}$ serves only the purpose of the analysis, the method can be written in the primal-only form as follows:

Algorithm 15 (Primal form)

Input: Vector of private values $c \in \mathbb{R}^n$, sequence of positive stepsizes $\{\lambda^t\}_{t=0}^{\infty}$
Initialize: Set $x^0 = c$.

1: **for** $t = 0, 1, \ldots, k-1$ **do**
2: Choose edge $e = (i, j) \in \mathcal{E}$ uniformly at random.
3: Set

$$\begin{aligned} x_i^{t+1} &= \begin{cases} x_i^t + \lambda^t, & x_i^t < x_j^t, \\ x_i^t - \lambda^t, & x_i^t \geq x_j^t. \end{cases} \\ x_j^{t+1} &= \begin{cases} x_j^t - \lambda^t, & x_i^t < x_j^t, \\ x_j^t + \lambda^t & x_i^t \geq x_j^t. \end{cases} \\ x_l^{t+1} &= x_l^t \qquad l \notin \{i, j\} \end{aligned}$$

4: **end for**
5: **Return** x^k

Given a sequence of stepsizes $\{\lambda^t\}$, it will be convenient to define $\alpha^k := \sum_{t=0}^k \lambda^t$ and $\beta^k := \sum_{t=0}^k (\lambda^t)^2$. In the following theorem, we study the convergence of the quantity

$$L^t := \frac{1}{m} \sum_{e=(i,j)\in\mathcal{E}} |x_i^t - x_j^t|. \tag{5.13}$$

Theorem 44. *For all $k \geq 1$ we have*

$$\min_{t=0,1,\ldots,k} \mathbb{E}\left[L^t\right] \leq \sum_{t=0}^k \frac{\lambda^t}{\alpha^k} \mathbb{E}\left[L^t\right] \leq U^k := \frac{D(y^*) - D(y^0)}{\alpha^k} + \frac{\beta^k}{\alpha^k}. \tag{5.14}$$

Moreover:

(i) If we set $\lambda^t = \lambda^0 > 0$ for all t, then $U^k = \frac{D(y^*)-D(y^0)}{\lambda^0(k+1)} + \lambda^0$.

(ii) Let R be any constant such that $R \geq D(y^*) - D(y^0)$. If we fix $k \geq 1$, then the choice of stepsizes $\{\lambda^0, \ldots, \lambda^k\}$ which minimizes U^k correspond to the constant stepsize rule $\lambda^t = \sqrt{\frac{R}{k+1}}$ for all $t = 0, 1, \ldots, k$, and $U^k = 2\sqrt{\frac{R}{k+1}}$.

(iii) If we set $\lambda^t = a/\sqrt{t+1}$ for all $t = 0, 1, \ldots, k$, then

$$U^k \leq \frac{D(y^*) - D(y^0) + a^2\left(\log(k+3/2) + \log(2)\right)}{2a\left(\sqrt{k+2} - 1\right)} = \mathcal{O}\left(\frac{\log(k)}{\sqrt{k}}\right)$$

Proof. See Section 5.6.4 □

The part *(ii)* of Theorem 44 is useful in the case that we know exactly the number of iterations before running the algorithm, providing in a sense optimal stepsizes and rate $\mathcal{O}(1/\sqrt{k})$. However, this might not be the case in practice. Therefore part *(iii)* is also relevant, which yields the rate $\mathcal{O}(\log(k)/\sqrt{k})$. These bounds are significantly weaker than the standard bound in Theorem 42. This should not be surprising though, as we use significantly less information than the Standard Gossip algorithm.

Nevertheless, there is a potential gap in terms of what rate can be practically achievable. The following theorem can be seen as a form of a bound on what convergence rate is possible to be attained by the Binary Oracle. However, this rate can be attained with access to very strong information. It requires a specific sequence of stepsizes λ^t which is likely unrealistic in practical scenarios. This result points to a gap in the analysis which we leave open. We do not know whether the sublinear convergence rate in Theorem 44 is necessary or improvable without additional information about the system.

Theorem 45. *For Algorithm 15 with stepsizes chosen in iteration t adaptively to the current values of x^t as $\lambda^t = \frac{1}{2m}\sum_{e\in\mathcal{E}} |x_i^t - x_j^t|$, we have*

$$\mathbb{E}\left[\|\bar{c}\mathbf{1} - x^k\|^2\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)^k \|\bar{c}\mathbf{1} - x^0\|^2$$

Proof. See Section 5.6.5 □

Comparing Theorem 45 with the result for standard Gossip in Theorem 42, the convergence rate is worse by factor of m, which is the price we pay for the weaker oracle.

An alternative to choosing adaptive stepsizes is the use of adaptive probabilities [28]. We leave such a study for future work.

5.3.3 Private gossip via ϵ-gap oracle

Here we present the gossip algorithm with ϵ-Gap Oracle in detail and provide theoretical convergence guarantees. The information exchanged between the sampled nodes is restricted to be one of three cases, based on the difference of their values. As mentioned earlier, we only

present the conceptual idea, not how exactly would the oracle be implemented within a secure multiparty protocol between participating nodes [26].

We will first introduce the dual version of the algorithm.

Algorithm 16 (Dual form)

Input: Vector of private values $c \in \mathbb{R}^n$; error tolerance $\epsilon > 0$
Initialize: Set $y^0 = 0 \in \mathbb{R}^m$; $x^0 = c$.

1: **for** $t = 0, 1, \ldots, k-1$ **do**
2: Choose edge $e = (i, j) \in \mathcal{E}$ uniformly at random.
3: Update the dual variable:

$$y^{t+1} = \begin{cases} y^t + \frac{\epsilon}{2} f_e, & x_i^t - x_j^t < -\epsilon \\ y^t - \frac{\epsilon}{2} f_e, & x_j^t - x_i^t < -\epsilon, \\ y^t, & \text{otherwise.} \end{cases}$$

4: If $x_i^t \leq x_j^t - \epsilon$ then $x_i^{t+1} = x_i^t + \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t - \frac{\epsilon}{2}$
5: If $x_j^t \leq x_i^t - \epsilon$ then $x_i^{t+1} = x_i^t - \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t + \frac{\epsilon}{2}$
6: **end for**
7: **Return** y^k

Note that the primal iterates $\{x^t\}$ associated with the dual iterates $\{y^t\}$ can be written in the form:

$$x^{t+1} = \begin{cases} x^t + \frac{\epsilon}{2} \mathbf{A}_{e:}^\top, & x_i^t - x_j^t < -\epsilon \\ x^t - \frac{\epsilon}{2} \mathbf{A}_{e:}^\top, & x_j^t - x_i^t < -\epsilon, \\ x^t, & \text{otherwise.} \end{cases}$$

The above is equivalent to setting $x^{t+1} = x^t + \mathbf{A}^\top (y^{t+1} - y^t) = c + \mathbf{A}^\top y^{t+1}$.

Since the evolution of dual variables $\{y^t\}$ serves only the purpose of the analysis, the method can be written in the primal-only form as follows:

Algorithm 16 (Primal form)

Input: Vector of private values $c \in \mathbb{R}^n$; error tolerance $\epsilon > 0$
Initialize: Set $x^0 = c$.

1: **for** $t = 0, 1, \ldots, k-1$ **do**
2: Set $x^{t+1} = x^t$
3: Choose edge $e = (i, j) \in \mathcal{E}$ uniformly at random.
4: If $x_i^t \leq x_j^t - \epsilon$ then $x_i^{t+1} = x_i^t + \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t - \frac{\epsilon}{2}$
5: If $x_j^t \leq x_i^t - \epsilon$ then $x_i^{t+1} = x_i^t - \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t + \frac{\epsilon}{2}$
6: **end for**
7: **Return** x^k

Before stating the convergence result, let us define a quantity the convergence will naturally depend on. For each edge $e = (i, j) \in \mathcal{E}$ and iteration $t \geq 0$ define the random variable

$$\Delta_e^t(\epsilon) \quad := \quad \begin{cases} 1, & |x_i^t - x_j^t| \geq \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, let

$$\Delta^t(\epsilon) \quad := \quad \frac{1}{m} \sum_{e \in \mathcal{E}} \Delta_e^t(\epsilon). \tag{5.15}$$

The following Lemma bounds the expected increase in dual function value in each iteration.

Lemma 46. *For all $t \geq 0$ we have $\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] \geq \frac{\epsilon^2}{4} \mathbb{E}\left[\Delta^t(\epsilon)\right]$.*

Proof. See Section 5.6.6 □

Our complexity result will be expressed in terms of the quantity:

$$\delta^k(\epsilon) \quad := \quad \mathbb{E}\left[\frac{1}{k}\sum_{t=0}^{k-1}\Delta^t(\epsilon)\right] = \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}\left[\Delta^t(\epsilon)\right]. \tag{5.16}$$

Theorem 47. *For all $k \geq 1$ we have*

$$\delta^k(\epsilon) \leq \frac{4\left(D(y^*) - D(y^0)\right)}{k\epsilon^2}.$$

Proof. See Section 5.6.7 □

Note that if $\Delta^k(\epsilon) = 0$, it does not mean the primal iterate x^k is optimal. This only implies that the values of all pairs of directly connected nodes differ by less than ϵ.

5.3.4 Private gossip via controlled noise insertion

In this section, we present the gossip algorithm with Controlled Noise Insertion. As mentioned in the introduction of this chapter, the approach is similar to the technique proposed in [123, 124]. Those works, however, address only algorithms in the synchronous setting, while our work is the first to use this idea in the asynchronous setting. Unlike the above, we provide finite time convergence guarantees and allow each node to add the noise differently, which yields a stronger result.

In our approach, each node adds noise to the computation independently of all other nodes. However, the noise added is correlated between iterations for each node. We assume that every node owns two parameters — the initial magnitude of the generated noise σ_i^2 and rate of decay of the noise ϕ_i. The node inserts noise $w_i^{t_i}$ to the system every time that an edge corresponding to the node was chosen, where variable t_i carries an information how many times the noise was added to the system in the past by node i. Therefore, if we denote by t the current number of iterations, we have $\sum_{i=1}^n t_i = 2t$.

In order to ensure convergence to the optimal solution, we need to choose a specific structure of the noise in order to guarantee the mean of the values x_i converges to the initial mean. In particular, in each iteration a node i is selected, we subtract the noise that was added last time, and add a fresh noise with smaller magnitude:

$$w_i^{t_i} = \phi_i^{t_i} v_i^{t_i} - \phi_i^{t_i-1} v_i^{t_i-1}, \tag{5.17}$$

where $0 \leq \phi_i < 1, v_i^{-1} = 0$ and $v_i^{t_i} \sim N(0, \sigma_i^2)$ for all iteration counters $k_i \geq 0$ is independent to all other randomness in the algorithm. This ensures that all noise added initially is gradually withdrawn from the whole network.

After the addition of noise, a standard Gossip update is made, which sets the values of sampled nodes to their average. Hence, we have

$$\begin{aligned}
\lim_{t\to\infty}\mathbb{E}\left[\left(\overline{c} - \frac{1}{n}\sum_{i=1}^n x_i^t\right)^2\right] &= \lim_{t\to\infty}\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&\leq \lim_{t\to\infty}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&= \frac{1}{n}\lim_{t\to\infty}\sum_{i=1}^n \mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&= \frac{1}{n}\lim_{t\to\infty}\sum_{i=1}^n \mathbb{E}\left[\phi_i^{2t_i-2}\right]\mathbb{E}\left[\left(v_i^{t_i-1}\right)^2\right] \\
&= \frac{1}{n}\lim_{t\to\infty}\sum_{i=1}^n \mathbb{E}\left[\phi_i^{2t_i-2}\right]\sigma_i^2 = \frac{1}{n}\sum_{i=1}^n \sigma_i^2 \lim_{t\to\infty}\mathbb{E}\left[\phi_i^{2t_i-2}\right] \\
&= 0,
\end{aligned}$$

as desired.

It is not the purpose of this work to define any quantifiable notion of protection of the initial values formally. However, we note that it is likely the case that the protection of private value c_i will be stronger for bigger σ_i and for ϕ_i closer to 1.

For simplicity, we provide only the primal algorithm below.

Algorithm 17 (Primal form)

Input: Vector of private values $c \in \mathbb{R}^n$; initial variances $\sigma_i^2 \in \mathbb{R}_+$ and variance decrease rate ϕ_i such that $0 \leq \phi_i < 1$ for all nodes i.

Initialize: Set $x^0 = c$; $t_1 = t_2 = \cdots = t_n = 0$, $v_1^{-1} = v_2^{-1} = \cdots = v_n^{-1} = 0$.

1: **for** $t = 0, 1, \ldots, k-1$ **do**

2: Choose edge $e = (i, j) \in \mathcal{E}$ uniformly at random.

3: Generate $v_i^{t_i} \sim N(0, \sigma_i^2)$ and $v_j^{t_j} \sim N(0, \sigma_j^2)$

4: Set

$$w_i^{t_i} = \phi_i^{t_i} v_i^{t_i} - \phi_i^{t_i - 1} v_i^{t_i - 1}$$
$$w_j^{t_j} = \phi_j^{t_j} v_j^{t_j} - \phi_j^{t_j - 1} v_j^{t_j - 1}$$

5: Update the primal variable:

$$x_i^{t+1} = x_j^{t+1} = \frac{x_i^t + w_i^{t_i} + x_j^t + w_j^{t_j}}{2}, \ \forall\, l \neq i, j : \ x_l^{t+1} = x_l^t$$

6: Set $t_i = t_i + 1$, $t_j = t_j + 1$

7: **end for**

8: **Return** x^k

We now provide results of dual analysis of Algorithm 17. The following lemma provides us the expected decrease in dual suboptimality for each iteration.

Lemma 48. *Let d_i denote the number of neighbours of node i. Then,*

$$\begin{aligned} \mathbb{E}\left[D(y^*) - D(y^{t+1})\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right) \mathbb{E}\left[D(y^*) - D(y^t)\right] + \frac{1}{4m} \sum_{i=1}^n d_i \sigma_i^2 \mathbb{E}\left[\phi_i^{2t_i}\right] \\ - \frac{1}{2m} \sum_{e \in \mathcal{E}} \mathbb{E}\left[\left(\phi_i^{t_i - 1} v_i^{t_i - 1} x_j^t + \phi_j^{t_j - 1} v_j^{t_j - 1} x_i^t\right)\right]. \end{aligned} \tag{5.18}$$

Proof. See Section 5.6.8 □

We use the lemma to prove our main result, in which we show linear convergence for the algorithm. For notational simplicity, we decided to have $\rho^t = (\rho)^t$, i.e. superscript of ρ denotes its power, not an iteration counter.

Theorem 49. *Let us define the following quantities:*

$$\begin{aligned} \rho &:= 1 - \frac{\alpha(\mathcal{G})}{2m}, \\ \psi^t &:= \frac{1}{\sum_{i=1}^n (d_i \sigma_i^2)} \sum_{i=1}^n d_i \sigma_i^2 \left(1 - \frac{d_i}{m}\left(1 - \phi_i^2\right)\right)^t. \end{aligned}$$

Then for all $k \geq 1$ we have the following bound

$$\mathbb{E}\left[D(y^*) - D(y^k)\right] \leq \rho^k \left(D(y^*) - D(y^0)\right) + \frac{\sum \left(d_i \sigma_i^2\right)}{4m} \sum_{t=1}^k \rho^{k-t} \psi^t.$$

Proof. See Section 5.6.9 □

Note that ψ^t is a weighted sum of t-th powers of real numbers smaller than one. For large enough t, this quantity will depend on the largest of these numbers. This brings us to define M as the set of indices i for which the quantity $1-\frac{d_i}{m}\left(1-\phi_i^2\right)$ is maximized:

$$M = \arg\max_i \left\{1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right\}.$$

Then for any $i_{\max} \in M$ we have

$$\psi^t \approx \frac{1}{\sum_{i=1}^n \left(d_i\sigma_i^2\right)} \sum_{i\in M} d_i\sigma_i^2 \left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t = \frac{\sum_{i\in M} d_i\sigma_i^2}{\sum_{i=1}^n \left(d_i\sigma_i^2\right)} \left(1-\frac{d_{i_{\max}}}{m}\left(1-\phi_{i_{\max}}^2\right)\right)^t,$$

which means that increasing ϕ_j for $j \notin M$ will not substantially influence convergence rate.

Note that as soon as we have

$$\rho > 1-\frac{d_i}{m}\left(1-\phi_i^2\right) \tag{5.19}$$

for all i, the rate from theorem 49 will be driven by ρ^k (as $k\to\infty$) and we will have

$$\mathbb{E}\left[D(y^*)-D(y^k)\right] = \tilde{O}\left(\rho^k\right) \tag{5.20}$$

One can think of the above as a threshold: if there is i such that ϕ_i is large enough so that the inequality (5.19) does not hold, the convergence rate is driven by $\phi_{i_{\max}}$. Otherwise, the rate is not influenced by the insertion of noise. Thus, in theory, we do not pay anything in terms of performance as long as we do not hit the threshold. One might be interested in choosing ϕ_i so that the threshold is attained for all i, and thus $M=\{1,\dots,n\}$. This motivates the following result:

Corollary 50. *Let us choose*

$$\phi_i := \sqrt{1-\frac{\gamma}{d_i}} \tag{5.21}$$

for all i, where $\gamma \le d_{\min}$. Then

$$\mathbb{E}\left[D(y^*)-D(y^k)\right] \quad\le\quad \left(1-\min\left(\frac{\alpha(\mathcal{G})}{2m},\frac{\gamma}{m}\right)\right)^k \left(D(y^*)-D(y^0)+\frac{\sum_{i=1}^n\left(d_i\sigma_i^2\right)}{4m}k\right).$$

As a consequence, $\phi_i = \sqrt{1-\frac{\alpha(\mathcal{G})}{2d_i}}$ is the largest decrease rate of noise for node i such that the guaranteed convergence rate of the algorithm is not violated.

Proof. See Section 5.6.10 □

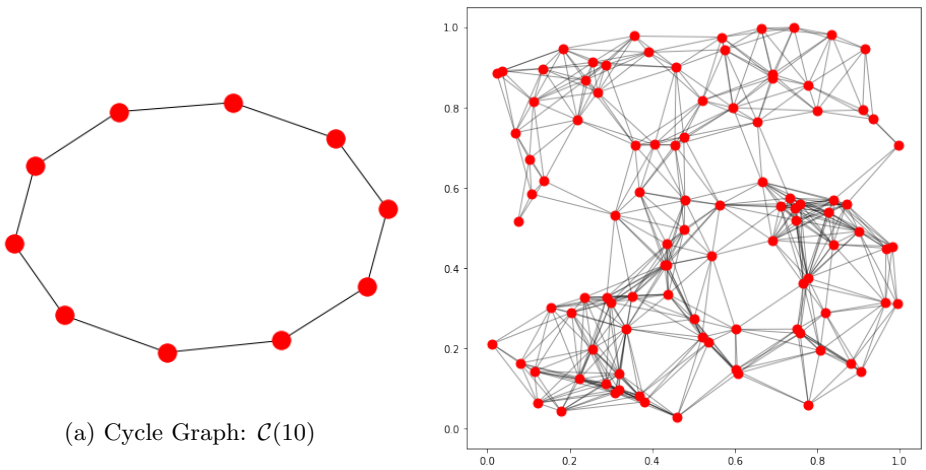
While the above result clearly states the important threshold, it is not always practical as $\alpha(\mathcal{G})$ might not be known. However, note that if we choose $\frac{nd_{\min}}{2(n-1)} \le \gamma \le d_{\min}$, we have $\min\left(\frac{\alpha(\mathcal{G})}{2m},\frac{\gamma}{m}\right) = \frac{\alpha(\mathcal{G})}{2m}$ since $\frac{\alpha(\mathcal{G})}{2} \le \frac{n}{n-1}\frac{d_{\min}}{2} \le \gamma$, where $e(\mathcal{G})$ denotes graph *edge connectivity*: the minimal number of edges to be removed so that the graph becomes disconnected. Inequality $\alpha(\mathcal{G}) \le \frac{n}{n-1}d_{\min}$ is a well known result in spectral graph theory [52]. As a consequence, if for all i we have

$$\phi_i \le \sqrt{1-\frac{(n-1)d_{\min}}{2nd_i}},$$

then the convergence rate is not driven by the noise.

5.4 Numerical Evaluation

We devote this section to experimentally evaluate the performance of the Algorithms 15, 16 and 17 we proposed in the previous sections, applied to the Average Consensus problem. In the experiments, we used two popular graph topologies the cycle graph (ring network) and the random geometric graph (see Figure 5.1 for an illustration of the two graphs).



(a) Cycle Graph: $\mathcal{C}(10)$

(b) Random Geometric Graph: $\mathcal{G}(n, r)$

Figure 5.1: Illustration of the two graph topologies we use in this section.

- *Cycle graph* with n nodes: $\mathcal{C}(n)$. In our experiments we choose $n = 10$. This small simple graph with regular topology is chosen for illustration purposes.

- *Random geometric graph* with n nodes and radius r: $\mathcal{G}(n, r)$. Random geometric graphs [152] are very important in practice because of their particular formulation which is ideal for modeling wireless sensor networks [76, 16]. In our experiments we focus on a 2-dimensional random geometric graph $\mathcal{G}(n, r)$ which is formed by placing n nodes uniformly at random in a unit square with edges between nodes which are having euclidean distance less than the given radius r. We set this to be to be $r = r(n) = \sqrt{\log(n)/n}$ — it is well know that the connectivity is preserved in this case [76]. We set $n = 100$.

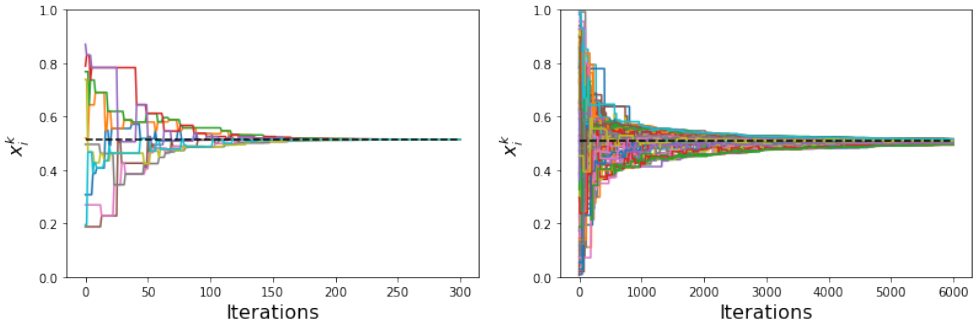
Setup: In all experiments we generate a vector with of initial values c_i from a uniform distribution over $[0, 1]$. We run several experiments and present two kinds of figures that help us to understand how the algorithms evolve and verify the theoretical results of the previous sections. These figures are:

1. The evolution of the initial values of the nodes. In these figures, we plot how the trajectory of the values x_i^t of each node i evolves throughout iterations. The black dotted horizontal line represents the exact average consensus value which all nodes should approach, and thus all other lines should approach this level.

2. The evolution of the relative error measure $\|x^t - x^*\|^2 / \|x^0 - x^*\|^2$ where $x^0 = c \in \mathbb{R}^n$ is the starting vector of the values of the nodes. In these figures we choose to have the relative error, both in normal and logarithmic scale on the vertical axis and the number of iterations on the horizontal axis.

For our evaluation we run each privacy preserving algorithm for several parameters and for a pre-specified number of iterations not necessarily the same for each experiment.

To illustrate the first concept (trajectories of the values x_i^t) , we provide a simple example of the evolution of the initial values x_i^t for the case of the Standard Gossip algorithm [16] in Figure 5.2. The horizontal black dotted line represents the average consensus value. It is the exact average of the initial values c_i of the nodes in the network.



(a) Cycle Graph (b) Random Geometric Graph

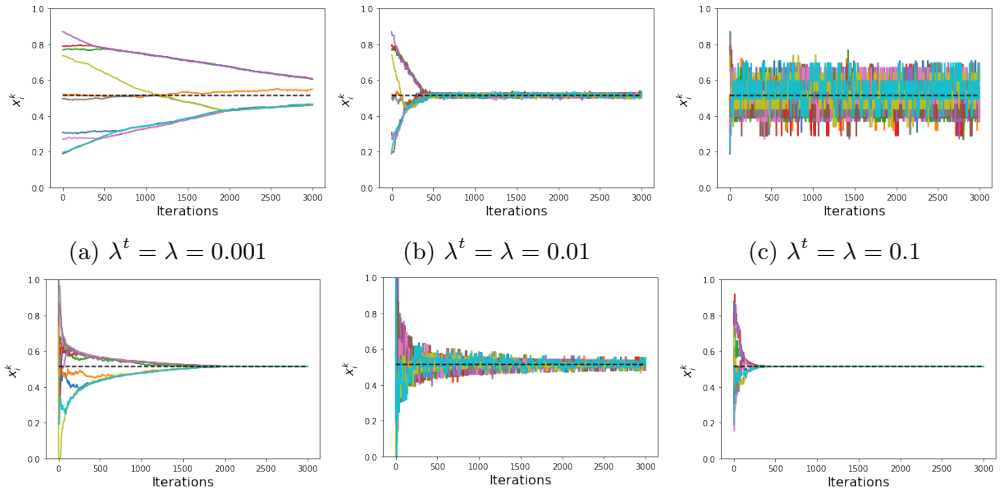
Figure 5.2: Trajectories of the values x_i^t for the Standard Gossip algorithm for Cycle Graph and a random geometric graph. Each line corresponds to value x_i of node $i \in \mathcal{V}$.

In the rest of this section we evaluate the performance of the three privacy preserving randomized gossip algorithms of Section 5.3, and contrast with the above Standard Gossip algorithm, which we refer to as "Baseline" in the following figures labels.

5.4.1 Private gossip via binary oracle

In this section, we evaluate the performance of Algorithm 15 presented in Section 5.3.2. In the algorithm, the input parameters are the positive stepsizes $\{\lambda^t\}_{t=0}^{\infty}$. The goal of the experiments is to compare the performance of the proposed algorithm using different choices of λ^t.

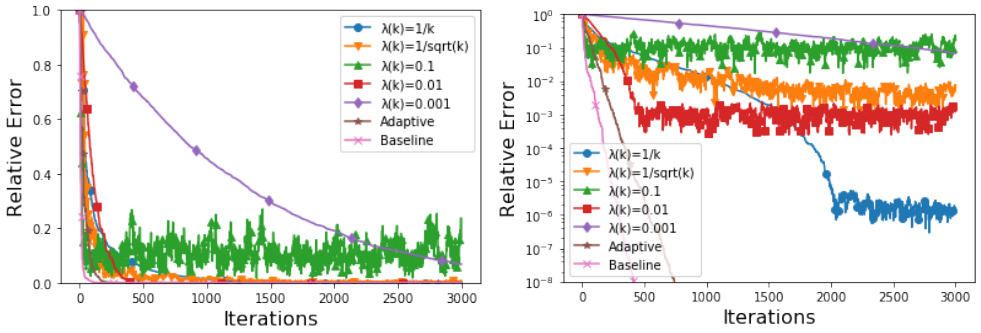
In particular, we use decreasing sequences of stepsizes $\lambda^t = 1/t$ and $\lambda^t = 1/\sqrt{t}$, and three different fixed values for the stepsizes $\lambda^t = \lambda \in \{0.001, 0.01, 0.1\}$. We also include the adaptive choice $\lambda^t = \frac{1}{4m}\sum_{e\in\mathcal{E}} |x_i^t - x_j^t|$ which we have proven to converge with linear rate in Theorem 45. We compare these choices in Figures 5.4 and 5.6, along with the Standard Gossip algorithm for clear comparison.



(a) $\lambda^t = \lambda = 0.001$ (b) $\lambda^t = \lambda = 0.01$ (c) $\lambda^t = \lambda = 0.1$

(d) $\lambda^t = \frac{1}{t}$ (e) $\lambda^t = \frac{1}{\sqrt{t}}$ (f) $\lambda^t =$ Adaptive

Figure 5.3: Trajectories of the values of x_i^t for Binary Oracle run on the cycle graph.



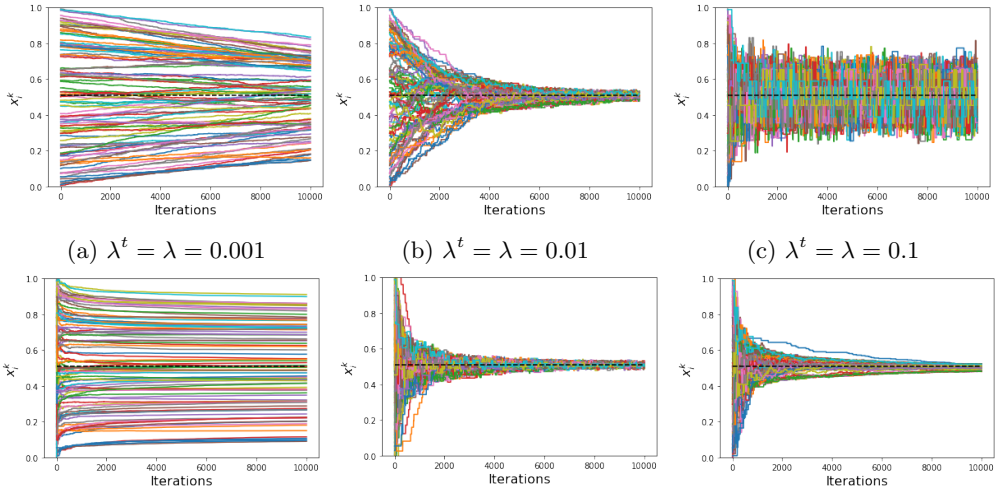
(a) Linear Scale

(b) Logarithmic Scale

Figure 5.4: Convergence of the Binary Oracle run on the cycle graph.

In general, we clearly see what is expected with the constant stepsizes — that they converge to a certain neighbourhood and oscillate around optimum. With smaller stepsize, this neighbourhood is more accurate, but it takes longer to reach. With decreasing stepsizes, Theorem 44 suggests that λ^t of order $1/\sqrt{t}$ should be optimal. Figure 5.6 demonstrates this, as the choice of $\lambda^t = 1/t$ decreases the stepsizes too quickly. However, this is not the case in Figure 5.4 in which we observe the opposite effect. This is due to the cycle graph being small and simple, and hence the diminishing stepsize becomes a problem only after a relatively large number of iterations. With the adaptive choice of stepsizes, we recover the linear convergence rate as predicted by Theorem 45.

The results in Figure 5.6 show one surprising comparison. The adaptive choice of stepsizes does not seem to perform better than $\lambda^t = 1/\sqrt{t}$. However, we verified that when running for more iterations, the linear rate of adaptive stepsize is present and converges significantly faster to higher accuracies. We chose to present the results for 6000 iterations since we found it overall cleaner.



(a) $\lambda^t = \lambda = 0.001$

(b) $\lambda^t = \lambda = 0.01$

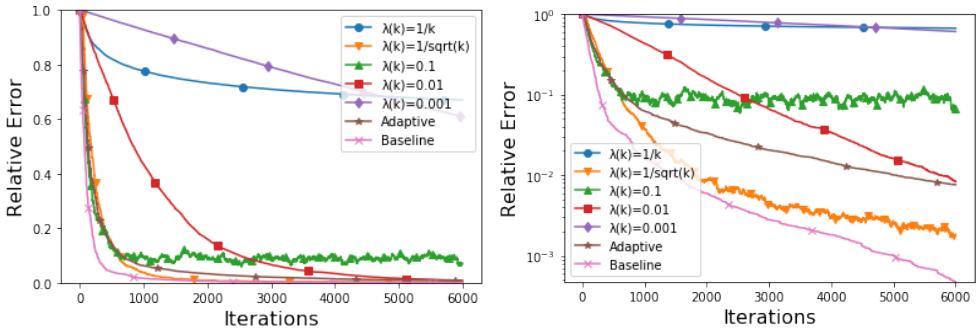
(c) $\lambda^t = \lambda = 0.1$

(d) $\lambda^t = \frac{1}{t}$

(e) $\lambda^t = \frac{1}{\sqrt{t}}$

(f) $\lambda^t =$ Adaptive

Figure 5.5: Trajectories of the values of x_i^t for Binary Oracle run on the random geometric graph.



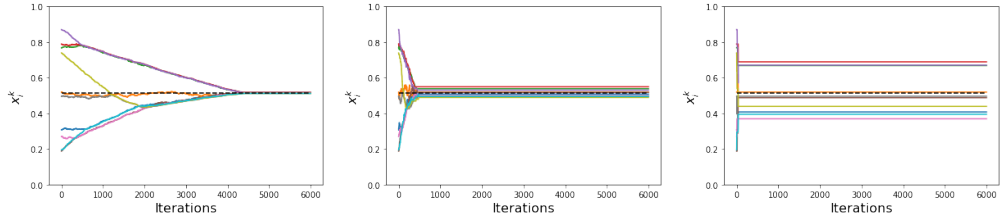
(a) Linear Scale (b) Logarithmic Scale

Figure 5.6: Convergence of the Binary Oracle run on the random geometric graph.

5.4.2 Private gossip via ϵ-gap oracle

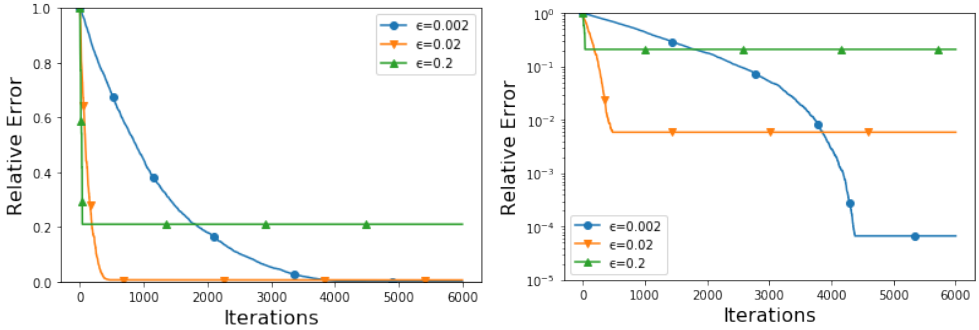
In this section, we evaluate the performance of the Algorithm 16 presented in Section 5.3.3. In the algorithm, the input parameter is the positive error tolerance variable ϵ. For experimental evaluation. we choose three different values for the input, $\epsilon \in \{0.2, 0.02, 0.002\}$, and again use the same cycle and random geometric graphs. The trajectories of the values x_i^t are presented in Figures 5.7 and 5.9, respectively. The performance of the algorithm in terms of the relative error is presented in Figures 5.8 and 5.10.

The performance is exactly matching the expectation — with larger ϵ, the method converges very fast to a wide neighbourhood of the optimum. For a small value, it converges much closer to the optimum, but it requires more iterations.



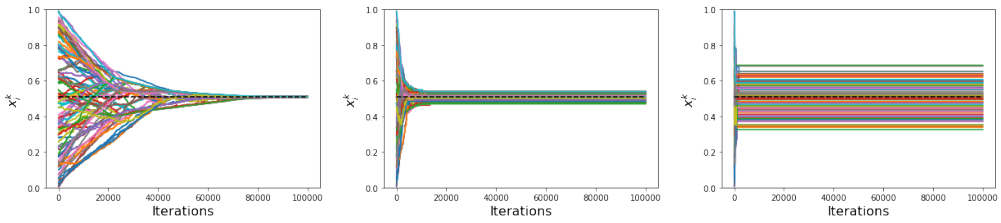
(a) $\epsilon = 0.002$ (b) $\epsilon = 0.02$ (c) $\epsilon = 0.2$

Figure 5.7: Trajectories of the values of x_i^t for ϵ-Gap Oracle run on the cycle graph.



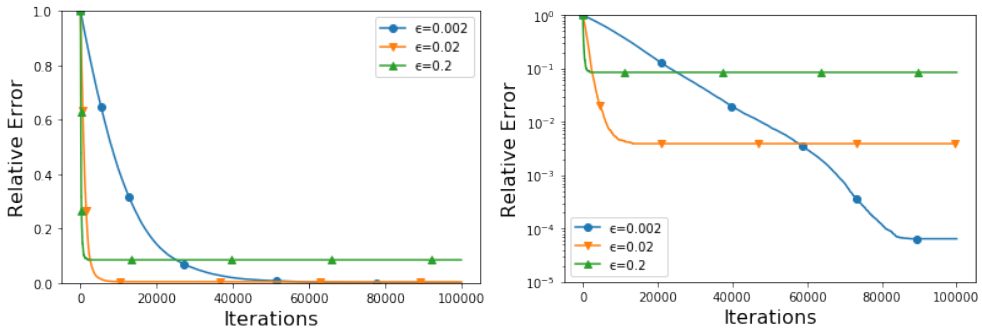
(a) Linear Scale (b) Logarithmic Scale

Figure 5.8: Convergence of the ϵ-Gap Oracle run on the cycle graph.



(a) $\epsilon = 0.002$ (b) $\epsilon = 0.02$ (c) $\epsilon = 0.2$

Figure 5.9: Trajectories of the values of x_i^t for ϵ-Gap Oracle run on the random geometric graph.



(a) Linear Scale (b) Logarithmic Scale

Figure 5.10: Convergence of the ϵ-Gap Oracle run on the random geometric graph.

5.4.3 Private gossip via controlled noise insertion

In this section, we evaluate the performance of Algorithm 17 presented in Section 5.3.4. This algorithm has two different parameters for each node i. These are the initial variance $\sigma_i^2 \geq 0$ and the rate of decay, ϕ_i, of the noise.

To evaluate the impact of these parameters, we perform several experiments. As earlier, we use the same graph structures for evaluation: cycle graph and random geometric graph. The algorithm converges with a linear rate depending on the minimum of two factors — see Theorem 49 and Corollary 50. We will verify that this is indeed the case, and for values of ϕ_i above a certain threshold, the convergence is driven by the rate at which the noise decays. This is true for both identical values of ϕ_i for all i, and for varying values as per (5.21). We further demonstrate the latter is superior in the sense that it enables insertion of more noise, without sacrificing the convergence speed. Finally, we study the effect of various magnitudes of the noise inserted initially.

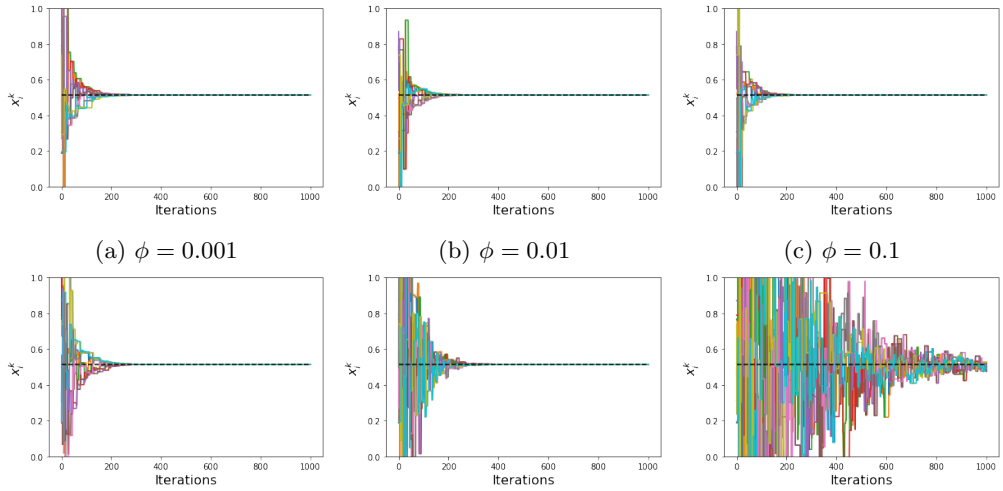
Fixed variance, identical decay rates

In this part, we run Algorithm 17 with $\sigma_i = 1$ for all i, and set $\phi_i = \phi$ for all i and some ϕ. We study the effect of varying the value of ϕ on the convergence of the algorithm.

In both Figures 5.12b and 5.14b, we see that for small values of ϕ, we eventually recover the same rate of linear convergence as the Standard Gossip algorithm. If the value of ϕ is sufficiently close to 1 however, the rate is driven by the noise and not by the convergence of the Standard Gossip algorithm. This value is $\phi = 0.98$ for cycle graph, and $\phi = 0.995$ for the random geometric graph in the plots we present.

Looking at the individual runs for small values of ϕ in Figure 5.14b, we see some variance in terms of when the asymptotic rate is realized. We would like to point out that this *does not* provide additional insight into whether specific small values of ϕ are in general better for the following reason. The Standard Gossip algorithm is itself a randomized algorithm, with an inherent uncertainty in the convergence of any particular run. If we ran the algorithms multiple times, we observe variance in the evolution of the suboptimality of similar magnitude, just as

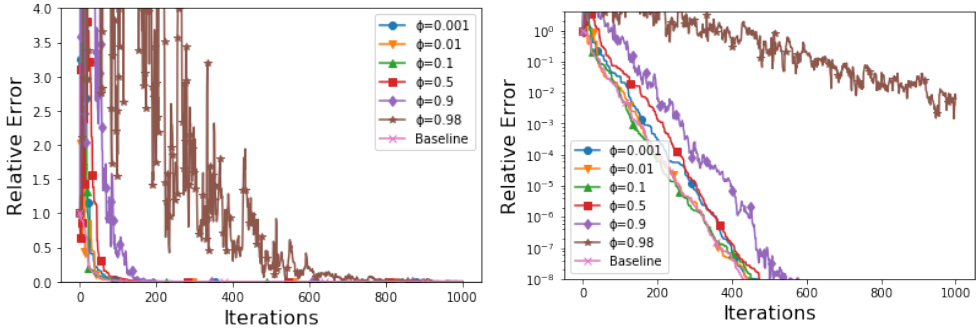
what we see in the figure. Hence, the variance is expected, and not significantly influenced by the noise.



(a) $\phi = 0.001$ (b) $\phi = 0.01$ (c) $\phi = 0.1$

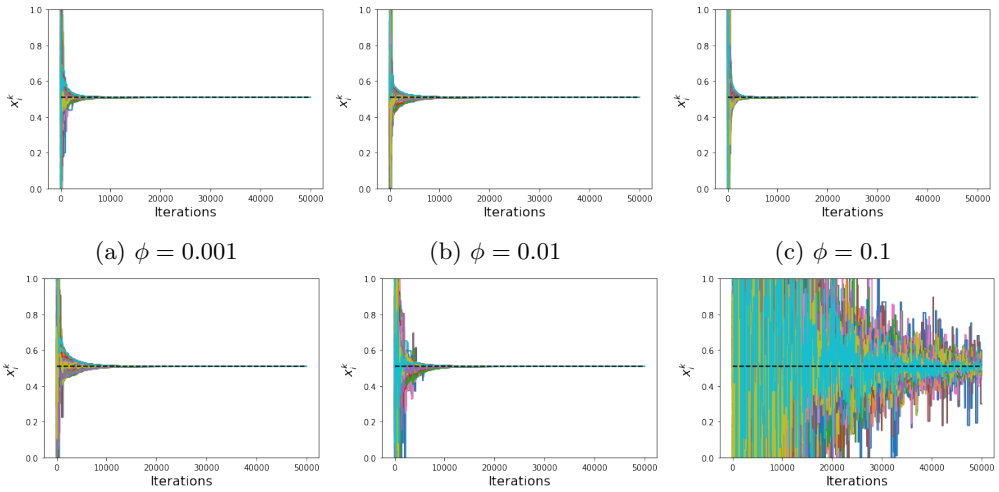
(d) $\phi = 0.5$ (e) $\phi = 0.9$ (f) $\phi = 0.98$

Figure 5.11: Trajectories of the values of x_i^t for Controlled Noise Insertion run on the cycle graph for different values of ϕ.



(a) Linear Scale (b) Logarithmic Scale

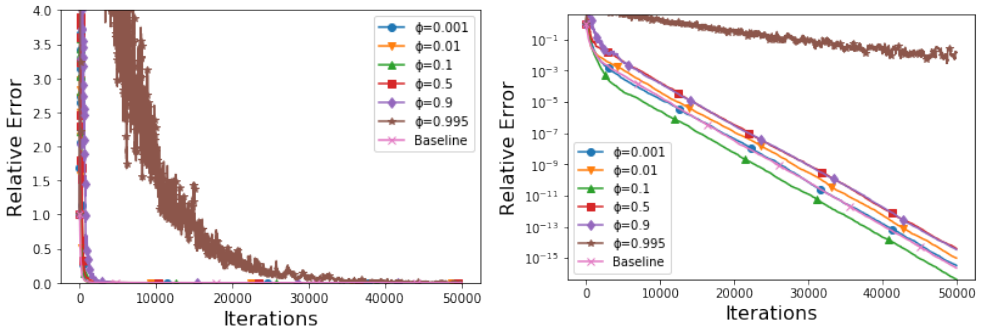
Figure 5.12: Convergence of the Controlled Noise Insertion run on the cycle graph for different values of ϕ.



(a) $\phi = 0.001$ (b) $\phi = 0.01$ (c) $\phi = 0.1$

(d) $\phi = 0.5$ (e) $\phi = 0.9$ (f) $\phi = 0.995$

Figure 5.13: Trajectories of the values of x_i^t for Controlled Noise Insertion run on the random geometric graph for different values of ϕ.



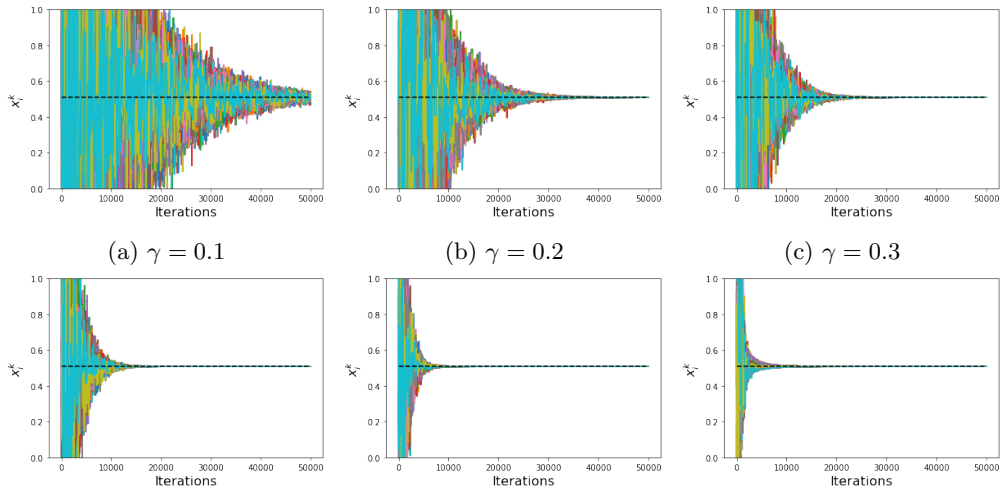
(a) Linear Scale (b) Logarithmic Scale

Figure 5.14: Convergence of the Controlled Noise Insertion run on the random geometric graph for different values of ϕ.

Variance 1 and different decay rates

In this section, we perform a similar experiment as above, but the values ϕ_i are not all the same. We rather control them by the choice of γ as in (5.21). Note that by decreasing γ, we increase ϕ_i, and thus smaller γ means the noise decays at a slower rate. Here, due to the regular structure of the cycle graph, we present only results for the random geometric graph.

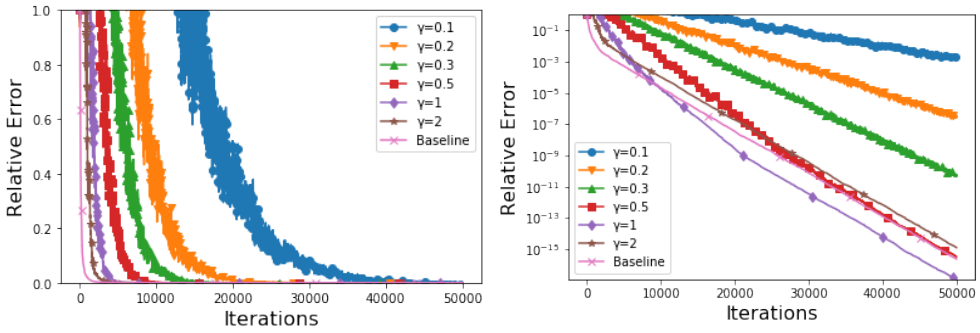
It is not straightforward to compare this setting with the setting of identical ϕ_i, and we return to it in the next section. Here we only remark that we again see the existence of a threshold predicted by theory, beyond which the convergence is dominated by the inserted noise. Otherwise, we recover the rate of the Standard Gossip algorithm.



(a) $\gamma = 0.1$ (b) $\gamma = 0.2$ (c) $\gamma = 0.3$

(d) $\gamma = 0.5$ (e) $\gamma = 1$ (f) $\gamma = 2$

Figure 5.15: Trajectories of the values of x_i^t for Controlled Noise Insertion run on the random geometric graph for different values of ϕ_i, controlled by γ.



(a) Normal Scale (b) Logarithmic Scale

Figure 5.16: Convergence of the Controlled Noise Insertion run on the random geometric graph for different values of ϕ_i, controlled by γ.

Impact of varying ϕ_i

In this experiment, we demonstrate the practical utility of letting the rate of decay ϕ_i to be different on each node i. In order to do so, we run the experiment on the random geometric graph and compare the settings investigated in the previous two sections — the noise decay rate driven by ϕ, or by γ.

In first place, we choose the values of ϕ_i such that that the two factors in Corollary 50 are equal. For the particular graph we used, this corresponds to $\gamma \approx 0.17$ with $\phi_i = \sqrt{1 - \frac{\alpha(\mathcal{G})}{2d_i}}$. Second, we make the factors equal, but with constraint of having ϕ_i to be equal for all i. This corresponds to $\phi_i \approx 0.983$ for all i.

The performance for a large number of iterations is displayed in the left side of Figure 5.17. We see that the above two choices indeed yield very similar practical performance, which also eventually matches the rate predicted by theory. For a complete comparison, we also include the performance of the Standard Gossip algorithm.

The important message is conveyed in the histogram in the right side of Figure 5.17. The histogram shows the distribution of the values of ϕ_i for different nodes i. The minimum of these values is what we needed in the case of identical ϕ_i for all i. However, most of the values are significantly higher. This means, that if we allow the noise decay rates to depend on the number of neighbours, we are able to increase the amount of noise inserted, without sacrificing

practical performance. This is beneficial, as more noise will likely be beneficial for any formal notion of protection of the initial values.

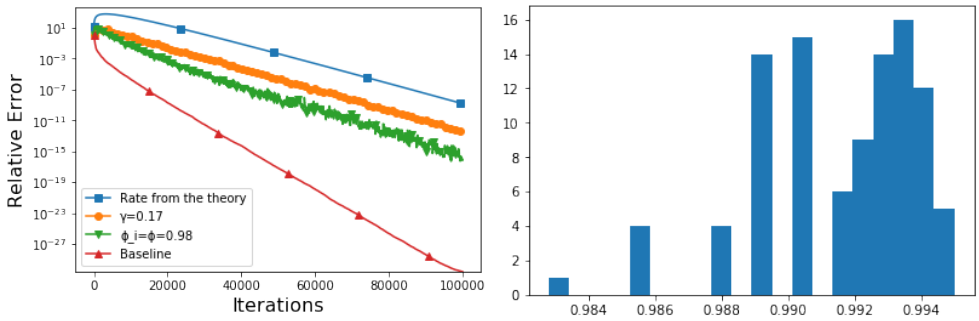


Figure 5.17: Left: Performance of the noise oracle with noise decrease rate chosen according to Corollary 50. Right: Histogram of of distribution of ϕ_i

5.5 Conclusion

In this chapter, we addressed the Average Consensus problem via novel asynchronous privacy preserving randomized gossip algorithms. In particular, we propose three different algorithmic tools for the protection of the initial private values of the nodes.

The first two proposed algorithms "Private Gossip via Binary Oracle" and "Private Gossip via ϵ-Gap Oracle" are based on the same idea of weakening the oracle used in the gossip update rule. In these two protocols the chosen pair of nodes of each gossip step instead of share their exact values they provide only categorical (or even binary) information to each other.

In the third protocol "Private Gossip via Controlled Noise Insertion", we systematically inject and withdraw noise throughout the iterations, so as to ensure convergence to the average consensus value and at the same time protect the private information of the nodes.

In all cases, we provide explicit convergence rates and evaluate practical convergence on common simulated network topologies.

Future work inludes the design of privacy preserving variants of several popular and fast gossip protocols [7, 127, 82, 19, 106]. One can also investigate more challenging types of consensus problems like the finite step consensus or consensus on networks with time-varying topology, and design gossip protocols that preserve the privacy of the participating agents. Designing the optimal network structure for information preservation is also an interesting research direction.

As we have already mentioned the gossip algorithms of this chapter do not address any specific notion of privacy (no clear measure of privacy is presented) and it is still not clear how the formal concept of differential privacy [47] can be applied in protocols for solving the average consensus problem. Propose efficient differential privacy guarantees for gossip protocols in general graphs is an interesting open problem.

5.6 Proofs of Main Results

5.6.1 Proof of Lemma 41

Let us first present a lemma that we use in the proof of Lemma 41.

Lemma 51. *The eigenvalues of* $\tilde{\mathbf{L}} = n\mathbf{I} - \mathbf{1}\mathbf{1}^\top$ *are* $\{0, n, n, \ldots, n\}$

Proof. Clearly, $\tilde{\mathbf{L}}\mathbf{1} = 0$. Consider some vector x such that $\langle x, \mathbf{1}\rangle = 0$. Then, $\tilde{\mathbf{L}}x = n\mathbf{I}x - \mathbf{1}\mathbf{1}^\top x = nx + \mathbf{1}\mathbf{1}^\top x = nx$ thus x is an eigenvector corresponding to eigenvalue n. Thus, we can pick $n-1$ linearly independent eigenvectors of $\tilde{\mathbf{L}}$ corresponding to eigenvalue n, which concludes the proof. □

Having established the above lemma let us present the proof Lemma 41.

The Laplacian matrix of $\mathcal{G}$ is the matrix $\mathbf{L} = \mathbf{A}^\top \mathbf{A}$. We have $\mathbf{L}_{ii} = d_i$ (degree of vertex i), $\mathbf{L}_{ij} = \mathbf{L}_{ji} = -1$ if $(i,j) \in \mathcal{E}$ and $\mathbf{L}_{ij} = 0$ otherwise. A simple computation reveals that for any $x \in \mathbb{R}^n$ we have

$$x^\top \mathbf{L} x = \sum_{e=(i,j)\in\mathcal{E}} (x_i - x_j)^2.$$

Let $\tilde{\mathbf{A}}$ be the $n(n-1)/2 \times n$ matrix corresponding to the complete graph $\tilde{\mathcal{G}}$ on $\mathcal{V}$. Let $\tilde{\mathbf{L}} = \tilde{\mathbf{A}}^\top \tilde{\mathbf{A}}$ be its Laplacian. We have $\tilde{\mathbf{L}}_{ii} = n-1$ for all i and $\tilde{\mathbf{L}}_{ij} = -1$ for $i \neq j$. So, $\tilde{\mathbf{L}} = n\mathbf{I} - \mathbf{1}\mathbf{1}^\top$. Then

$$x^\top \tilde{\mathbf{L}} x = n\|x\|^2 - \left(\sum_{i=1}^n x_i\right)^2 = \sum_{(i,j)} (x_i - x_j)^2.$$

Inequality (5.6) can therefore be recast as follows:

$$x^\top (n\mathbf{I} - \mathbf{1}\mathbf{1}^\top) x \leq x^\top \beta(\mathcal{G}) \mathbf{L} x, \qquad x \in \mathbb{R}^n.$$

Let $\beta = \beta(\mathcal{G})$. Note that both $\tilde{\mathbf{L}}$ and $\beta\mathbf{L}$ are Hermitian thus have real eigenvalues and there exist an orthonormal basis of their eigenvectors. Suppose that $\{x_1, \dots x_n\}$ are eigenvectors of βL corresponding to eigenvalues $\lambda_1(\beta\mathbf{L}), \lambda_2(\beta\mathbf{L}) \dots, \lambda_n(\beta\mathbf{L})$. Without loss of generality assume that these eigenvectors form an orthonormal basis and $\lambda_1(\beta\mathbf{L}) \geq \cdots \geq \lambda_n(\beta\mathbf{L})$.

Clearly, $\lambda_n(\beta\mathbf{L}) = 0$, $x_n = \mathbf{1}/\sqrt{n}$, and $\lambda_{n-1}(\beta\mathbf{L}) = n$. Lemma 51 states that eigenvalues of $\tilde{\mathbf{L}}$ are $\{0, n, n, \dots, n\}$.

One can easily see that eigenvector corresponding to zero eigenvalue of $\tilde{\mathbf{L}}$ is x_n. Note that eigenvectors $x_1, \dots, x_{n-1}$ generate an eigenspace corresponding to eigenvalue n of $\tilde{\mathbf{L}}$.

Consider some $x = \sum_{i=1}^n c_i x_i$, $c_i \in \mathbb{R}$ for all i. Then we have

$$x^\top \tilde{\mathbf{L}} x = \sum_{i=1}^n \lambda_i \left(\tilde{\mathbf{L}}\right) c_i^2 \leq \sum_{i=1}^n \lambda_i \left(\beta\mathbf{L}\right) c_i^2 = x^\top \beta \mathbf{L} x,$$

which concludes the proof.

5.6.2 Proof of Theorem 42

We first establish two lemmas which will be needed to prove Theorem 42.

Lemma 52. *Assume that edge $e = (i,j)$ is selected in iteration t of Algorithm 14. Then*

$$D(y^{t+1}) - D(y^t) = \frac{1}{4}(x_i^t - x_j^t)^2. \tag{5.22}$$

Proof. We have $y^{t+1} = y^t + \lambda^t f_e$ where λ^t is chosen so that $D(y^{t+1}) - D(y^t)$ is maximized. Applying Lemma 40, we have

$$D(y^{t+1}) - D(y^t) = \max_\lambda -\lambda(x_i^t - x_j^t) - \lambda^2 = \frac{1}{4}(x_i^t - x_j^t)^2.$$

□

Lemma 53. *Let $x \in \mathbb{R}^n$ such that $\frac{1}{n}\sum_i x_i = \bar{c}$. Then*

$$\frac{1}{2}\|\bar{c}\mathbf{1} - x\|^2 \leq \frac{1}{2\alpha(\mathcal{G})} \sum_{e=(i,j)\in\mathcal{E}} (x_i - x_j)^2. \tag{5.23}$$

Proof.

$$\frac{1}{2}\|\bar{c}\mathbf{1}-x\|^2 \overset{(5.9)}{=} \frac{1}{4n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j-x_i)^2 = \frac{1}{2n}\sum_{(i,j)}(x_j-x_i)^2$$
$$\overset{(5.6)}{\leq} \frac{\beta(\mathcal{G})}{2n}\sum_{e=(i,j)\in\mathcal{E}}(x_i-x_j)^2 \overset{\text{Lemma 41}}{=} \frac{1}{2\alpha(\mathcal{G})}\sum_{e=(i,j)\in\mathcal{E}}(x_i-x_j)^2$$

□

Having established Lemmas 52 and 53, we can now proceed with the proof of Theorem 42:

$$\begin{aligned}
\mathbb{E}\left[D(y^*)-D(y^{t+1}) \mid y^t\right] &= D(y^*)-D(y^t)-\mathbb{E}\left[D(y^{t+1})-D(y^t)\mid y^t\right] \\
&\overset{(5.22)}{=} D(y^*)-D(y^t)-\sum_{e=(i,j)\in\mathcal{E}}\frac{1}{4m}(x_i^t-x_j^t)^2 \\
&\overset{(5.8)}{=} \frac{1}{2}\|\bar{c}\mathbf{1}-x^t\|^2-\sum_{e=(i,j)\in\mathcal{E}}\frac{1}{4m}(x_i^t-x_j^t)^2 \\
&\overset{(5.23)}{\leq} \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\frac{1}{2}\|\bar{c}\mathbf{1}-x^t\|^2 \\
&\overset{(5.8)}{=} \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\left(D(y^*)-D(y^t)\right) .
\end{aligned}$$

Taking expectation again, we get the recursion

$$\mathbb{E}\left[D(y^*)-D(y^{t+1})\right] \leq \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right].$$

5.6.3 Proof of Lemma 43

Let us first present a Lemma that we use in the proof of Lemma 43.

Lemma 54.

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{n}(x_j-x_i)\right)^2 = \frac{n}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j-x_i)^2 \tag{5.24}$$

Proof. Using simple algebra we have

$$\begin{aligned}
\sum_{i=1}^{n}\left(\sum_{j=1}^{n}(x_j-x_i)\right)^2 &= \sum_{i=1}^{n}\left(\sum_{j=1}^{n}x_j-nx_i\right)^2 \\
&= \sum_{i=1}^{n}\left(\left(\sum_{j=1}^{n}x_j\right)^2+n^2x_i^2-2nx_i\left(\sum_{j=1}^{n}x_j\right)\right) \\
&= n\left(\sum_{j=1}^{n}x_j\right)^2+n^2\sum_{i=1}^{n}x_i^2-2n\left(\sum_{j=1}^{n}x_j\right)^2 \\
&= n^2\sum_{i=1}^{n}x_i^2-n\left(\sum_{i=1}^{n}x_i\right)^2 .
\end{aligned}$$

Manipulating right hand side of (5.24) we obtain

$$\begin{aligned}\frac{n}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j-x_i)^2 &= \frac{n}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_j^2+x_i^2-2x_ix_j\right)\\ &= n^2\sum_{i=1}^{n}x_i^2-n\sum_{i=1}^{n}\sum_{j=1}^{n}x_ix_j \quad = \quad n^2\sum_{i=1}^{n}x_i^2-n\left(\sum_{i=1}^{n}x_i\right)^2.\end{aligned}$$

Clearly, LHS and RHS of (5.24) are equal. □

In order to show (5.9) it is enough to notice that

$$\begin{aligned}\|\bar{c}\mathbf{1}-x\|^2 &= \sum_{i=1}^{n}(\bar{c}-x_i)^2 \quad = \quad \sum_{i=1}^{n}\left(\frac{1}{n}\sum_{j=1}^{n}x_j-x_i\right)^2\\ &= \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{1}{n}(x_j-x_i)\right)^2 \overset{(5.24)}{=} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{n}(x_j-x_i)^2\\ &= \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j-x_i)^2.\end{aligned}$$

Note that we have

$$\frac{1}{nm}\left(\sum_{e=(i,j)\in\mathcal{E}}|x_i-x_j|\right)^2 \le \frac{1}{n}\sum_{e\in\mathcal{E}}(x_i-x_j)^2 \le \frac{1}{n}\sum_{(i,j)}(x_i-x_j)^2 \overset{(5.9)}{=} \|\bar{c}\mathbf{1}-x\|^2,$$

which proves (5.10). On the other hand, we have

$$\frac{1}{\alpha(\mathcal{G})}\left(\sum_{e=(i,j)\in\mathcal{E}}|x_i-x_j|\right)^2 \ge \frac{1}{\alpha(\mathcal{G})}\sum_{e\in\mathcal{E}}(x_i-x_j)^2 \overset{(5.23)}{\ge} \|\bar{c}\mathbf{1}-x\|^2,$$

which concludes (5.11). Inequality (5.12) holds trivially.

5.6.4 Proof of Theorem 44

The following lemma is used in the proof of Theorem 44.

Lemma 55. *Fix $k \ge 0$ and let $R > 0$. Then*

$$\min_{\lambda=(\lambda^0,\dots,\lambda^k)\in\mathbb{R}^{k+1}}\frac{R+\beta^k}{\alpha^k} = 2\sqrt{\frac{R}{k+1}},$$

and the optimal solution is given by $\lambda^t = \sqrt{\frac{R}{k+1}}$ for all t.

Proof. Define $\phi(\lambda) = \frac{R+\beta^k}{\alpha^k}$. If we write $\lambda = rx$, where $r = \|\lambda\|$ and x is of unit norm, then $\phi(tx) = \frac{R+r^2}{r\langle\mathbf{1},x\rangle}$. Clearly, for any fixed r, the $x \in \mathbb{R}^{k+1}$ minimizing $x \mapsto \phi(rx)$ is $x = \mathbf{1}/\|\mathbf{1}\|$, where $\mathbf{1}$ is the vector of ones in $\mathbb{R}^{k+1}$. It now only remains to minimize the function $r \mapsto \frac{R+r^2}{r\|\mathbf{1}\|}$. This function is convex and differentiable. Setting the derivative to zero leads to $r = \sqrt{R}$. Combining the above, we get the optimal solution $\lambda = \frac{r}{\|\mathbf{1}\|}\mathbf{1} = \frac{\sqrt{R}}{\|\mathbf{1}\|}\mathbf{1}$. □

Let $e = (i,j)$ be the edge selected at iteration $t \ge 0$. Applying Lemma 40, we see that $D(y^{t+1}) - D(y^t) = \lambda^t|x_i^t - x_j^t| - (\lambda^t)^2$. Taking expectation with respect to edge selection, we

get

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid y^t\right] = -\left(\lambda^t\right)^2 + \lambda^t \cdot \frac{1}{m} \sum_{e=(i,j)\in\mathcal{E}} |x_i^t - x_j^t|,$$

and taking expectation again and using the tower property, we get the identity

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] = -\left(\lambda^t\right)^2 + \lambda^t \cdot \mathbb{E}\left[L^t\right].$$

Therefore,

$$\begin{aligned} D(y^*) - D(y^0) &\geq \mathbb{E}\left[D(y^{k+1}) - D(y^0)\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{k} D(y^{t+1}) - D(y^t)\right] \\ &= \sum_{t=0}^{k} \mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] \quad = \quad -\sum_{t=0}^{k}\left(\lambda^t\right)^2 + \sum_{t=0}^{k} \lambda^t \cdot \mathbb{E}\left[L^t\right]. \end{aligned}$$

It remains to reshuffle the resulting inequality to obtain (5.14).

We can see that part (i) follows directly. Optimality of stepsizes in (ii) is due to Lemma 55. To show (iii) we should state that

$$\alpha^k = \sum_{t=0}^{k} \lambda^k = \sum_{1}^{k+1} \frac{a}{\sqrt{t}} \geq a \int_{t=1}^{k+2} t^{-1/2} dt = 2a\left(\sqrt{k+2} - 1\right)$$

$$\beta^k = \sum_{t=0}^{k}\left(\lambda^k\right)^2 = \sum_{t=1}^{k+1} \frac{a^2}{t} \leq a^2 \int_{1/2}^{k+3/2} t^{-1} dt = a^2\left(\log(k+3/2) + \log(2)\right)$$

The inequality above holds due to the fact that for $t > 1/2$ we have $t^{-1} \leq \int_{t-1/2}^{t+1/2} x^{-1} dx$ since x^{-1} is convex function.

5.6.5 Proof of Theorem 45

Using Lemma 40 with we have

$$\begin{aligned} \mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid y^t\right] &= -\left(\lambda^t\right)^2 + \lambda^t \frac{1}{m} \sum_{e\in\mathcal{E}} |x_i^t - x_j^t| \\ &= \frac{1}{4m^2}\left(\sum_{e\in\mathcal{E}} |x_i^t - x_j^t|\right)^2 \quad \geq \quad \frac{1}{4m^2} \sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2. \end{aligned}$$

Taking the expectation again we obtain

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] \geq \frac{1}{4m^2}\mathbb{E}\left[\sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2\right]. \tag{5.25}$$

On the other hand, we have

$$\begin{aligned} D(y^{t+1}) - D(y^t) &= \big(D(y^{t+1}) - D(y^*)\big) + \big(D(y^*) - D(y^t)\big) \\ &= \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \\ &= \frac{\alpha(\mathcal{G})}{4m^2}\|\bar{c}\mathbf{1} - x^t\|^2 + \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)\frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 \\ &\quad - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \\ &\overset{(5.23)}{\leq} \frac{1}{4m^2}\sum_{e=(i,j)\in\mathcal{E}}\big(x_i^t - x_j^t\big)^2 + \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)\frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 \\ &\quad - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2. \end{aligned}$$

Taking the expectation of the above and combining with (5.25) we obtain the desired recursion

$$\mathbb{E}\left[\|\bar{c}\mathbf{1} - x^{t+1}\|^2\right] \quad \leq \quad \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)\mathbb{E}\left[\|\bar{c}\mathbf{1} - x^t\|^2\right].$$

5.6.6 Proof of Lemma 46

Let $e = (i, j)$ be the edge selected at iteration t. Applying Lemma 40, we see that

$$D(y^{t+1}) - D(y^t) = \begin{cases} -\frac{\epsilon}{2}(x_i^t - x_j^t) - \frac{\epsilon^2}{4}, & x_i^t - x_j^t \leq -\epsilon \\ \frac{\epsilon}{2}(x_i^t - x_j^t) - \frac{\epsilon^2}{4}, & x_j^t - x_i^t \leq -\epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

This implies that

$$D(y^{t+1}) - D(y^t) \begin{cases} \geq \frac{\epsilon^2}{4}, & \text{if } \Delta_e^t = 1, \\ = 0, & \text{if } \Delta_e^t = 0. \end{cases}$$

Taking expectation in the selection of e, we get

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid y^t\right] \geq \frac{\epsilon^2}{4} \cdot \mathbb{P}(\Delta_e^t = 1 \mid y^t) + 0 \cdot \mathbb{P}(\Delta_e^t = 0 \mid y^t) = \frac{\epsilon^2}{4}\Delta^t.$$

It remains to take expectation again.

5.6.7 Proof of Theorem 47

Since for all $k \geq 0$ we have $D(y^k) \leq D(y^*)$, it follows that

$$D(y^*) - D(y^0) \geq \mathbb{E}\left[D(y^k) - D(y^0)\right] = \mathbb{E}\left[\sum_{t=0}^{k-1} D(y^{t+1}) - D(y^t)\right] = \sum_{t=0}^{k-1}\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right].$$

It remains to apply Lemma 46.

5.6.8 Proof of Lemma 48

Let us first present three lemmas that we use in the proof of Lemma 48.

Lemma 56. *Suppose that we run Algorithm 17 for t iterations and t_i denotes the number of times that some edge corresponding to node i was selected during the algorithm.*

1. *$v_i^{t_i}$ and t_j are independent for all (i.e., not necessarily distinct) i, j.*
2. *$v_i^{t_i}$ and $\phi_j^{t_j}$ are independent for all (i.e., not necessarily distinct) i, j.*

3. $w_i^{t_i}$ *and* $w_j^{t_j}$ *have zero correlation for all* $i \neq j$.

4. x_j^t *and* $\phi_i^{t_i} v_i^{t_i}$ *have zero correlation for all (i.e., not necessarily distinct)* i, j.

Proof. 1. Follows from the definition of v_i^t.

2. Follows from the definition of v_i^t.

3. Note that we have $w_i^{t_i} = \phi_i^{t_i} v_i^{t_i} - \phi_i^{t_i-1} v_i^{t_i-1}$ and $w_j^{t_j} = \phi_j^{t_j} v_j^{t_j} - \phi_j^{t_j-1} v_j^{t_j-1}$. Clearly, $v_i^{t_i}$ and $w_j^{t_j}$ have zero correlation. Similarly $v_i^{t_i-1}$ and $w_j^{t_j}$ have zero correlation. Thus, $w_i^{t_i}$ and $w_j^{t_j}$ have zero correlation.

4. Clearly, x_j^t is a function initial state and all instances of random variables up to the iteration t. Thus, $v_i^{t_i}$ is independent to x_j^t from the definition. Thus, x_j^t and $\phi_i^{t_i} v_i^{t_i}$ have zero correlation.

□

Lemma 57.

$$\mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1} x_i^t\right] = \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1}\right)^2\right]. \tag{5.26}$$

Proof.

$$\begin{aligned}
&\mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1} x_i^t\right] \\
&= \mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1}\left(\left(x_i^t - \frac{\phi_i^{t_i-1} v_i^{t_i-1}}{2}\right) + \frac{\phi_i^{t_i-1} v_i^{t_i-1}}{2}\right)\right] \\
&= \mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1}\left(x_i^t - \frac{\phi_i^{t_i-1} v_i^{t_i-1}}{2}\right)\right] + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1}\right)^2\right] \\
&\overset{(*)}{=} \mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1}\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + w^{t_i-1} + w^{t_l^0} - \phi_i^{t_i-1} v_i^{t_i-1}}{2}\right)\right] + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1}\right)^2\right] \\
&\overset{(5.17)}{=} \mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1}\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + \phi_i^{t_i-1} v_i^{t_i-1} - \phi_i^{t_i-2} v_i^{t_i-2}}{2}\right)\right] \\
&\quad + \mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1}\left(\frac{\phi_i^{t_l^0} v_l^{t_l^0} - \phi_i^{t_l^0-1} v_l^{t_l^0-1} - \phi_i^{t_i-1} v_i^{t_i-1}}{2}\right)\right] \\
&\quad + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1}\right)^2\right] \\
&= \mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1}\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + \phi_i^{t_l^0} v_l^{t_l^0} - \phi_i^{t_l^0-1} v_l^{t_l^0-1} - \phi_i^{t_i-2} v_i^{t_i-2}}{2}\right)\right] \\
&\quad + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1}\right)^2\right] \\
&\overset{L.56}{=} \cancel{\mathbb{E}\left[\phi_i^{t_i-1} v_i^{t_i-1}\right]}\,\mathbb{E}\left[\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + \phi_i^{t_l^0} v_l^{t_l^0} - \phi_i^{t_l^0-1} v_l^{t_l^0-1} - \phi_i^{t_i-2} v_i^{t_i-2}}{2}\right)\right] \\
&\quad + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1}\right)^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1}\right)^2\right],
\end{aligned}$$

where in the first equality we add and subtracting $\frac{\phi_i^{t_i-1} v_i^{t_i-1}}{2}$. In step $(*)$ we denote by l a node such that that the noise $\phi_i^{t_i-1} v_i^{t_i-1}$ was added to the system when the edge (i, l) was chosen (we do not consider $t_i = 0$ since in this case the Lemma 57 trivially holds). □

Lemma 58.

$$\mathbb{E}\left[\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right)^2 | x^t, e^t\right] = \sigma_i^2 \phi_i^{2t_i} + \sigma_j^2 \phi_j^{2t_j}. \tag{5.27}$$

Proof. Since we have $\mathbb{E}\left[\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right) | x^t, e^t\right] = 0$, and also for any random variable X: $\mathbb{E}\left[X^2\right] = \mathbb{V}(X) + \mathbb{E}[X]^2$, we only need to compute the variance:

$$\mathbb{V}\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right) = \mathbb{V}\left(\phi_i^{t_i} v_i^{t_i}\right) + \mathbb{V}\left(\phi_j^{t_j} v_j^{t_j}\right) = \left(\phi_i^{t_i}\right)^2 \mathbb{V}\left(v_i^{t_i}\right) + \left(\phi_j^{t_j}\right)^2 \mathbb{V}\left(v_j^{t_j}\right).$$

□

Having presented the above lemmas we can now proceed with the proof of Lemma 48.

Firstly, let us compute the increase of the dual function value at iteration t:

$$\begin{aligned}
D(y^{t+1}) - D(y^t) &= \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \\
&= \frac{1}{2}\left((\bar{c} - x_j^t)^2 + (\bar{c} - x_i^t)^2 - (\bar{c} - x_j^{t+1})^2 - (\bar{c} - x_i^{t+1})^2\right) \\
&= -\bar{c}\left(x_j^{t+1} + x_i^{t+1} - x_j^t - x_i^t\right) + \frac{1}{2}\left(\left(x_j^t\right)^2 + \left(x_i^t\right)^2 - \left(x_j^{t+1}\right)^2 - \left(x_i^{t+1}\right)^2\right) \\
&= -\bar{c}\left(w_j^{t_j} + w_i^{t_i}\right) + \frac{1}{2}\left(\left(x_j^t\right)^2 + \left(x_i^t\right)^2\right) \\
&\qquad -\frac{1}{4}\left(\left(x_j^t + x_i^t\right)^2 + 2\left(x_j^t + x_i^t\right)\left(w_j^{t_j} + w_i^{t_i}\right) + \left(w_j^{t_j} + w_i^{t_i}\right)^2\right) \\
&= \frac{1}{4}\left(x_j^t - x_i^t\right)^2 - \left(w_i^{t_i} + w_j^{t_j}\right)\left(\bar{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) - \frac{1}{4}\left(w_i^{t_i} + w_j^{t_j}\right)^2 \quad (5.28)
\end{aligned}$$

Our goal is to estimate an upper bound of the quantity $\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right]$. There are three terms in (5.28). Since the expectation is linear, we will evaluate the expectations of these three terms separately and merge them at the end.

Taking the expectation over the choice of edge and inserted noise in iteration t we obtain

$$\mathbb{E}\left[\frac{1}{4}\left(x_i^t - x_j^t\right)^2 | x^t\right] = \frac{1}{4m}\sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2. \quad (5.29)$$

Thus we have

$$\begin{aligned}
&\mathbb{E}\left[D(y^{t+1}) - D(y^t) - \frac{1}{4}\left(x_i^t - x_j^t\right)^2 \mid x^t\right] \\
&\overset{(5.29)}{=} \mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid x^t\right] - \frac{1}{4m}\sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2 \\
&= \mathbb{E}\left[D(y^*) - D(y^t) \mid x^t\right] + \mathbb{E}\left[D(y^{t+1}) - D(y^*) \mid x^t\right] - \frac{1}{4m}\sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2 \\
&\overset{(5.8)}{=} \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \mid x^t\right] - \frac{1}{4m}\sum_{e=(i,j)\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2 \\
&\overset{(5.23)}{\leq} \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \mid x^t\right] - \frac{\alpha(\mathcal{G})}{4m}\|\bar{c}\mathbf{1} - x^t\|^2 \\
&= \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \mid x^t\right] \\
&\overset{(5.8)}{=} \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\left(D(y^*) - D(y^t)\right) - \mathbb{E}\left[D(y^*) - D(y^{t+1}) \mid y^t\right].
\end{aligned}$$

Taking the full expectation of the above and using tower property, we get

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t) - \frac{1}{4}\left(x_i^t - x_j^t\right)^2\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*) - D(y^t)\right] - \mathbb{E}\left[D(y^*) - D(y^{t+1})\right]. \quad (5.30)$$

Now we are going to take the expectation of the second term of (5.28). We will use the

"tower rule" of expectations in the form

$$\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[X \mid Y, Z\right] \mid Y\right]\right] = \mathbb{E}\left[X\right] \tag{5.31}$$

where X, Y, Z are random variables. In particular, we get

$$\begin{aligned}
&\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)\right] \\
&\qquad = \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \mid e^t, x^t\right] \mid x^t\right]\right].
\end{aligned} \tag{5.32}$$

In equation (5.32), e^t denotes the edge selected at iteration t.

Let us first calculate the inner most expectation of the right hand side of (5.32):

$$\begin{aligned}
&\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \mid e^t, x^t\right] \\
&\overset{(5.17)}{=} \mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1} - \phi_i^{t_i} v_i^{t_i} - \phi_j^{t_j} v_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \mid e^t, x^t\right] \\
&= \mathbb{E}\left[\overbrace{\left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)}^{\text{constant}} \overbrace{\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)}^{\text{constant}} \mid e^t, x^t\right] \\
&\qquad + \mathbb{E}\left[\left(-\phi_i^{t_i} v_i^{t_i} - \phi_j^{t_j} v_j^{t_j}\right) \overbrace{\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)}^{\text{constant}} \mid e^t, x^t\right] \\
&= \left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \\
&\qquad + \mathbb{E}\left[\left(-\phi_i^{t_i} v_i^{t_i} - \phi_j^{t_j} v_j^{t_j}\right) \mid e, x^t\right]\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \to 0 \\
&\overset{L.56}{=} \left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \\
&= \left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)\overline{c} + \frac{1}{2}\left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)\left(x_j^t + x_i^t\right).
\end{aligned}$$

Now we take the expectation of the last expression above with respect to the choice of an

edge at t-th iteration. We obtain

$$
\begin{aligned}
&\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)\overline{c}+\frac{1}{2}\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t+x_i^t\right)|x^t\right]\\
&=\frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t+x_i^t\right)|x^t\right]\\
&\quad+\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)\overline{c}|x^t\right]\to 0\\
&\overset{L.56}{=}\frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t+x_i^t\right)|x^t\right]\\
&=\frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t+x_i^t\right)\\
&=\frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_i^t+\phi_j^{t_j-1}v_j^{t_j-1}x_j^t\right)\\
&\quad+\frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\\
&=\frac{1}{2m}\sum_{i=1}^{n}d_i\phi_i^{t_i-1}v_i^{t_i-1}x_i^t+\frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right),
\end{aligned}
$$

where in the last step we change the summation order.

Taking the expectation with respect to the algorithm we obtain

$$
\begin{aligned}
&\mathbb{E}\left[-\left(w_i^{t_i}+w_j^{t_j}\right)\left(\overline{c}+\frac{1}{2}\left(x_j^t+x_i^t\right)\right)\right]\\
&\overset{(5.31)}{=}\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[-\left(w_i^{t_i}+w_j^{t_j}\right)\left(\overline{c}+\frac{1}{2}\left(x_j^t+x_i^t\right)\right)|x^t,e\right]|x^t\right]\right]\\
&=\mathbb{E}\left[\frac{1}{2m}\sum_{i=1}^{n}d_i\phi_i^{t_i-1}v_i^{t_i-1}x_i^t+\frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right]\\
&=\frac{1}{2m}\sum_{i=1}^{n}d_i\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_i^t\right]+\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right]\\
&\overset{L.57}{=}\frac{1}{4m}\sum_{i=1}^{n}d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]+\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right]. \quad (5.33)
\end{aligned}
$$

Taking an expectation of the third term of (5.28) with respect to lastly added noise, the

expression $\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2|x^t,e^t\right]$ is equal to

$$\begin{aligned}
\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2|x^t,e^t\right] &\overset{(5.17)}{=} \mathbb{E}\left[\left(\phi_i^{t_i}v_i^{t_i}+\phi_j^{t_j}v_j^{t_j}-\phi_i^{t_i-1}v_i^{t_i-1}-\phi_j^{t_j-1}v_j^{t_j-1}\right)^2|x^t,e^t\right] \\
&= \mathbb{E}\left[\left(\phi_i^{t_i}v_i^{t_i}+\phi_j^{t_j}v_j^{t_j}\right)^2|x^t,e^t\right] \\
&\quad -2\mathbb{E}\left[\overbrace{\left(\phi_i^{t_i}v_i^{t_i}+\phi_j^{t_j}v_j^{t_j}\right)}^{\mathbb{E}\left[\phi_i^{t_i}v_i^{t_i}+\phi_j^{t_j}v_j^{t_j}\right]=0}\overbrace{\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)}^{\text{constant}}|x^t,e^t\right] \\
&\quad +\overbrace{\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)^2|x^t,e^t\right]}^{\text{constant}} \\
&\overset{(5.27)}{=} \sigma_i^2\phi_i^{2t_i}+\sigma_j^2\phi_j^{2t_j}+\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)^2.
\end{aligned}$$

Taking the expectation over e^t we obtain:

$$\begin{aligned}
\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2|x^t\right] &\overset{(5.31)}{=} \mathbb{E}\left[\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2|e^t,x^t\right]|x^t\right] \\
&= \mathbb{E}\left[\sigma_i^2\phi_i^{2t_i}+\sigma_j^2\phi_j^{2t_j}+\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)^2|x^t\right] \\
&= \frac{1}{m}\sum_{e\in\mathcal{E}}\left(\sigma_i^2\phi_i^{2t_i}+\sigma_j^2\phi_j^{2t_j}\right) \\
&\quad +\frac{1}{m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)^2 \\
&= \frac{1}{m}\sum_{i=1}^{n}d_i\sigma_i^2\phi_i^{2t_i}+\frac{1}{m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)^2.
\end{aligned}$$

where in the last step we change the summation order.

Finally, taking the expectation with respect to the algorithm we get

$$
\begin{aligned}
&\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2\right]\\
&\overset{(5.31)}{=}\mathbb{E}\left[\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2|x^t\right]\right]\\
&=\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)^2\right]\\
&=\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]\\
&\quad+\frac{1}{m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2+\left(\phi_j^{t_j-1}v_j^{t_j-1}\right)^2+2\phi_i^{t_i-1}v_i^{t_i-1}\phi_j^{t_j-1}v_j^{t_j-1}\right]\\
&\overset{(*)}{=}\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]+\frac{2}{m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\phi_j^{t_j-1}v_j^{t_j-1}\right]\\
&\overset{L.56}{=}\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]\\
&\quad+\frac{2}{m}\sum_{e\in\mathcal{E}}\cancelto{0}{\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\right]}\cancelto{0}{\mathbb{E}\left[\phi_j^{t_j-1}v_j^{t_j-1}\right]}\\
&=\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right],
\end{aligned}
\tag{5.34}
$$

where in step (*) we change the summation order.

Combining (5.28) with (5.30), (5.33) and (5.34) we obtain

$$
\begin{aligned}
\mathbb{E}\left[D(y^*)-D(y^{t+1})\right] &\leq \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]-\frac{1}{4m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]\\
&\quad-\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right]\\
&\quad+\frac{1}{4m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{4m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]\\
&= \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{1}{4m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]\\
&\quad-\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right],
\end{aligned}
$$

which concludes the proof.

5.6.9 Proof of Theorem 49

Let us present two lemmas that we use in the proof of Theorem 49.

Lemma 59. *After t iterations of algorithm 17 we have*

$$
\mathbb{E}\left[\phi_i^{2t_i}\right]=\left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t. \tag{5.35}
$$

Proof.

$$\begin{aligned}\mathbb{E}\left[\phi_i^{2t_i}\right] &= \sum_{j=0}^{t}\mathbb{P}(t_i=j)\phi_i^{2j} = \sum_{j=0}^{t}\binom{t}{j}\left(\frac{m-d_i}{m}\right)^{t-j}\left(\frac{d_i}{m}\phi_i^2\right)^j \\ &= \left(\frac{m-d_i}{m}+\frac{d_i}{m}\phi_i^2\right)^t = \left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t.\end{aligned}$$

□

Lemma 60. *Random variables $\phi_i^{t_i-1}v_i^{t_i-1}$ and x_j^t are nonegatively correlated, i.e.*

$$\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t\right] \geq 0. \tag{5.36}$$

Proof. Denote $R_{i,j}$ to be a random variable equal to 1 if the noise $w_i^{t_i}$ was added to the system when edge (i,j) was chosen and equal to 0 otherwise. We can rewrite the expectation in the following way:

$$\begin{aligned}\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t\right] &= \overbrace{\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t \mid R_{i,j}=1\right]}^{\geq 0}\mathbb{P}(R_{i,j}=1) \\ &\quad + \overbrace{\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t \mid R_{i,j}=0\right]}^{0}\mathbb{P}(R_{i,j}=0) \quad \geq \quad 0.\end{aligned}$$

The inequality $\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t \mid R_{i,j}=1\right] \geq 0$ holds due to the fact that $\phi_i^{t_i-1}v_i^{t_i-1}$ was added to x_j with the positive sign. □

Combining (5.18) with the results of Lemmas 59 and 60 we obtain:

$$\begin{aligned}\mathbb{E}\left[D(y^*)-D(y^{t+1})\right] &\overset{(5.18)}{\leq} \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{1}{4m}\sum_{i=1}^{n}d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right] \\ &\quad -\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_i^{t_j-1}v_j^{t_j-1}x_i^t\right)\right] \\ &\overset{(5.36)}{\leq} \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{1}{4m}\sum_{i=1}^{n}d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right] \\ &\overset{(5.35)}{=} \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right] \\ &\quad +\frac{1}{4m}\sum_{i=1}^{n}d_i\sigma_i^2\left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t \\ &= \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{\sum\left(d_i\sigma_i^2\right)}{4m}\psi^t.\end{aligned}$$

The recursion above gives us inductively the following

$$\mathbb{E}\left[D(y^*)-D(y^k)\right] \leq \rho^k\left(D(y^*)-D(y^0)\right)+\frac{\sum\left(d_i\sigma_i^2\right)}{4m}\sum_{t=1}^{k}\rho^{k-t}\psi^t,$$

which concludes the proof of the theorem.

5.6.10 Proof of Corollary 50

Note that we have

$$\begin{aligned}\psi^t &= \frac{1}{\sum_{i=1}^n d_i\sigma_i^2}\sum_{i=1}^n d_i\sigma_i^2\left(1-\frac{d_i}{m}\left(1-\left(1-\frac{\gamma}{d_i}\right)\right)\right)^t \\ &= \frac{1}{\sum_{i=1}^n d_i\sigma_i^2}\sum_{i=1}^n d_i\sigma_i^2\left(1-\frac{\gamma}{m}\right)^t \quad = \quad \left(1-\frac{\gamma}{m}\right)^t.\end{aligned}$$

In view of Theorem 49, this gives us the following:

$$\begin{aligned}\mathbb{E}\left[D(y^*)-D(y^k)\right] &\leq \left(D(y^*)-D(y^0)\right)\rho^k + \frac{\sum\left(d_i\sigma_i^2\right)}{4m}\sum_{t=1}^k \rho^{k-t}\psi^t \\ &\leq \left(D(y^*)-D(y^0)\right)\rho^k + \frac{\sum\left(d_i\sigma_i^2\right)}{4m}\sum_{t=1}^k \rho^{k-t}\left(1-\frac{\gamma}{m}\right)^t \\ &\leq \left(D(y^*)-D(y^0)\right)\rho^k + \frac{\sum\left(d_i\sigma_i^2\right)}{4m}k\max\left(\rho,1-\frac{\gamma}{m}\right)^k \\ &\leq \left(D(y^*)-D(y^0)+\frac{\sum\left(d_i\sigma_i^2\right)}{4m}k\right)\max\left(\rho,1-\frac{\gamma}{m}\right)^k.\end{aligned}$$

Chapter 6

Conclusion and Future Work

6.1 Conclusions

In this thesis we studied the design and analysis of novel efficient randomized iterative methods for solving large scale linear systems, stochastic quadratic optimization problems, the best approximation problem and quadratic optimization problems. Using these methods we also proposed and analyzed efficient gossip protocols for solving the average consensus problem on large scale networks.

In Chapter 2, we studied the convergence of several stochastic optimization algorithms enriched with *heavy ball momentum* for solving stochastic optimization problems of special structure. We proved global, non-asymptotic linear convergence rates of all of these methods as well as accelerated linear rate for the case of the norm of expected iterates. We also introduced a new momentum strategy called *stochastic momentum* which is beneficial in the case of sparse data, and proved linear convergence in this setting. We corroborated our theoretical results with extensive experimental testing.

In Chapter 3, we proposed and analyzed *inexact* variants of several stochastic algorithms for solving quadratic optimization problems and linear systems. We provided linear convergence rate under several assumptions on the inexactness error. The proposed methods require more iterations than their exact variants to achieve the same accuracy. However, as we show through our numerical evaluations, the inexact algorithms require significantly less time to converge.

In Chapter 4, we presented a general framework for the analysis and design of *randomized gossip* algorithms for solving the average consensus problem. Using tools from numerical linear algebra and the area of randomized projection methods for solving linear systems, we proposed novel serial, block and accelerated gossip protocols for solving the average consensus and weighted average consensus problems.

In Chapter 5, we addressed the average consensus problem via novel asynchronous *privacy preserving* randomized gossip algorithms. In particular, we proposed three different algorithmic tools for the protection of the initial private values of the nodes. The first two proposed algorithms "Private Gossip via Binary Oracle" and "Private Gossip via ϵ-Gap Oracle" are based on the same idea of weakening the oracle used in the gossip update rule. Instead of sharing their exact values, in these two protocols the chosen pair of nodes of each gossip step provide only categorical (or even binary) information to each other. In the third protocol, "Private Gossip via Controlled Noise Insertion", we systematically inject and withdraw noise throughout the iterations, so as to ensure convergence to the average consensus value, and at the same time protect the private information of the nodes. For all proposed protocols, we provide explicit convergence rates and evaluate practical convergence on common simulated network topologies.

6.2 Future Work

Perhaps the most exciting direction for future work is to extend the analysis of the proposed randomized iterative methods to more general settings. In particular, the more natural exten-

sion of our results is the analysis of heavy ball momentum variants and inexact variants of the proposed methods (SGD, SN, SPP, SPM and SDSA) in the case of general convex or strongly convex functions.

From numerical linear algebra viewpoint, we believe that the proposed randomized iterative methods of Chapters 2 and 3 could have great potential to make a practical difference to iterative solvers for large-scale linear systems. In this aspect, a future effort needs to be devoted to the practical development and implementations of the algorithms. For example, one promising direction is to use new sophisticated sketching matrices $\mathbf{S}$, such as the Walsh-Hadamard matrix [155, 119] in the update rules of the proposed methods.

In this thesis we focused on algorithms with a fixed constant step-size. An interesting extension will be to study the effect of decreasing or adaptive choice for the relaxation parameter. This might provide novel insights, even in the case of quadratic functions and (not necessarily consistent) linear systems. As we have mentioned in several parts of the thesis, the obtained results hold under the exactness condition, which as we explained, is very weak, allowing for virtually arbitrary distributions $\mathcal{D}$ from which the random matrices are drawn. A different future direction will be the design of optimized distributions in order to improve further the convergence rates and the overall complexity of the proposed algorithms.

In addition, we believe that the gossip protocols proposed in Chapters 4 and 5 would be particularly useful in the development of efficient decentralized protocols.

Using the novel framework presented in this thesis, many popular projection methods can be interpreted as gossip algorithms when used to solve linear systems encoding the underlying network. This can lead to the development of novel distributed protocols for average consensus.

Our work on gossip algorithms is amenable to further extensions. For instance, the proposed novel gossip protocols (block, accelerated, privacy-preserving) can be extended to the more general setting of multi-agent consensus optimization, where the goal is to minimize the average of convex or non-convex functions $(1/n)\sum_{i=1}^{n} f_i(x)$ in a decentralized way. Such protocols will be particularly useful in settings where the data describing a given optimization problem is so big that it becomes impossible to store it on a single machine. These situations often arise in modern machine learning and deep learning applications.

Bibliography

[1] Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205. ACM, 2017.

[2] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *International Conference on Machine Learning*, pages 1110–1119, 2016.

[3] S.MR Arnold, P.A. Manzagol, R. Babanezhad, I. Mitliagkas, and N.L. Roux. Reducing the variance in online optimization by transporting past gradients. *arXiv preprint arXiv:1906.03532*, 2019.

[4] M. Assran, N. Loizou, N. Ballas, and M. Rabbat. Stochastic gradient push for distributed deep learning. *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[5] M. Assran and M. Rabbat. Asynchronous subgradient-push. *arXiv preprint arXiv:1803.08950*, 2018.

[6] N. S. Aybat and M. Gürbüzbalaban. Decentralized computation of effective resistances and acceleration of consensus algorithms. In *Signal and Information Processing (GlobalSIP), 2017 IEEE Global Conference on*, pages 538–542. IEEE, 2017.

[7] T.C. Aysal, M.E. Yildiz, A.D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Trans. Signal Process.*, 57(7):2748–2761, 2009.

[8] F. Bénézit, A.G. Dimakis, P. Thiran, and M. Vetterli. Order-optimal consensus through randomized path averaging. *IEEE Trans. Inf. Theory*, 56(10):5150–5167, 2010.

[9] A.S. Berahas, R. Bollapragada, and J. Nocedal. An investigation of Newton-sketch and subsampled Newton methods. *arXiv preprint arXiv:1705.06211*, 2017.

[10] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[11] D.P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: A survey. *Optimization for Machine Learning*, 2010(1-38):3, 2011.

[12] P. Birken. Termination criteria for inexact fixed-point schemes. *Numerical Linear Algebra with Applications*, 22(4):702–716, 2015.

[13] D. Blatt, A.O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

[14] R. Bollapragada, R. Byrd, and J. Nocedal. Exact and inexact subsampled Newton methods for optimization. *arXiv preprint arXiv:1609.08502*, 2016.

[15] J. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.

[16] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 14(SI):2508–2530, 2006.

[17] C.L. Byrne. *Applied iterative methods*. AK Peters Wellesley, 2008.

[18] B. Can, M. Gurbuzbalaban, and L. Zhu. Accelerated linear convergence of stochastic momentum methods in wasserstein distances. In *International Conference on Machine Learning*, pages 891–901, 2019.

[19] M. Cao, D.A. Spielman, and E.M. Yeh. Accelerated gossip algorithms for distributed computation. In *Proc. of the 44th Annual Allerton Conference on Communication, Control, and Computation*, pages 952–959, 2006.

[20] A. Cassioli, D. Di Lorenzo, and M. Sciandrone. On the convergence of inexact block coordinate descent methods for constrained optimization. *European Journal of Operational Research*, 231(2):274–281, 2013.

[21] A. Cauchy. Méthode générale pour la résolution des systemes déquations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[22] A. Chambolle, M.J. Ehrhardt, P. Richtárik, and C.B. Schonlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

[23] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[24] T. Charalambous, M.G. Rabbat, M. Johansson, and C.N. Hadjicostis. Distributed finite-time computation of digraph parameters: Left eigenvector, out-degree and spectrum. *IEEE Trans. Control of Network Systems*, 3(2):137–148, June 2016.

[25] I. Colin, A. Bellet, J. Salmon, and S. Clémençon. Gossip dual averaging for decentralized optimization of pairwise functions. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48*, pages 1388–1396. JMLR. org, 2016.

[26] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[27] D. Csiba and P. Richtárik. Global convergence of arbitrary-block gradient methods for generalized Polyak-Lojasiewicz functions. *arXiv preprint arXiv:1709.03014*, 2017.

[28] Dominik Csiba, Zheng Qu, and Peter Richtárik. Stochastic Dual Coordinate Ascent with Adaptive Probabilities. *International Conference on Machine Learning, 2015*, 2015.

[29] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *J. Parallel Distrib. Comput.*, 7(2):279–301, 1989.

[30] L. Dai, M. Soltanalian, and K. Pelckmans. On the randomized Kaczmarz algorithm. *IEEE Signal Processing Letters*, 21(3):330–333, 2014.

[31] Nair Maria Maia De Abreu. Old and new results on algebraic connectivity of graphs. *Linear Algebra and its Applications*, 423(1):53–73, 2007.

[32] J. A. De Loera, J. Haddock, and D. Needell. A sampling Kaczmarz–Motzkin algorithm for linear feasibility. *SIAM Journal on Scientific Computing*, 39(5):S66–S87, 2017.

[33] A. Defazio. A simple practical accelerated method for finite sums. In *Advances in Neural Information Processing Systems*, pages 676–684, 2016.

[34] A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[35] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[36] R.S. Dembo, S.C. Eisenstat, and T. Steihaug. Inexact Newton methods. *SIAM Journal on Numerical Analysis*, 19(2):400–408, 1982.

[37] A. Desai, M. Ghashami, and J.M. Phillips. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1678–1690, 2016.

[38] C.A. Desoer and B.H. Whalen. A note on pseudoinverses. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):442–447, 1963.

[39] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

[40] A.M. Devraj, A. Bušić, and S. Meyn. Zap meets momentum: Stochastic approximation algorithms with optimal convergence rate. *arXiv preprint arXiv:1809.06277*, 2018.

[41] A.M. Devraj, A. Bušic, and S.P. Meyn. Optimal matrix momentum stochastic approximation and applications to q-learning. *arXiv preprint arXiv:1809.06277*, 2018.

[42] A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[43] A.G. Dimakis, A.D. Sarwate, and M.J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Trans. Signal Process.*, 56(3):1205–1216, 2008.

[44] P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlós. Faster least squares approximation. *Numerische mathematik*, 117(2):219–249, 2011.

[45] K. Du. Tight upper bounds for the convergence of the randomized extended Kaczmarz and Gauss–Seidel algorithms. *Numerical Linear Algebra with Applications*, 26(3):e2233, 2019.

[46] P. Dvurechensky, A. Gasnikov, and A. Tiurin. Randomized similar triangles method: A unifying framework for accelerated randomized optimization methods (coordinate descent, directional search, derivative-free method). *arXiv preprint arXiv:1707.08486*, 2017.

[47] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[48] S. Elaydi. *An Introduction to Difference Equations*. Springer Science & Business Media, 2005.

[49] Y.C. Eldar and D. Needell. Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma. *Numerical Algorithms*, 58(2):163–177, 2011.

[50] O. Fercoq and P. Richtárik. Smooth minimization of nonsmooth functions by parallel coordinate descent. *arXiv:1309.5885*, 2013.

[51] O. Fercoq and P. Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[52] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[53] J.P. Fillmore and M.L. Marx. Linear recursive sequences. *SIAM Review*, 10(3):342–353, 1968.

[54] K. Fountoulakis and R. Tappenden. A flexible coordinate descent method. *Computational Optimization and Applications*, 70(2):351–394, 2018.

[55] M. Franceschelli, A. Giua, and C. Seatzu. Distributed averaging in sensor networks based on broadcast gossip algorithms. *IEEE Sensors Journal*, 11(3):808–817, 2011.

[56] N.M. Freris and A. Zouzias. Fast distributed smoothing of relative measurements. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1411–1416. IEEE, 2012.

[57] Valerio Freschi, Emanuele Lattanzi, and Alessandro Bogliolo. Accelerating distributed averaging in sensor networks: Randomized gossip over virtual coordinates. In *Sensors Applications Symposium (SAS), 2016 IEEE*, pages 1–6. IEEE, 2016.

[58] Valerio Freschi, Emanuele Lattanzi, and Alessandro Bogliolo. Fast distributed consensus through path averaging on random walks. *Wireless Pers Commun*, doi:10.1007/s11277-017-4451-5:1–15, 2017.

[59] M.P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

[60] S. Gadat, F. Panloup, and S. Saadane. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461–529, 2018.

[61] S. Geman. A limit theorem for the norm of random matrices. *The Annals of Probability*, pages 252–261, 1980.

[62] E. Ghadimi, H.R. Feyzmahdavian, and M. Johansson. Global convergence of the heavy-ball method for convex optimization. In *Control Conference (ECC), 2015 European*, pages 310–315. IEEE, 2015.

[63] E. Ghadimi, I. Shames, and M. Johansson. Multi-step gradient methods for networked optimization. *IEEE Transactions on Signal Processing*, 61(21):5417–5429, 2013.

[64] E. Ghadimi, A. Teixeira, M.G. Rabbat, and M. Johansson. The admm algorithm for distributed averaging: Convergence rates and optimal parameter selection. In *2014 48th Asilomar Conference on Signals, Systems and Computers*, pages 783–787. IEEE, 2014.

[65] S. Ghadimi and G. Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

[66] M. Ghashami, E. Liberty, J. M. Phillips, and D. P. Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

[67] G.H. Golub and C.F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[68] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtarik. SGD: General analysis and improved rates. *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

[69] R. M. Gower and P. Richtárik. Randomized quasi-newton updates are linearly convergent matrix inversion algorithms. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1380–1409, 2017.

[70] R. M. Gower, P. Richtárik, and F. Bach. Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching. *arXiv preprint arXiv:1805.02632*, 2018.

[71] R.M. Gower, D. Goldfarb, and P. Richtárik. Stochastic block BFGS: squeezing more curvature out of data. In *International Conference on Machine Learning*, pages 1869–1878, 2016.

[72] R.M. Gower, F. Hanzely, P. Richtárik, and S. U. Stich. Accelerated stochastic matrix inversion: general theory and speeding up BFGS rules for faster second-order optimization. In *Advances in Neural Information Processing Systems*, pages 1619–1629, 2018.

[73] R.M. Gower and P. Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.

[74] R.M. Gower and P. Richtárik. Stochastic dual ascent for solving linear systems. *arXiv preprint arXiv:1512.06890*, 2015.

[75] R.M. Gower and P. Richtárik. Linearly convergent randomized iterative methods for computing the pseudoinverse. *arXiv preprint arXiv:1612.06255*, 2016.

[76] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[77] M. Gurbuzbalaban, A. Ozdaglar, and P.A. Parrilo. On the convergence rate of incremental aggregated gradient algorithms. *SIAM Journal on Optimization*, 27(2):1035–1048, 2017.

[78] J. Haddock and D. Needell. On motzkins method for inconsistent linear systems. *BIT Numerical Mathematics*, pages 1–15, 2018.

[79] F. Hanzely, J. Konečný, N. Loizou, P. Richtárik, and D. Grishchenko. Privacy preserving randomized gossip algorithms. *arXiv preprint arXiv:1706.07636*, 2017.

[80] F. Hanzely, J. Konečnỳ, N. Loizou, P. Richtárik, and D. Grishchenko. A privacy preserving randomized gossip algorithm via controlled noise insertion. *NeurIPS Privacy Preserving Machine Learning Workshop*, 2018.

[81] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: stability of stochastic gradient descent. In *33rd International Conference on Machine Learning*, 2016.

[82] F. He, A. S. Morse, J. Liu, and S. Mou. Periodic gossiping. *IFAC Proceedings Volumes*, 44(1):8718–8723, 2011.

[83] H. Hendrikx, L. Massoulié, and F. Bach. Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. *arXiv preprint arXiv:1810.02660*, 2018.

[84] A. G. Hernandes, M. L. Proença Jr, and T. Abrão. Improved weighted average consensus in distributed cooperative spectrum sensing networks. *Transactions on Emerging Telecommunications Technologies*, 29(3):e3259, 2018.

[85] B. Hu, P. Seiler, and L. Lessard. Analysis of approximate stochastic gradient using quadratic constraints and sequential semidefinite programs. *arXiv preprint arXiv:1711.00987*, 2017.

[86] Zhenqi Huang, Sayan Mitra, and Geir Dullerud. Differentially private iterative synchronous consensus. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, pages 81–90. ACM, 2012.

[87] A. Jalilzadeh, U.V. Shanbhag, J.H. Blanchet, and P.W. Glynn. Optimal smoothed variable sample-size accelerated proximal methods for structured nonsmooth stochastic convex programs. *arXiv preprint arXiv:1803.00718*, 2018.

[88] A. Jofré and P. Thompson. On variance reduction for stochastic smooth convex optimization with multiplicative noise. *Mathematical Programming*, pages 1–40, 2017.

[89] R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[90] Ye Yu Jun and Michael Rabbat. Performance comparison of randomized gossip, broadcast gossip and collection tree protocol for distributed averaging. In *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 93–96. IEEE, 2013.

[91] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de lAcademie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.

[92] H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

[93] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 482–491. IEEE, 2003.

[94] E. Kokiopoulou and P. Frossard. Polynomial filtering for fast convergence in distributed consensus. *IEEE Transactions on Signal Processing*, 57(1):342–354, 2009.

[95] A. Koloskova, S. U. Stich, and M. Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. *arXiv preprint arXiv:1902.00340*, 2019.

[96] J. Konečný, J. Liu, P. Richtárik, and M. Takáč. Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255, 2016.

[97] J. Konečný and P. Richtárik. Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3(9):1–14, 2017.

[98] D. Kovalev, S. Horváth, and P. Richtárik. Dont jump through hoops and remove those loops: SVRG and Katyusha are better without the outer loop. *arXiv preprint arXiv:1901.08689*, 2019.

[99] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[100] A. Kulakova, M. Danilova, and B. Polyak. Non-monotone behavior of the heavy ball method. *arXiv preprint arXiv:1811.00658*, 2018.

[101] Jerome Le Ny and George J Pappas. Differentially private filtering. *IEEE Transactions on Automatic Control*, 59(2):341–354, 2014.

[102] Y.T. Lee and A. Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 147–156. IEEE, 2013.

[103] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Optim.*, 26(1):57–95, 2016.

[104] D. Leventhal and A.S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.

[105] X. Lian, C. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pages 3049–3058, 2018.

[106] J. Liu, B.D.O. Anderson, M. Cao, and A.S. Morse. Analysis of accelerated gossip algorithms. *Automatica*, 49(4):873–883, 2013.

[107] J. Liu and S. Wright. An accelerated randomized Kaczmarz algorithm. *Mathematics of Computation*, 85(297):153–178, 2016.

[108] Ji Liu, Shaoshuai Mou, A Stephen Morse, Brian DO Anderson, and Changbin Yu. Deterministic gossiping. *Proceedings of the IEEE*, 99(9):1505–1524, 2011.

[109] Y. Liu, J. Wu, I. Manchester, and G. Shi. Privacy-preserving gossip algorithms. *arXiv preprint arXiv:1808.00120*, 2018.

[110] Yang Liu, Bo Li, Brian Anderson, and Guodong Shi. Clique gossiping. *arXiv preprint arXiv:1706.02540*, 2017.

[111] N. Loizou. Distributionally robust games with risk-averse players. *Proceedings of the 5th International Conference on Operations Research and Enterprise Systems, 2016*, pages 186–196, 2016.

[112] N. Loizou, M. Rabbat, and P. Richtárik. Provably accelerated randomized gossip algorithms. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7505–7509. IEEE, 2019.

[113] N. Loizou and P. Richtárik. A new perspective on randomized gossip algorithms. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 440–444. IEEE, 2016.

[114] N. Loizou and P. Richtárik. Linearly convergent stochastic heavy ball method for minimizing generalization error. *NIPS-Workshop on Optimization for Machine Learning*, 2017.

[115] N. Loizou and P. Richtárik. Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods. *arXiv preprint arXiv:1712.09677*, 2017.

[116] N. Loizou and P. Richtárik. Accelerated gossip via stochastic heavy ball method. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 927–934. IEEE, 2018.

[117] N. Loizou and P. Richtárik. Convergence analysis of inexact randomized iterative methods. *arXiv preprint arXiv:1903.07971*, 2019.

[118] N. Loizou and P. Richtárik. Revisiting randomized gossip algorithms: General framework, convergence rates and novel block and accelerated protocols. *arXiv preprint arXiv:1905.08645*, 2019.

[119] Y. Lu, P. Dhillon, D. P. Foster, and L. Ungar. Faster ridge regression via the subsampled randomized hadamard transform. In *Advances in neural information processing systems*, pages 369–377, 2013.

[120] A. Ma, D. Needell, and A. Ramdas. Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1590–1604, 2015.

[121] J. Ma and D. Yarats. Quasi-hyperbolic momentum and adam for deep learning. *arXiv preprint arXiv:1810.06801*, 2018.

[122] M. W Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

[123] Nicolaos E Manitara and Christoforos N Hadjicostis. Privacy-preserving asymptotic average consensus. In *Control Conference (ECC), 2013 European*, pages 760–765. IEEE, 2013.

[124] Yilin Mo and Richard M Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765, 2017.

[125] E.H. Moore. On the reciprocal of the general algebraic matrix. *Bull. Am. Math. Soc.*, 26:394–395, 1920.

[126] M. S. Morshed, M.S. Islam, et al. Accelerated sampling Kaczmarz Motzkin algorithm for linear feasibility problem. *arXiv preprint arXiv:1902.03502*, 2019.

[127] S. Mou, C. Yu, B.D.O Anderson, and A. S. Morse. Deterministic gossiping with a periodic protocol. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5787–5791. IEEE, 2010.

[128] I. Necoara and V. Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *IEEE Transactions on Automatic Control*, 59(5):1232–1243, 2014.

[129] I. Necoara, Y. Nesterov, and F. Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, pages 1–39, 2018.

[130] I. Necoara, P. Richtárik, and A. Patrascu. Randomized projection methods for convex feasibility problems: conditioning and convergence rates. *arXiv preprint arXiv:1801.04873*, 2018.

[131] A. Nedić, A. Olshevsky, and M. G. Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

[132] D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, 50(2):395–403, 2010.

[133] D. Needell, N. Srebro, and R. Ward. Stochastic gradient descent and the randomized Kaczmarz algorithm. *Mathematical Programming, Series A*, 155(1):549–573, 2016.

[134] D. Needell and J.A. Tropp. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra and its Applications*, 441:199–221, 2014.

[135] D. Needell, R. Zhao, and A. Zouzias. Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra and its Applications*, 484:322–343, 2015.

[136] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[137] A. Nemirovskii and D.B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley Interscience, 1983.

[138] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[139] Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[140] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.

[141] L. Nguyen, P. H. Nguyen, M. van Dijk, P. Richtárik, K. Scheinberg, and M. Takáč. SGD and hogwild! Convergence without the bounded gradients assumption. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3750–3758. PMLR, 2018.

[142] Erfan Nozari, Pavankumar Tallapragada, and Jorge Cortés. Differentially private average consensus: obstructions, trade-offs, and optimal algorithm design. *Automatica*, 81:221–231, 2017.

[143] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In *International Conference on Machine Learning*, pages 1632–1641, 2015.

[144] J. Nutini, B. Sepehry, I. Laradji, M. Schmidt, H. Koepke, and A. Virani. Convergence rates for greedy Kaczmarz algorithms, and faster randomized Kaczmarz rules using the orthogonality graph. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 547–556. AUAI Press, 2016.

[145] P. Ochs, T. Brox, and T. Pock. iPiasco: Inertial proximal algorithm for strongly convex optimization. *Journal of Mathematical Imaging and Vision*, 53(2):171–181, 2015.

[146] P. Ochs, Y. Chen, T. Brox, and T. Pock. iPiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.

[147] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[148] A. Olshevsky. Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control. *arXiv preprint arXiv:1411.4186*, 2014.

[149] A. Olshevsky and J.N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.

[150] B. N. Oreshkin, M. J. Coates, and M. G. Rabbat. Optimization and analysis of distributed averaging with short node memory. *IEEE Transactions on Signal Processing*, 58(5):2850–2865, 2010.

[151] F. Pedroche Sánchez, M. Rebollo Pedruelo, C. Carrascosa Casamayor, and A. Palomares Chust. Convergence of weighted-average consensus for undirected graphs. *International Journal of Complex Systems in Science*, 4(1):13–16, 2014.

[152] M. Penrose. *Random Geometric Graphs*. Number 5. Oxford University Press, 2003.

[153] R. Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge University Press, 1955.

[154] R. Penrose. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge University Press, 1956.

[155] M. Pilanci and M. J. Wainwright. Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *The Journal of Machine Learning Research*, 17(1):1842–1879, 2016.

[156] B.T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[157] B.T. Polyak. Introduction to optimization. translations series in mathematics and engineering. *Optimization Software*, 1987.

[158] C. Popa. Least-squares solution of overdetermined inconsistent linear systems using Kaczmarz's relaxation. *International Journal of Computer Mathematics*, 55(1-2):79–89, 1995.

[159] C. Popa. Convergence rates for Kaczmarz-type algorithms. *Numerical Algorithms*, 79(1):1–17, 2018.

[160] Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.

[161] Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling ii: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

[162] Z. Qu, P. Richtárik, M. Takáč, and O. Fercoq. SDNA: Stochastic dual Newton ascent for empirical risk minimization. *International Conference on Machine Learning*, 2016.

[163] Z. Qu, P. Richtárik, and T. Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in Neural Information Processing Systems*, pages 865–873, 2015.

[164] M.G. Rabbat, R.D. Nowak, and J.A. Bucklew. Generalized consensus computation in networked systems with erasure links. In *IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pages 1088–1092. IEEE, 2005.

[165] Wei Ren, Randal W Beard, and Ella M Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems*, 27(2):71–82, 2007.

[166] P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

[167] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016.

[168] P. Richtárik and M. Takáč. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv:1706.01108*, 2017.

[169] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[170] Ohad S. and Tong Z. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on Machine Learning*, pages 71–79, 2013.

[171] S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

[172] M. Schmidt, D. Kim, and S. Sra. Projected Newton-type methods in machine learning. *Optimization for Machine Learning*, page 305, 2011.

[173] M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

[174] M. Schmidt, N.L. Roux, and F.R. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems*, pages 1458–1466, 2011.

[175] F. Schöpfer and D.A. Lorenz. Linear convergence of the randomized sparse Kaczmarz method. *Mathematical Programming*, pages 1–28, 2018.

[176] Devavrat Shah. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009.

[177] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: primal estimated subgradient solver for SVM. In *24th International Conference on Machine Learning*, pages 807–814, 2007.

[178] Sh. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

[179] W. Shi, Q. Ling, G. Wu, and W. Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[180] Anthony Man-Cho So and Z. Zhou. Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity. *Optimization Methods and Software*, 32(4):963–992, 2017.

[181] M.V. Solodov and B.F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numer. Func. Anal. Opt.*, 22(7-8):1013–1035, 2001.

[182] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2):262–278, 2009.

[183] T. Sun, D. Li, Z. Quan, H. Jiang, S. Li, and Y. Dou. Heavy-ball algorithms always escape saddle points. *arXiv preprint arXiv:1907.09697*, 2019.

[184] T. Sun, P. Yin, D. Li, C. Huang, L. Guan, and H. Jiang. Non-ergodic convergence analysis of heavy-ball algorithms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5033–5040, 2019.

[185] I. Sutskever, J. Martens, G.E. Dahl, and G.E. Hinton. On the importance of initialization and momentum in deep learning. *International Conference on Machine Learning*, 28:1139–1147, 2013.

[186] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[187] R. Tappenden, P. Richtárik, and J. Gondzio. Inexact coordinate descent: complexity and preconditioning. *Journal of Optimization Theory and Applications*, 170(1):144–176, 2016.

[188] P. Tseng. An incremental gradient (-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

[189] K. Tsianos, S. Lawlor, and M. G. Rabbat. Communication/computation tradeoffs in consensus-based distributed optimization. In *Conference on Neural Information Processing Systems*, 2012.

[190] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.

[191] John N Tsitsiklis. Problems in decentralized decision making and computation. Technical report, DTIC Document, 1984.

[192] S. Tu, S. Venkataraman, A.C. Wilson, A. Gittens, M.I. Jordan, and B. Recht. Breaking locality accelerates block Gauss-Seidel. In *International Conference on Machine Learning*, 2017.

[193] D. Ustebay, M. Coates, and M. Rabbat. Greedy gossip with eavesdropping. In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pages 759–763. IEEE, 2008.

[194] S. Vaswani, F. Bach, and M. Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. *arXiv:1810.07288*, 2018.

[195] E. Wei and A. Ozdaglar. On the o (1= k) convergence of asynchronous distributed alternating direction method of multipliers. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 551–554. IEEE, 2013.

[196] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017.

[197] D.P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

[198] S. Wright and J. Nocedal. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

[199] S.J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[200] Shaochuan Wu and Michael G Rabbat. Broadcast gossip algorithms for consensus on strongly connected digraphs. *IEEE Transactions on Signal Processing*, 61(16):3959–3971, 2013.

[201] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[202] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 63–70. IEEE, 2005.

[203] P. Xu, B. He, C. De Sa, I. Mitliagkas, and C. Re. Accelerated stochastic power iteration. In *International Conference on Artificial Intelligence and Statistics*, pages 58–67, 2018.

[204] P. Xu, F. Roosta-Khorasani, and M.W. Mahoney. Newton-type methods for non-convex optimization under inexact hessian information. *arXiv preprint arXiv:1708.07164*, 2017.

[205] P. Xu, . Yang, J, F. Roosta-Khorasani, C. Ré, and M.W. Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pages 3000–3008, 2016.

[206] T. Yang, Q. Lin, and Z. Li. Unified convergence analysis of stochastic momentum methods for convex and non-convex optimization. *arXiv preprint arXiv:1604.03257*, 2016.

[207] Zhewei Yao, Peng Xu, Farbod Roosta-Khorasani, and Michael W Mahoney. Inexact non-convex Newton-type methods. *arXiv preprint arXiv:1802.06925*, 2018.

[208] C. B. Yu, B.D.O Anderson, S. Mou, J. Liu, F. He, and A. S. Morse. Distributed averaging using periodic gossiping. *IEEE Transactions on Automatic Control*, 2017.

[209] K. Yuan, Q. Ling, and W. Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

[210] S.K. Zavriev and F.V. Kostyuk. Heavy-ball method in nonconvex optimization problems. *Computational Mathematics and Modeling*, 4(4):336–341, 1993.

[211] J. Zhang, I. Mitliagkas, and C. Ré. Yellowfin and the art of momentum tuning. *arXiv preprint arXiv:1706.03471*, 2017.

[212] W. Zhang, Y. Guo, H. Liu, Y. J. Chen, Z. Wang, and J. Mitola III. Distributed consensus-based weight design for cooperative spectrum sensing. *IEEE Transactions on Parallel and Distributed Systems*, 26(1):54–64, 2015.

[213] W. Zhang, Z. Wang, Y. Guo, H. Liu, Y. Chen, and J. Mitola III. Distributed cooperative spectrum sensing based on weighted average consensus. In *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*, pages 1–6. IEEE, 2011.

[214] K. Zhou. Direct acceleration of SAGA using sampled negative momentum. *arXiv preprint arXiv:1806.11048*, 2018.

[215] K. Zhou, F. Shang, and J. Cheng. A simple stochastic variance reduced algorithm with fast convergence rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *PMLR*, pages 5980–5989, 2018.

[216] A. Zouzias and N.M. Freris. Randomized extended Kaczmarz for solving least squares. *SIAM Journal on Matrix Analysis and Applications*, 34(2):773–793, 2013.

[217] A. Zouzias and N.M. Freris. Randomized gossip algorithms for solving Laplacian systems. In *Control Conference (ECC), 2015 European*, pages 1920–1925. IEEE, 2015.

Appendix A

Notation Glossary

A.1 Notation used in Chapters 1, 2 and 3

The Basics	
$\mathbf{A}, b$	$m \times n$ matrix and $m \times 1$ vector defining the system $\mathbf{A}x = b$
$\mathcal{L}$	$\{x \ : \ \mathbf{A}x = b\}$ (solution set of the linear system)
$\mathbf{B}$	$n \times n$ symmetric positive definite matrix
$\langle x, y \rangle_{\mathbf{B}}$	$x^\top \mathbf{B} y$ ($\mathbf{B}$-inner product)
$\|x\|_{\mathbf{B}}$	$\sqrt{\langle x, x \rangle_{\mathbf{B}}}$ ($\mathbf{B}$-norm)
$\mathbf{S}$	a random real matrix with m rows
$\mathcal{D}$	distribution from which matrix $\mathbf{S}$ is drawn ($\mathbf{S} \sim \mathcal{D}$)
$\mathbf{H}$	$\mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top$
$\mathbf{Z}$	$\mathbf{A}^\top \mathbf{H} \mathbf{A}$
$\text{Range}\,(\mathbf{M})$	range space of matrix $\mathbf{M}$
$\text{Null}\,(\mathbf{M})$	null space of matrix $\mathbf{M}$
$\mathbb{E}[\cdot]$	expectation
Projections	
$\Pi_{\mathcal{L},\mathbf{B}}(x)$	projection of x onto $\mathcal{L}$ in the $\mathbf{B}$-norm
$\mathbf{B}^{-1}\mathbf{Z}$	projection matrix, in the $\mathbf{B}$-norm, onto $\text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}\right)$
Optimization	
$\mathcal{X}$	set of minimizers of f
x^*	a point in $\mathcal{L}$
$f_{\mathbf{S}}$, $\nabla f_{\mathbf{S}}$, $\nabla^2 f_{\mathbf{S}}$	stochastic function, its gradient and Hessian
$\mathcal{L}_{\mathbf{S}}$	$\{x \ : \ \mathbf{S}^\top \mathbf{A}x = \mathbf{S}^\top b\}$ (set of minimizers of $f_{\mathbf{S}}$)
f	$\mathbb{E}[f_{\mathbf{S}}]$
∇f	gradient of f with respect to the $\mathbf{B}$-inner product
$\nabla^2 f$	$\mathbf{B}^{-1}\mathbb{E}[\mathbf{Z}]$ (Hessian of f in the $\mathbf{B}$-inner product)
Eigenvalues	
$\mathbf{W}$	$\mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}$ (psd matrix with the same spectrum as $\nabla^2 f$)
$\lambda_1, \ldots, \lambda_n$	eigenvalues of $\mathbf{W}$
$\lambda_{\max}, \lambda_{\min}^+$	largest and smallest nonzero eigenvalues of $\mathbf{W}$
Algorithms	
ω	relaxation parameter / stepsize
β	heavy ball momentum parameter
γ	stochastic heavy ball momentum parameter
ϵ^k	inexactness error
q	inexactness parameter
ρ	$1 - \omega(2-\omega)\lambda_{\min}^+$

Table A.1: Frequently used notation appeared in Chapters 1, 2 and 3

A.2 Notation used in Chapter 4

The Basics	
$\mathbf{A}, b$	$m \times n$ matrix and $m \times 1$ vector defining the system $\mathbf{A}x = b$
$\mathcal{L}$	$\{x \,:\, \mathbf{A}x = b\}$ (solution set of the linear system)
$\mathbf{B}$	$n \times n$ symmetric positive definite matrix
$\langle x, y \rangle_{\mathbf{B}}$	$x^\top \mathbf{B} y$ ($\mathbf{B}$-inner product)
$\|x\|_{\mathbf{B}}$	$\sqrt{\langle x, x \rangle_{\mathbf{B}}}$ ($\mathbf{B}$-norm)
$\mathbf{M}^\dagger$	Moore-Penrose pseudoinverse of matrix $\mathbf{M}$
$\mathbf{S}$	a random real matrix with m rows
$\mathcal{D}$	distribution from which matrix $\mathbf{S}$ is drawn ($\mathbf{S} \sim \mathcal{D}$)
$\mathbf{H}$	$\mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top$
$\mathbf{Z}$	$\mathbf{A}^\top \mathbf{H} \mathbf{A}$
Range $(\mathbf{M})$	range space of matrix $\mathbf{M}$
Null$(\mathbf{M})$	null space of matrix $\mathbf{M}$
$\mathbb{P}(\cdot)$	probability of an event
$\mathbb{E}[\cdot]$	expectation
Projections	
$\Pi_{\mathcal{L},\mathbf{B}}(x)$	projection of x onto $\mathcal{L}$ in the $\mathbf{B}$-norm
$\mathbf{B}^{-1}\mathbf{Z}$	projection matrix, in the $\mathbf{B}$-norm, onto Range $\left(\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}\right)$
Graphs	
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	an undirected graph with vertices $\mathcal{V}$ and edges $\mathcal{E}$
n	$= \|\mathcal{V}\|$ (number of vertices)
m	$= \|\mathcal{E}\|$ (number of edges)
$e = (i, j) \in \mathcal{E}$	edge of $\mathcal{G}$ connecting nodes $i, j \in \mathcal{V}$
d_i	degree of node i
$c \in \mathbb{R}^n$	$= (c_1, \ldots, c_n)$; a vector of private values stored at the nodes of $\mathcal{G}$
$\bar{c}$	$\bar{c} = \frac{\sum_i^n \mathbf{B}_{ii} c_i}{\sum_i^n \mathbf{B}_{ii}}$ (the weighted average of the private values)
$\mathbf{Q} \in \mathbb{R}^{m \times m}$	Incidence matrix of $\mathcal{G}$
$\mathbf{L} \in \mathbb{R}^{n \times n}$	$= \mathbf{Q}^\top \mathbf{Q}$ (Laplacian matrix of $\mathcal{G}$)
$\mathbf{D} \in \mathbb{R}^{n \times n}$	$= \mathbf{Diag}(d_1, d_2, \ldots, d_n)$ (Degree matrix of $\mathcal{G}$)
$\mathbf{L}^{rw} \in \mathbb{R}^{n \times n}$	$= \mathbf{D}^{-1}\mathbf{L}$ (random walk normalized Laplacian matrix of $\mathcal{G}$)
$\mathbf{L}^{sym} \in \mathbb{R}^{n \times n}$	$= \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ (symmetric normalized Laplacian matrix of $\mathcal{G}$)
$\alpha(\mathcal{G})$	$= \lambda_{\min}^+(\mathbf{L})$ (algebraic connectivity of $\mathcal{G}$)
Eigenvalues	
$\mathbf{W}$	$\mathbf{B}^{-1/2}\mathbb{E}[\mathbf{Z}]\mathbf{B}^{-1/2}$ (psd matrix)
$\lambda_1, \ldots, \lambda_n$	eigenvalues of $\mathbf{W}$
$\lambda_{\max}, \lambda_{\min}^+$	largest and smallest nonzero eigenvalues of $\mathbf{W}$
Algorithms	
ω	relaxation parameter / stepsize
β	heavy ball momentum parameter
ρ	$1 - \omega(2 - \omega)\lambda_{\min}^+$

Table A.2: Frequently used notation appeared in Chapter 4.

A.3 Notation used in Chapter 5

Graphs	
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	an undirected graph with vertices $\mathcal{V}$ and edges $\mathcal{E}$
n	$= \lvert\mathcal{V}\rvert$ (number of vertices)
m	$= \lvert\mathcal{E}\rvert$ (number of edges)
$e = (i, j) \in \mathcal{E}$	edge of $\mathcal{G}$ connecting nodes $i, j \in \mathcal{V}$
d_i	degree of node i
$c \in \mathbb{R}^n$	$= (c_1, \ldots, c_n)$; a vector of private values stored at the nodes of $\mathcal{G}$
$\bar{c}$	$\bar{c} = \frac{1}{n}\sum_i c_i$ (the average of the private values)
$\mathbf{A} \in \mathbb{R}^{m \times n}$	
$\mathbf{L} \in \mathbb{R}^{m \times m}$	$= \mathbf{A}\mathbf{A}^\top$ (Laplacian of $\mathcal{G}$)
$\alpha(\mathcal{G})$	$= \lambda^+_{\min}(\mathbf{L})$ (algebraic connectivity of $\mathcal{G}$)
$\beta(\mathcal{G})$	$= n/\alpha(\mathcal{G})$
Randomness	
$\mathbb{E}$	expectation
$\mathbb{P}$	probability
$\mathbb{V}$	variance
v_i^k	random variable from $N(0, \sigma_i^2)$
Optimization	
$P : \mathbb{R}^n \to \mathbb{R}$	primal objective function
$D : \mathbb{R}^m \to \mathbb{R}$	dual objective function (a concave quadratic)
$y \in \mathbb{R}^m$	dual variable
$y^* \in \mathbb{R}^m$	optimal dual variable
$x \in \mathbb{R}^n$	primal variable
$x^* \in \mathbb{R}^n$	$= \bar{c}\mathbf{1}$ (optimal primal variable)
$\mathbf{1}$	a vector of all ones in $\mathbb{R}^n$
Summation	
$\sum_i \sum_j$	sum through all ordered pairs of i and j
$\sum_{(i,j)}$	sum through all unordered pairs of i and j
$\sum_{(i,j)\in\mathcal{E}}$	sum through all edges of $\mathcal{G}$

Table A.3: Frequently used notation appeared in Chapter 5.