FROM COORDINATE DESCENT TO SOCIAL SAMPLING: COORDINATE SAMPLING FOR LARGE SCALE OPTIMIZATION

BY MOHSEN GHASSEMI

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Prof. Anand Sarwate

and approved by

New Brunswick, New Jersey

May, 2016

ABSTRACT OF THE THESIS

From Coordinate Descent to Social Sampling: Coordinate Sampling for Large Scale Optimization

by Mohsen Ghassemi

Thesis Director: Prof. Anand Sarwate

The unprecedented rate at which data is being created and stored calls for scalable optimization techniques that allow efficient "Big Data" analysis. In this work where there is only one computing node, that modifies the coordinate-sampling distribution for stochastic coordinate descent: we call this proportional stochastic coordinate descent (PSCD). This method treats the gradient of the function as a probability distribution to sample the coordinates, and may be useful in so-called lock-free decentralized optimization schemes. Although stochastic coordinate descent methods seem attractive due to their small per-iteration complexity, they show high variance in performance compared to full gradient descent algorithms. In order to address this issue we propose stochastic variance reduced coordinate descent that takes information from the previous gradient estimates into account. Lastly, we consider stochastic message passing algorithms that limit the communication required for decentralized and distributed convex optimization and provide convergence guarantees on the objective value. For general distributed optimization in which agents jointly minimize the sum of local objectives we propose treating the iterates as gradients and propose a stochastic coordinate-wise primal averaging algorithm for optimization.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Prof. Anand Sarwate for his support and invaluable guidance. His sound advice and patience with correcting the initial drafts has helped shape up this work.

My sincere thanks also goes to Prof. Narayan Mandayam for giving me the chance to pursue my studies at Rutgers. I am grateful to Prof. Waheed Bajwa and Prof. Vishal Patel for being on my thesis committee and taking the time to review my work.

I would like to thank my labmates and also members of the INSPIRE lab for providing moral support and encouragement. I especially thank Elham Shakeri and Haroon Raja for all the insightful discussions we had almost every day.

I am also thankful to my friends all around the world for being there for me. Carolina Cocito, Mohammad Hajimirsadeghi, Talal Ahmed, Parishad Karimi, Farzin Shamloo, Omid kardan, Saeed Soori, Abtin Mahtabani, and many others who have made my life more enjoyable and meaningful through the years of my study at Rutgers.

Finally, I cannot thank my family enough for their endless love and support.

Dedication

This Thesis is Dedicated to My Parents and My Grandmother

Table of Contents

Chapter 1
Introduction

In the recent years, with emergence of many problems dealing with large scale data sets or very high dimensional data, large scale convex optimization has received a great amount of attention. Large scale machine learning, which has applications in bioinformatics, computer vision, text processing, product recommendation systems, and many other areas where "Big Data" is available, uses convex optimization as a powerful tool for empirical risk minimization. Applications of large scale convex optimization, however, are not limited to machine learning problems. For instance, resource allocation in large wireless networks can be modeled as a convex optimization problem.

In general, such problems can be formulated as

$$\min_{w\in\mathcal{R}^d} \; f(w) \triangleq \frac{1}{n}\sum_{i=1}^{n} f^i(w), \tag{1.1}$$

where $\{f^i(w)\}$ are convex functions and n or d (or both) are very large. Traditional methods such as gradient descent have very high complexity when applied to large scale problems. A plethora of different methods try to tackle this issue. Some of these methods use stochastic optimization algorithms that have low computational cost and are scalable enough to process large data sets. Another approach is distributing of the computation task among many computing agents in a network. Models such as MapReduce [1] have been proposed for distributed processing of large data sets. In this work, we employ numerical methods and network message passing protocols to efficiently solve large scale convex optimization problems. Here, we mainly focus on methods that perform coordinate-wise operations in order to reduce complexity.

We first propose a method for the simpler case of centralized optimization that uses a novel non-uniform sampling of the coordinates. In this scheme, the chance of a coordinate j being selected is proportional to partial gradients $\frac{\partial f(w)}{\partial w_j}$. We name this method Proportional Stochastic Coordinate Descent (PSCD). We show that for convex smooth objective functions our algorithm, with constant step size, achieves $\mathcal{O}\left(\frac{1}{t}\right)$ convergence rate in expectation. Our centralized analysis is based on the analysis of the uniform scheme by Sahlev-Shwartz and Tewari [2]. The recent survey of Wright [3] summarizes much of the early work on coordinate descent methods. Many other authors have studied non-uniform sampling algorithms that differ from ours in the coordinate selection method [4–7]. Of particular note is the seminal work of Nesterov, who proves linear convergence rate for his non-uniform method for strongly convex objective functions [4]. Our centralized setup is different in that we consider optimization of convex and smooth objectives rather than strongly convex objectives. Our method, like other first-order gradient methods, has sublinear convergence rate for non-strongly convex objective functions which is not surprising since strongly convex functions (unlike convex functions) cannot be arbitrarily "flat", which means that the algorithm takes sufficiently "steep" steps at every iteration.

Our proportional method can also be adopted for shared memory systems where the nodes (computational agents) are arranged in a star network. For this setup, our algorithm is based on the framework used by Recht et al. [8] where a central node (memory node) keeps the current global decision vector and the rest of nodes (computing nodes) access to this value and update it in an asynchronous manner. In this framework, it is assumed that while each working node is computing its update based on its local objective function value and transmitting it to the central node, other working nodes can also access or update the decision vector. This means that the estimates of the gradient vector that are transmitted to the central node could be obsolete. We propose that each node evaluate its estimate of the gradient according to the PSCD update rule.

Although stochastic optimization methods seem attractive due to their small per-iteration complexity, they show high variance in performance which rises from the high variance in estimates of the gradient vectors. In order to address this issue we propose that we can make use of the previous gradient estimates. The idea is that if the objective function is smooth enough so that gradient values do not change drastically from one point to another in its vicinity, the last gradient estimate is still a good estimate of the current gradient vector. This idea stems from stochastic variance reduced optimization methods proposed in the machine learning literature [9, 10], especially the SAG method presented in [9]. The coordinate-wise method that we propose based on this idea updates a random coordinate of the gradient vector at every iteration while keeping the old values for the other coordinates. We call our method *Stochastic Variance Reduced Coordinate descent* (SVRC).

One drawback of the SAG algorithm is that it requires a large amount of memory to keep the values of sub-gradients at all data points. The coordinate-wise method we present here addresses the memory requirement issue of SAG. In SVRC, the memory required to save the gradient vector is proportional to the dimension of the data d (and not the size the data set n, which is the case in SAG). However, this is achieved at the cost of improvement in the run-time (time to achieve a certain subopyimality gap). In SAG, the per-iteration complexity is improved by a factor of n, while the improvement in SVRC is relative to d.

Large-network paradigms for communication and distributed computation have driven renewed interest in opinion and belief formation models from mathematical sociology and psychology. One such recent work is the novel message passing protocol called *social sampling* [11] that uses limited communication to perform distributed estimation. This protocol is similar to consensus-based multi-agent optimization models – the goal of this work is to investigate the connection between the two. The idea is that every agent performs local processing based on its local objective function, then samples its *belief* or *state* of the global at random to send to its neighbors. Subsequently, agents update their belief based on the messages they receive from their neighbors. Transmitting samples of the belief instead of the complete information makes this method suitable for distributed settings with limited communication resources. We note that for centralized optimization, coordinate descent methods can be considered as centralized variants of the social sampling method where social samples are the partial derivatives.

Despite the popularity and success of stochastic and semi-stochastic gradient methods in centralized settings, when it comes to distributed optimization over general connected networks, from the perspective of one node the gradient information from other nodes is useful only if their current states are not very different from its own. For a strongly convex objective function, this is the case if the estimates are close to the optimum. However, we are interested in methods that guarantee convergence (at least in expectation) to the optimal point regardless of the initial estimate given to the algorithm.

For distributed settings we propose social sampling [11]. We treat the primal iterate as a probability distribution and exchange coordinate samples in the network. This solution may be useful for networks with limited computation and communication resources. Our methods build on the framework developed by Nedić and Ozdaglar [12]. We assume that the nodes broadcast information about their current local decision vectors to their neighbors to cooperatively optimize the global objective function which is the average of the local objective functions. However, in contrast to the mentioned works, our methods rely on sharing partial information with the neighbors, namely information about a small subset of the coordinates.

Our distributed optimization method is related to many existing works in the literature, especially consensus-based algorithms under various assumptions and constraints [13–20]. An over view is provided in chapter 2.

1.1 Notations and Definitions

Throughout this report, superscript i indicates node i of a network, except for e^j that denotes the j-th standard coordinate vector. Furthermore, the discrete time (iteration index) is either represented by subscript t or as the argument of a time-variant function. All element indexes in matrices and vectors are also indicated by subscripts. We denote the set $\{1, \ldots, k\}$ by $[k]$. The vector $\mathbf{1}_A$ for $A \subseteq [d]$ is a d-dimensional vector with 1's for indices $i \in A$ and 0 elsewhere. We denote by $\nabla_j f(x) = \frac{\partial f(x)}{\partial x_j}$ the partial derivative of $f(x)$ w.r.t. its jth coordinate. Assume V is a $d \times d$ matrix. We denote by $\text{diag}(V) \in \mathbb{R}^d$ a vector with elements being diagonal elements of V. Also, $\text{Diag}(V)$ is a $d \times d$ diagonal matrix with the same diagonal elements as V.

Let $\mathcal{D} = \{x_1,\ x_2, \ldots,\ x_m\}$ be a *data set* whose elements, the *data points*, are i.i.d samples drawn from $\mathcal{P}$, a distribution on $\mathcal{R}^d$.

Let $\mathcal{G} = (V, E)$ represent a graph with vertex set $V = \{1, \ldots, n\}$ and edge set $E \subseteq V \times V$. Let $\mathcal{N}^i \subset V$ be the set of the neighbors of node (vertex) i and $\tilde{\mathcal{N}^i} = \mathcal{N}^i \cup i$.

We define projection of a vector w to a convex set $\mathcal{B}$ as $P_{\mathcal{B}}(w) = \underset{w' \in \mathcal{B}}{\text{argmin}}\ \|w - w'\|$.

Definition 1. *A function $f : \mathcal{R}^d \rightarrow \mathcal{R}$ is convex if for all vectors u and v,*

$$f(u) - f(v) \geq \nabla f(v)^\top (u - v)\,. \tag{1.2}$$

Definition 2. *A function $f : \mathcal{R}^d \rightarrow \mathcal{R}$ is λ-strongly convex if for all vectors u and v,*

$$f(u) - f(v) \geq \nabla f(v)^\top (u - v) + \frac{\lambda}{2}\|u - v\|_2^2. \tag{1.3}$$

Definition 3. *A function $f : \mathcal{R}^d \rightarrow \mathcal{R}$ is L-Lipschitz continuous if for all vectors u and v,*

$$\|f(u) - f(v)\|_2 \leq L\,\|u - v\|_2\,. \tag{1.4}$$

Definition 4. *A function $f : \mathcal{R}^d \rightarrow \mathcal{R}$ is L-smooth if it is twice differentiable and has L-Lipschitz continuous gradients. Equivalently, $\mathcal{R}^d \rightarrow \mathcal{R}$ is L-smooth if [21]*

$$f(u) - f(v) \leq \nabla f(v)^T (u - v) + \frac{L}{2}\,\|u - v\|_2\,. \tag{1.5}$$

Definition 5. *A function $f : \mathcal{R}^d \rightarrow \mathcal{R}$ is β-coordinate-wise smooth w.r.t. coordinate j if it is* *twice differentiable and for all vectors u and basis vectors e^j and any scalar η,*

$$f(u+\eta e^j) - f(u) \geq \eta \nabla_j f(u) + \frac{\beta \eta^2}{2}. \tag{1.6}$$

1.1.1 Convergence Notations

The optimal solution to an optimization problem is denoted by w^*. In this work, we use iterative optimization methods that generate a random sequence of iterates $\{w_t\}_{t=0}^T$ (also referred to in this work as estimates, beliefs, or decision vectors) and a sequence of corresponding function values $\{f(w_t)\}_{t=0}^T$.

We study the convergence of our algorithms in terms of guaranteed expected error bounds as functions of the number of iterations T. Error bounds provided for stochastic algorithms are in some cases in terms of

$$\mathbb{E}\left[\|w_t - w^*\|^2\right] \leq \epsilon(t) \tag{1.7}$$

and in most cases in terms of the suboptimality gap $f(w_t) - f(w^*)$

$$\mathbb{E}\left[f(w_t) - f(w^*)\right] \leq \mathcal{E}(t), \tag{1.8}$$

where $\epsilon(t)$ and $\mathcal{E}(t)$ are decreasing function of t. If the objective function $f(w)$ is L-smooth (1.5), then we can relate these two notions: $\mathcal{E}(t) = \frac{L}{2}\epsilon(t)$.

We define *convergence rate* of a method as the rate at which these error bounds converge to zero. We say a sequence $\{x_t\}$ converges linearly to its limit x^* if $\lim\limits_{t\to\infty} \frac{|x_{t+1}-x^*|}{|x_t - x^*|} = C$ where C is a constant. Moreover, a sequence $\{x_t\}$ converges sublinearly to x^* if $\lim\limits_{t\Rightarrow\infty} \frac{|x_{t+1}-x^*|}{|x_t - x^*|} = c(t)$, where $c(t) \to 1$ as $t \to \infty$.

We also use the big ($\mathcal{O}$) notation [22] in describing the asymptotic behavior of some of the algorithms in this work. For functions $f : \mathcal{R}^d \to \mathcal{R}$ and $g : \mathcal{R}^d \to \mathcal{R}$, we have

$$f(w) = \mathcal{O}\left(g(w)\right) \qquad \text{as} \quad t \to \infty$$

if and only if there is a positive real number K and a real number $w^\dagger$ such that $|f(w)| \leq M|g(w)|$ for all $w \geq w^\dagger$.

1.2 Model and Problem Setup

We will consider three types of problems in this work: centralized, shared memory and distributed. For the distributed setting, we make the following assumptions on the network.

In the distributed setup, the optimization task is jointly accomplished by the n processing units that are arranged in a network represented by a graph $\mathcal{G} = (V, E)$ which we assume to be connected; we further assume $(i, i) \in E$ for all i. An $n \times n$ matrix Q is called graph-conformant if $Q_{ik} = 0$ for $(i, k) \notin E$. We consider matrix-valued processes $Q(t)$ where $Q(t)$ is doubly stochastic. We use the notation $Q_{ik}(t) = q^i_k(t)$. We think of $q^i_k(t)$ as the weight that node i assigns to the information from node k at time t. Throughout the paper we assume that the expectation of each stochastic graph-conformant matrix corresponds to a connected graph. Deterministic matrices correspond to connected graphs as well.

1.2.1 Centralized System

First, we consider an optimization problem in a centralized setup. Then, we study the case of optimizing the sum of functions f_i for $i \in [n]$ where each function is associated with one node of a network.

In the centralized problem, we aim to minimize the following objective function:

$$\min_{w \in \mathcal{R}^d} f(w), \tag{1.9}$$

where $f(w)$ is a convex smooth function.

1.2.2 Shared Memory System

Our proportional sampling scheme extends naturally to shared-memory models for distributed optimization. In these models, a common memory element holding the current iterate w_t is accessed by a collection of n processors, each with its own local objective function $f^i(w)$. The goal of such a system is to minimize the average of the local objectives:

$$\min_{w \in \mathcal{R}^d} f_S(w) = \frac{1}{n} \sum_{i=1}^{n} f^i(w), \tag{1.10}$$

where $\{f^i(w)\}_{i=1}^n$ are strongly convex functions with Lipschitz continuous gradients. In this setup, a central node has a memory that keeps a shared estimate vector and all other nodes have access to this node to read or update the estimate. In this work, we assume that each node reads the shared vector at arbitrary times and updates the estimate using its local gradient information.

1.2.3 Distributed System

Similar to shared memory model, in a general connected network, we aim to minimize the average of the local objective functions associated with the nodes of the network:

$$\min_{w \in \mathcal{R}^d} f_D(w) = \frac{1}{n} \sum_{i=1}^{n} f^i(w), \tag{1.11}$$

where $\{f^i(w)\}$ are strongly convex functions.

Chapter 2
Literature Review

2.1 Social Sampling

An important issue in any large-scale distributed optimization setup, like any other large network, is implementing an efficient communication method its nodes. Many such "message passing protocols" have suggested, choosing the right protocol requires taking into account the topology and information flow in the network. One of the rather recent protocols is social sampling [11]. This method can be considered as a randomized version of the consensus method. In a consensus-based setup, every node shares its entire information (belief, state or estimate) of some phenomenon with its neighbors iteratively, until they all converge to the true belief. In social sampling, however, instead of sharing complete information, only partial information (sampled belief) is shared with the neighbors.

Sarwate and Javidi [11] use this idea in a distributed learning problem were nodes of a network locally estimate global empirical distribution (histogram) of opinions in a network:

$$\Pi(j) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\left(X_i = j\right) e^j \qquad \forall j \in [d], \tag{2.1}$$

where X_i is a random variable representing node i's opinion which takes value in a known discrete set of values.

In this setup, at each iteration every node sends an unbiased sample drawn according to its current estimate of the true distribution. Formally, at each time t, each node i maintains an estimate $\Pi_i(t)$ of the true distribution Π and sends its neighbors a random message $Y_i(t) \in \{0,\ e^1,\ e^2, \ldots,\ e^d\}$ drawn according to $\Pi_i(t)$. Each node i assigns a weight $Q_{ik}(t)$ to its neighbor k. Then, nodes update their estimates using the following rule:

$$\Pi_i(t+1) = (1 - \eta_t A_{ii}(t))\Pi_i(t) - \eta_t B_{ii}(t)\ Y_i(t) + \sum_{k \in \mathcal{N}_i(t)} \eta_t Q_{ik}(t) Y_k(t), \tag{2.2}$$

where η_t is a step size and A_{ii} and B_{ii} are problem-specific. The authors show that under certain assumptions, the estimates converge to a common value almost surely.

2.2 Stochastic Optimization Methods

There is a considerable body of work on stochastic optimization methods in general and stochastic coordinate descent methods in particular. These methods are especially popular for large scale machine learning problems where conventional deterministic methods are too slow or seem impractical. In this section some well-known algorithms in the literature are discussed.

2.2.1 Stochastic Gradient Descent for SVM

In the paper "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM", the basic stochastic gradient descent method is applied for optimizing objective functions of support vector machines (SVM) learning problems [23]. The SVM problem with hinge loss function has the following form:

$$\min_{w \in \mathcal{R}^d} f(w) = \frac{\lambda}{2}\|w\|^2 + \frac{1}{m}\sum_{(x,y)\in\mathcal{S}} \ell(w;(x,y)), \tag{2.3}$$

where

$$\ell(w;(x,y)) = \max\{0, 1 - y\langle w, x\rangle\} \tag{2.4}$$

and $\mathcal{S}$ is a data set with m data points.

Algorithm

At every iteration, k samples out of m data points are chosen uniformly at random in order to find an unbiased estimate of the subgradient of the objective function $f(w)$. In other words, at each iteration the gradient of the following approximate objective function is computed:

$$f(w_t) = \frac{\lambda}{2}\|w_t\|^2 + \frac{1}{k}\sum_{(x,y)\in\mathcal{D}_t} \max\{0, 1 - y\langle w_t, x\rangle\}, \tag{2.5}$$

where $\mathcal{D}_t \subset \mathcal{D}$ with $|\mathcal{D}_t| = k$. The computed gradient is the following:

$$\begin{aligned} \widehat{g_t} &= \nabla f_t(w_t) && (2.6)\\ &= \lambda w_t - \frac{1}{k}\sum_{(x,y)\in\mathcal{D}_t} \mathbb{1}[y w_t^T x < 1] y x. && (2.7) \end{aligned}$$

Finally, a projection step is adopted to keep w_t inside the feasible domain $\mathcal{W}$ during the algorithm.

Convergence Analysis

Let $f_1, \ldots, f_T$ be a sequence of λ-strongly convex functions, and $w_1, ..., w_{T+1}$ be a sequence of vectors in $\mathcal{W}$. Assume $w_{t+1} = P_{\mathcal{W}}(w_t - \eta_t g_t)$ where g_t is a subgradient of f_t at w_t and $\eta_t = \frac{1}{\lambda t}$. Assume that for all t, $\|g_t\| \leq L$ and for all data points $(x, y) \in \mathcal{D}$ we have $\|x\| \leq R$. Suppose that $\mathcal{D}_t$ is sampled i.i.d from data set $\mathcal{D}$ for all n. Let r be a number chosen uniformly at random from $[T]$. Then:

$$\mathbb{E}_{\mathcal{D}_1,\ldots,\mathcal{D}_T}\mathbb{E}_r[f(w_r)] \leq f(w^*) + \frac{c \log T}{\lambda T}. \tag{2.8}$$

where c is a constant. Also, for the same conditions, with probability at least $1 - \delta$ we have

$$f(w_T) - f(w^*) \leq \frac{c \log T}{\delta \lambda T}. \tag{2.9}$$

2.2.2 Stochastic Gradient Descent with Suffix Averaging

Authors of "Making Gradient Descent Optimal for Strongly Convex Stochastic Optimization" [24] try to improve the well-known $\mathcal{O}(\frac{\log T}{T})$ convergence guarantee for SGD with averaging to achieve the optimal rate of $\mathcal{O}(\frac{1}{T})$.

Algorithm

The general SGD algorithm is described in the following pseudo-code:

Algorithm 1 Stochastic Gradient Descent

```
Require: $S, \eta_t, T$
  arbitrarily select $w_1 \in \mathcal{W}$
  for $t = 1, 2, \ldots, T$ do
    calculate $\widehat{g_t}$ s.t. $\mathbb{E}[\widehat{g_t}] \in \partial f(w_t)$
    set $w_{t+1} = P_{\mathcal{W}}(w_t - \eta_t \widehat{g_t})$
  end for
  return $w_{T+1}$
```

Note that this is a very general scheme and the results obtained in this paper are valid for different variants of SGD and SCD (or any other stochastic method) as long as they satisfy $\mathbb{E}[\widehat{g_t}] = \nabla f(w_t)$ and some assumptions that we will see in the sequel. The algorithm returns a sequence of points $w_1, \dots, w_T$. To obtain in a final estimate, common methods in the literature are interested in the last point w_T or the average point $\widetilde{w}_T = \frac{1}{T}\sum_{i=1}^{T} w_i$.

However, the authors claim that by averaging just over a portion of the points instead of all points a convergence rate of $\mathcal{O}(\frac{1}{T})$ is achievable even for non-smooth objective functions. This method, called *α-suffix averaging*, for $\alpha \in (0,1)$ returns the following point:

$$w_T^{\alpha} = \frac{w_{T(1-\alpha)+1} + \dots + w_T}{\alpha T}. \tag{2.10}$$

Convergence Analysis

First, we consider smooth functions and show that if we return the last point, we achieve an $\mathcal{O}(\frac{1}{T})$ convergence guarantee.

Suppose f is L-smooth and λ-strongly convex over a convex set $\mathcal{W}$ and $\mathbb{E}[\|g_t\|^2] \le M^2$. Then, if we set step size $\eta_t = c/\lambda t$ for some constant c, for any T we have

$$\mathbb{E}[f(w_T) - f(w^*)] \le \frac{C_1 L M^2}{\lambda^2 T}. \tag{2.11}$$

Also, for the average point $\widetilde{w}_T = \frac{1}{T}\sum\limits_{t=1}^{T} w_t$ we have:

$$\mathbb{E}[f(\widetilde{w}_T) - f(w^*)] \le \frac{C_2 L M^2}{\lambda T}. \tag{2.12}$$

For non-smooth loss functions such as the hinge loss, however, the analyses above do not hold. for some problems it can be proved that the error gap between $f(\widetilde{w}_T)$ and $f(w^*)$ is lower bounded by $\Omega(\frac{\log T}{T})$. The authors of this paper show that by the use of α-suffix averaging, the $\mathcal{O}(\frac{1}{T})$ convergence rate can be recovered:

$$\mathbb{E}[f(w_T^{\alpha}) - f(w^*)] \le \frac{\Big(C_3 + C_4 \log(\frac{1}{1-\alpha})\Big)}{\alpha}\frac{M^2}{\lambda T}, \tag{2.13}$$

Moreover, if $\|g_t\|^2] \le M^2$ with probability 1, we have with probability at least $1-\delta$:

$$\|w_T - w^*\|^2 \le \frac{C_5 M^2}{\lambda^2 T} + \frac{C_6 M^2 \log(\log(T)/\delta)}{\lambda T}. \tag{2.14}$$

In these inequalities, $C_1, \dots, C_6$ are constants.

2.2.3 Reducing Computational Cost Using Stale Gradients

Our work in Chapter 4 is based on the algorithm suggested in "Minimizing Finite Sums with the Stochastic Average Gradient" by Schmidt, Le Roux, and Bach [9]. The authors present a variance reduced variant of SGD for optimizing strongly convex cost functions that can be written as sum of a finite set of smooth functions. The so-called "Stochastic Average Gradient" (SAG) method achieves an exponential convergence rate by keeping the the previous gradient value in memory.

Stochastic Average Algorithm

Objective functions in machine learning problems can usually be written as a sample average (or sum) over a finite data set:

$$f(w) = \frac{1}{m}\sum_{i=1}^{m} f^i(w). \tag{2.15}$$

Therefore, the full gradient method for computing the gradient of $f(w)$ needs to find the gradient of all f_i functions. On the other hand, stochastic gradient descent (SGD) randomly selects one i (or a small number of them in case of minibatching) and takes the gradient at that (those) point(s) to find the approximate gradient. The SAG method combines these methods by updating the solution w according to the following rule:

$$w_{t+1} = w_t - \eta_t \, \frac{1}{m}\sum_{i=1}^{m} \widehat{g}_i(t), \tag{2.16}$$

where

$$\widehat{g}_i(t) = \begin{cases} \nabla f^i(w_t) & \text{if } \ i \in \mathcal{D}_t \, , \\ \widehat{g}_i(t-1) & \text{otherwise.} \end{cases} \tag{2.17}$$

The set $\mathcal{D}_t$ is a subset of data set $\mathcal{D}$ which is randomly selected so that, similar to SGD, the gradient at a few data points is calculated. For the rest of data points, however, we use the gradient value from the previous steps, as opposed to setting it to zero which is the case in SGD. As we will soon see in the analysis, this methods achieves a linear convergence rate similar to that of standard gradient descent, while enjoying low per-iteration computational complexity of SGD. However, it requires a huge amount of memory for large scale problems to keep track of the gradient values.

Convergence Rate

Assume that in 2.15 f is λ-strongly convex and f^i is L-smooth for all i. Let the step size be $\frac{1}{2mL}$. Then, for any T it holds that

$$\mathbb{E}\left[f(w_T) - f(w^*)\right] \leq \left(1 - \frac{\lambda}{8Lm}\right)^T \frac{L}{2}\left[3\|w_0 - w^*\|^2 + \frac{9\ \sum_{i=1}^{n} \|\nabla f^i(w^*)\|^2}{4nL^2}\right]. \tag{2.18}$$

Variance Reduction

In order to show how SAG can improve the variance of the results, we write its update rule in the following from:

$$w_{t+1} = w_t - \eta_t \left[\frac{\nabla f^i(w_t) - \nabla f^i(\Phi_t)}{m} + \frac{1}{m}\sum_{k=1}^{m} \nabla f^k(\Phi_t)\right]. \tag{2.19}$$

The idea behind variance reduction methods is as follows [10]. Say we have random variable X and we are interested in estimating $\mathbb{E}[X]$. Moreover, we have random variable Y which is highly correlated with X and we can easily compute $\mathbb{E}[Y]$. Now, define $\theta_\alpha = \alpha\,(X - Y) + \mathbb{E}[Y]$ as an estimator of $\mathbb{E}[X]$. We observe that $\mathbb{E}[\theta_\alpha] = \alpha\mathbb{E}[X] + (1-\alpha)\,\mathbb{E}[Y]$ and $Var\,(\theta_\alpha) = \alpha^2\,[Var(X) + Var(Y) - 2\ Cov(X,Y)]$. We observe that for large enough covariance, $Var\,(\theta_\alpha)$ can be smaller than $Var(X)$. Also, note that parameter α can be used to further control the variance while taking into consideration that smaller variance (smaller α) is achieved at the cost of more bias in the estimate.

In (2.15), $X = \nabla f^i(w_t)$ and $Y = \nabla f^i(\Phi_t)$ and $\alpha = \frac{1}{n}$, so we have considerable improvement in variance while enduring some bias.

2.2.4 Unbiased SAG

Defazio, Bach, and Lacoste-Julien [10] present another variance-reduced stochastic gradient algorithm. However, unlike the SAG method, the estimates of the gradient vector are unbiased. The update rule in this method is as follows:

$$w_{t+1} = w_t - \eta_t \left[\nabla f^i(w_t) - \nabla f^i(\Phi_t) + \frac{1}{m}\sum_{k=1}^{m} \nabla f^k(\Phi_t)\right]. \tag{2.20}$$

where $\nabla f_i(\Phi_t)$ is the current stored gradient estimate and the index i is selected uniformly at random from $[m]$. Let $\widehat{g}_t = \nabla f^i(w_t) - \nabla f^i(\Phi_t) + \frac{1}{m}\sum\limits_{k=1}^{m} \nabla f^k(\Phi_t)$. Note that $\mathbb{E}_i\left[\widehat{g}_t\right] = \frac{1}{m}\sum\limits_{k=1}^{m} \nabla f^k(w_t) - \frac{1}{m}\sum\limits_{k=1}^{m} \nabla f^k(\Phi_t) + \frac{1}{m}\sum\limits_{k=1}^{m} \nabla f^k(\Phi_t) = \nabla f(w_t)$ which shows that $\widehat{g}_t$ is an unbiased estimate of the true gradient vector. The unbiasedness of the estimate makes it easier to analyze the convergence of the algorithm, however, this comes at the cost of having higher variance (for SAGA, α is n times that of SAG).

Convergence Results

Consider optimization problem with objective function in form of (2.15). For λ-strongly convex L-smooth $f_i(w)$, using SAGA algorithm with step size $\eta = \frac{1}{2(\lambda m+L)}$, after T iterations we have:

$$\mathbb{E}[f(w_T) - f(w^*)] \leq$$
$$\left(1 - \frac{\lambda}{2\left(\lambda m + L\right)}\right)^T \frac{L}{2}\left[\|w_0 - w^*\|^2 + \frac{m\left[f(w_0) - f(w^*) - (w_0 - w^*)^\top \nabla f(w^*)\right]}{\lambda m + L}\right]. \tag{2.21}$$

Also, if we relax the strong convexity condition and use step size $\eta = \frac{1}{3L}$, we get

$$\mathbb{E}[f(\widetilde{w}_T) - f(w^*)] \leq \left(\frac{4m}{T}\right)\left[\frac{2L}{m}\|w_0 - w^*\|^2 + f(w_0) - f(w^*) - (w_0 - w^*)^\top \nabla f(w^*)\right], \tag{2.22}$$

where $\widetilde{w_T} = \frac{1}{T}\sum_{t=0}^{T} w_t$.

2.2.5 Stochastic Coordinate Descent

Authors of "Stochastic Methods for ℓ_1-Regularized Loss Minimization" [2] suggest another stochastic method for efficiently solving ℓ_1-regularized loss minimization problems, that is:

$$\min_{w\in\mathcal{R}^d} f(w) = \lambda\|w\|_1 + \frac{1}{m}\sum_{i=1}^{m}\ell(w;(x_i, y_i)). \tag{2.23}$$

In order to get rid of the absolute value terms which are non-differentiable, the method presented here solves the following equivalent problem:

$$\min_{w\in\mathcal{R}^{2d}} f(w) = \lambda\sum_{j=1}^{2d} w_j + \frac{1}{m}\sum_{i=1}^{m}\ell(w;(\widehat{x}_i, y_i)), \tag{2.24}$$

such that $w > 0$, where $\widehat{x}_i = [x_i; -x_i]$. The authors claim that if $v^* \in \mathcal{R}^{2d}$ is the minimizer of the problem 2.24, then $w^* \in \mathcal{R}^d$ defined by $w^*_j = v^*_{d+j} - v^*_j$ minimizes the original problem 2.23. Initializing w to be zero, at each iteration t a coordinate j is picked uniformly at random from $[2d]$. Then the derivative of $f(w)$ w.r.t. $(w_j(t)$, $g_j(t) = \nabla_j f(w(t))$, is calculated. We assume that $f(w)$ is L-coordinate-wise smooth. Then, the update rule is the following:

$$w(t+1) = w(t) - \eta(t)e^j, \tag{2.25}$$

where e^j is the jth unit vector and $\eta_n = \max\{w_{j_n}, \frac{g_{jn}}{\beta}\}$. The step size is trimmed to ensure that the condition $w \geq 0$ is always satisfied. The pseudo-code of Stochastic Coordinate Descent (SCD) is given in Algorithm 2.

Algorithm 2 Stochastic Coordinate Descent

```
Require: $S, T, L$
  set $w_1 = 0$
  for $t = 1, 2, \ldots, T$ do
    sample $j$ uniformly at random from $[2d]$
    set $g_j(t) = \nabla_j f(w(t))$
    set $\eta(t) = \min\{w^j(t), \frac{g_j(t)}{L}\}$
    set $w(t+1) = w(t) - \eta(t)$
  end for
  return $w(T+1)$
```

Convergence Rate of SCD

Let $f(w)$ be a convex and L-coordinate-wise smooth function. Then, it holds for every output of the SCD algorithm, w_T, that [2]

$$\mathbb{E}[f(w_T) - f(w^*)] \leq \frac{d\left(L\|w^*\|^2 + 2f(0)\right)}{2T}. \tag{2.26}$$

Proof Sketch. Define the "double potential" function $\psi(w) = \frac{L}{2}\|w - w^*\|^2 + f(w)$. Using smoothness of f, we can prove that $\psi(w) - \psi(w - \eta e^j) \geq (w_j - w_j^*)g_j$. Taking expectation of both sides w.r.t. the choice of the coordinate j, we get $\mathbb{E}\left[\psi(w_t) - \psi(w_{t+1})\right] \geq \frac{1}{d}\mathbb{E}\left[\nabla f(w_t)^T(w_t - w^*)\right]$. But by convexity of f, we have $\frac{1}{d}\mathbb{E}\left[\nabla f(w_t)^T(w_t - w^*)\right] \geq \frac{1}{d}\mathbb{E}\left[f(w_t) - f(w^*)\right]$. Considering the fact that $\mathbb{E}[f(w_t) - f(w^*)]$ is monotonically non-increasing, by summing over t we obtain $\mathbb{E}\left[f(w_{T+1}) - f(w^*)\right] \leq \mathbb{E}\left[\frac{1}{T}\sum_{n=1}^{T} f(w_t) - f(w^*)\right] \leq \frac{d}{T}\left[\psi(w_1) - \psi(w_{T+1})\right]$. By doing some algebraic manipulation we get the result in 2.26.

We use the idea of this proof in Chapter 3 to prove the convergence of our proportional coordinate descent method.

2.3 Optimization in Shared Memory Systems

In this section we discuss some of the existing works on shared memory models introduced in Chapter 1.

2.3.1 Asynchronous Parallel SCD

Liu and Wright [25] propose an asynchronous parallel stochastic coordinate descent method for optimizing a cost function in the following form:

$$f(w) = g(w) + h(w), \tag{2.27}$$

where $g(w)$ is smooth and convex and $h(w)$ is convex and "separable" in the sense that it can be written as $h(w) = \sum_{j=1}^{d} g_j(w_j)$ where j is the coordinate index.

Algorithm

In this method, each processor updates a random coordinate of the vector w independent of the other processors, so many coordinates might be updated simultaneously and the vector that is read by processor i can be different from the vector that it is writing the update to, since some of the coordinates might be updated by other processors during the update process by i. The pseudo-code for the algorithm is described below. Note that this describes the procedure in a single processor, and the same procedure is done in every processor at the same time.

Algorithm 3 Asynchronous SCD

```
Require: $S, T, L, \lambda$
  set $w(1) = 0$
  set $\eta = \frac{\lambda}{L}$
  for $t = 1, 2, \ldots, T$ do
    sample $j$ uniformly at random from $[d]$
    set $g^j(t) = \nabla_j f(w(t))$
    set $w(t+1) = w(t) - \frac{\eta}{L} g^j(t) e^j$
  end for
  return $w(T+1)$
```

Convergence Analysis

Assume that $f(w)$ is convex and L-smooth. Then, for certain choices of the step size η, for convex $f(w)$ we have

$$\mathbb{E}[f(w_T) - f(w^*)] \leq \frac{d\left(L\|w_0 - w^*\|^2 + f(w_0) - f(w^*)\right)}{T + d} \tag{2.28}$$

and for λ-strongly convex function $f(w)$ it holds that

$$\mathbb{E}\left[f(w_T) - f(w^*)\right] \leq \left(1 - \frac{\lambda}{d(\lambda + 2L)}\right)^T d\left(L\|w_0 - w^*\|^2 + f(w_0) - f(w^*)\right). \tag{2.29}$$

We can see that the reduction factor after d iterations of this algorithm is $\left(1 - \frac{\lambda}{d(\lambda+2L)}\right)^d \approx 1 - \frac{\lambda}{\lambda+2L}$, while for the standard gradient descent method the rate constant is $1 - \frac{2\lambda}{\gamma}$ if we apply the algorithm to L-smooth λ-strongly convex functions. Thus, for some values of the smoothness parameter L, the asynchronous SCD method may need less computation than the standard GD.

Finally, under the same conditions and parameter values as before, with probability higher than $1 - \delta$, it holds that

$$f(w_T) - f(w^*) \leq \epsilon, \tag{2.30}$$

if we run the algorithm at least T times, where for convex functions

$$T = \frac{d\left(L\|w_0 - w^*\|^2 + f(w_0) - f(w^*)\right)}{\epsilon\rho} - d \tag{2.31}$$

and for λ-strongly convex functions

$$T = \frac{d(\lambda + 2L)}{\lambda}\left|\log\frac{L\|w_0 - w^*\|^2 + f(w_0) - f(w^*)}{\epsilon\rho}\right|. \tag{2.32}$$

2.3.2 A Lock-Free Method

The HOGWILD! method [8] is another parallel stochastic optimization method for shared memory systems. This algorithm, like the algorithm in previous section, is also lock-free, in the sense that the nodes can access and write on the decision vector at any time and the memory is not locked when the vector is being updated by any node. Here, the objective function is separable:

$$f(w) = \sum_{j=1}^{d} f_j(w_j), \tag{2.33}$$

and each node updates a random component i at arbitrary times:

$$w_j(t+1) = w_j(t) - \eta \; d \; \frac{\partial f_j(w(t))}{\partial w_j(t)}, \tag{2.34}$$

or:

$$w(t+1) = w(t) - \eta \; d \; \frac{\partial f_j(w(t))}{\partial w_j(t)} e^j. \tag{2.35}$$

Convergence Results

Consider using HOGWILD! to solve the optimization problem (2.33) with f being β-smooth and λ-strongly convex with $\|\frac{\partial f_j(w)}{\partial w_j}\| \leq M$ for all $w \in R^d$. Assume that the number of updates written to w by other nodes while a certain node is updating it is bounded by τ. Define $P_2(x)$ to be a second order polynomial in x. With step size

$$\eta = \frac{c_1 \epsilon}{L M^2 P_2(\tau)}, \tag{2.36}$$

and for large enough number of iterations, i.e.

$$T > \frac{L M^2 P_2(\tau) \log(L \|w_0 - w^*\|^2 / \epsilon)}{c_2 \epsilon}, \tag{2.37}$$

for any $\epsilon > 0$ and some constants c_1 and c_2, we have $\mathbb{E}\left[f(w(T)) - f(w^*)\right] \leq \epsilon$. Note that this is an expectation guarantee and not a w.h.p. guarantee as in the previous work mentioned in Subsection 2.3.1.

2.4 Distributed Optimization

There is a plethora of works on distributed optimization. A few related works are discussed here.

2.4.1 Distributed Asynchronous Gradient Optimization

In this seminal work by Tsitsiklis et al., a model for distributed asynchronous optimization is proposed [18]. In this model, n nodes compute and communicate their estimates of components of a decision vector $w = \left(w^1, \dots,\ w^d\right)$. Nodes have access to a global objective function and computation of components of the decision vector is distributed among them. Each node, at each step, performs some computations to update its belief on some components of the decision vector and also combines its own estimate with the last estimates it has received from other nodes (either directly or through some intermediate nodes). Let t^j_{ik} be the time the last message containing w^j sent from node k to node i is computed.

Communication assumptions

In this work the following assumption on network communications hold:

- A directed link from node k to node i exists for component w^j if and only if k sends infinite number of messages with a value of w^j to node i.
- The lag between any two consecutive such messages is bounded.
- There are constants a and b such that at least one message is sent during interval $[at^b,\ a(t+1)^b]$ and number of the messages during this interval is bounded.
- The communication delays, $t - t^j_{ik}$, are bounded.

Update rule

Each node updates the value of component j of its estimate according to the following rule:

$$w^j_i(t+1) = \sum_{k=1}^{n} Q^j_{ik} w^j_k(t^j_{ik}) + \eta_i(t) s^j_i(t), \tag{2.38}$$

where Q^j is the weight assigned by node i to opinions from node k, and $\eta_i(t)$ is a step size used by node i. $s^j_i(t)$ is some computation locally performed by node i, e.g. in distributed stochastic approximation it is defined as $s^j_i(t) = -\frac{\partial f(w_t)}{\partial w^j} + \mathcal{E}_i(t)$, a noisy unbiased estimate of the partial gradient w.r.t. the jth component.

Convergence results

Consider solving distributed problem (1.11) with a smooth objective function. Under assumptions mentioned above and some further assumptions on the noise term, the algorithm presented in this section converges almost surely for both constant and decaying step size.

2.4.2 Consensus-Based Distributed Subgradient Optimization

Unlike the work mentioned above [18], in the seminal work by Nedić and Ozdaglar [12] each node (agent) has access to a local function and the goal of the optimization problem is to minimize the sum (or average) of this local objectives. In this model, nodes share their current estimates with their neighbors. Each node performs a local gradient step on its local objective function at its current estimate $(f_i(w_i(t)))$ and combines its own belief with those of its neighbors.

Update rule

Each node updates its estimate according to the following update rule:

$$w_i(t+1) = \sum_{k=1}^{n} Q_{ik}(t) w_i(t) - \eta_i(t) g_i(t), \tag{2.39}$$

where $g_i(t)$ is a subgradient of $f_i(w_t)$.

Communication assumptions

It is assumed here that:

- A directed link exists from k to i if and only if j communicates directly with i.
- The weight coefficients are time dependent which means that some existing links may be inactive (zero weight) in some iterations. Therefore, the network connectivity graph is also time dependent.
- Every agent's information at any time will reach each and every other node directly or indirectly.
- The interval between two consecutive messages from each node to its neighbors is bounded.
- The links are bidirectional with the same weight in both directions.
- The weight matrix $Q(t)$ is doubly stochastic.

Convergence results

Consider solving distributed problem (1.11) with convex objective function. Assume that the subgradients of the local functions are bounded by L. Under these assumptions and the above-mentioned communication assumptions, with constant step size $\eta_i(t) = \eta$ for all $i \in [n]$ and t, we have

$$f(\widetilde{w}_T^i) - f(w^*) \leq \frac{n\|\bar{w}_0 - w^*\|^2}{2\eta T} + \eta C L^2, \tag{2.40}$$

where $\widetilde{w}_T = \frac{1}{T}\sum_{t=1}^{T} w_t^i$ and $\bar{w}_t = \frac{1}{n}\sum_{i=1}^{n} w_t^i$.

2.4.3 Consensus-based optimization Over Random Networks

This work builds on the works by Nedić and Ozdaglar [12] and Tsitsiklis et al. [18] mentioned in this section. The main difference here is that the availability of a link between any two nodes is a random process. Therefore, at any time, link availability between among nodes is a random event.

Update rule

Each node updates its estimate in in this way:

$$w_i(t+1) = \sum_{k=1}^{n} Q_{ik}(t) w_i(t) - \eta(t) g_i(t), \tag{2.41}$$

where $g_i(t)$ is a subgradient of $f_i(w_i(t))$. Note that all nodes use the same step size $\eta(t)$.

Communication assumptions

The main assumptions here are:

- A directed link exists from k to i if and only if j communicates directly with i.
- The link availability is random at any time, therefore the the weight matrix is a random matrix and the connectivity graph is a random graph.
- The average connectivity graph is strongly connected.
- The weight matrices $Q(t)$ are doubly stochastic with probability 1.

Convergence results:

Consider solving distributed problem (1.11) with convex objective function. Assume that the subgradients of the local functions are bounded by L. Under these assumptions and the above-mentioned communication, with diminishing step sizes satisfying $\sum_{t=1}^{\infty} \eta(t) = \infty$ and $\sum_{t=1}^{\infty} \eta^2(t) < \infty$, we have:

$$\lim_{t \to \infty} w_i(t) = w^* \qquad w.p.\ 1 \tag{2.42}$$

and for constant step size we have the following result:

$$\limsup_{T \to \infty} |f(\widetilde{w}_i(T)) - f(w^*)| \leq nCL^2 \qquad w.p.\ 1, \tag{2.43}$$

where $\widetilde{w}_i(T) = \frac{1}{\sum_{t=1}^{T} \eta(t)} \sum_{t=1}^{T} \eta(t) w_i(t)$ and C is a constant.

2.4.4 Asynchronous Broadcast-Based Distributed Optimization

This asynchronous method [13] also builds on the works by Nedić and Ozdaglar [12] and Tsitsiklis et al. [18], as well as the paper by Aysal et al. [26]. In this paper, a broadcast-based asynchronous distributed optimization algorithm is proposed and its convergence behavior is studied. The model used here takes into account two types of uncertainty: *i)* link availability is random due to possible link failures and *ii)* agents use noisy estimates of their local gradient vectors to update their beliefs. In this setup, all nodes are by default in sleep mode. At random times, a node i (only one node at a time) sends its current estimate of the decision vector to its neighbors. Only a random subset of its neighbors receive the message because of link failures. These nodes wake up and combine their current estimate with that of node i and perform a noisy local gradient step. Then, they go back to sleep mode until they receive another message. The distributed optimization problem solved in this paper is the following constrained problem:

$$\begin{aligned} & f(w) = \frac{1}{m} \sum_{i=1}^{m} f_i(w) \\ & s.t. \qquad w \in \mathcal{W}, \end{aligned} \tag{2.44}$$

where $\mathcal{W} \subset \mathcal{R}^d$ is a convex set.

Update rule

If node i broadcasts at time t and a subset $\mathcal{J}_i$ of its neighbors receive its message, for node i and nodes $k \notin \mathcal{J}_i$ we have

$$w_k(t+1) = w_k(t) \tag{2.45}$$

and for $k \in \mathcal{J}_i$ we have the following two-step update rule:

$$\begin{aligned} v_k(t) &= \beta w_i(t) + (1-\beta) w_k(t) \\ w_k(t+1) &= P_{\mathcal{W}}(v_k(t)) - \eta_k(t)\widehat{g}_k(t), \end{aligned} \tag{2.46}$$

where $P_{\mathcal{W}}$ is the projection on the feasible set $\mathcal{W}$ and $\widehat{g}_k(t) = \nabla f_k(v_k(t)) + \mathcal{E}_k(t)$. Note that $\mathcal{E}_k(t)$ is zero mean noise.

Communication assumptions

Here the following assumptions hold:

- A directed link exists from k to i if and only if j communicates directly with i.
- The underlying network is connected.
- The link failures process is i.i.d, that is the probability of failure of each link is constant over time and failure of a link is independent of other links in the network.

Convergence results

Consider solving distributed problem (2.44) with convex objective function and closed convex constraint set. Assume that the subgradients of the local functions are bounded by L. Also, assume that the error in computing the gradient is bounded: $\mathbb{E}\left[\mathcal{E}_k^2(t)|\mathcal{F}_{t-1}, i, \mathcal{J}_i\right] \leq \nu^2$ and zero mean: $\mathbb{E}\left[\mathcal{E}_k(t)|\mathcal{F}_{t-1}, i, \mathcal{J}_i\right] = 0$ with probability 1 for $k \in \mathcal{J}_i$. Under these assumptions and the above-mentioned communication assumptions, with diminishing step sizes satisfying $\sum_{t=1}^{\infty} \eta(t) = \infty$ and $\sum_{t=1}^{\infty} \eta^2(t) < \infty$, we have that the sequence of the estimates of the nodes converge to the same random point in the optimal set.

Moreover, with constant step size $\eta_i(t) = \eta_i$ for averaged iterates $\widetilde{w}_i(T) = \frac{1}{T}\sum\limits_{t=1}^{T} w_i(t)$, we get

$$\limsup_{T\to\infty} |f(\widetilde{w}_i(T)) - f(w^*)| \leq nC_1(L+\nu)^2 + C_2 n\sqrt{n} + C_3\sqrt{n} \qquad w.p.\ 1, \qquad (2.47)$$

where C_1, C_2, and C_3 are constants.

Chapter 3
Proportional Stochastic Coordinate Descent

In this chapter we propose and analyze a non-uniform variant of stochastic coordinate descent which we call *Proportional Stochastic Coordinate Descent (PSCD).* In this method, unlike conventional Stochastic coordinate descent, a random coordinate is selected at each step according to a non-uniform probability distribution. Here, we treat the gradient vector as a probability distribution, i.e. the probability of selecting the jth coordinate is proportional to the jth coordinate of the gradient vectors, $\nabla_j f(w)$. The idea behind this method is that this biased sampling method selects the "most important" coordinate, the coordinate along which the function has the largest reduction, at each step. This intuitively results in reduced variance of results across the sample paths, since in this method only a limited number of "important" paths have high probability of being selected.

We first study this method for centralized problems. Then, we discuss its application to shared memory systems and to complete networks where all the nodes are connected.

3.1 Centralized Proportional Stochastic Gradient descent

In order solve the centralized minimization problem in form of

$$\min_{w \in \mathcal{R}^d} f(w), \tag{3.1}$$

where $f(w)$ is a convex smooth function, we use a variant of the stochastic coordinate descent method, which we call centralized Proportional Stochastic Gradient Descent (centralized PSCD). At every iteration t, a coordinate j is randomly selected and the j-th coordinate of $w(t)$ is updated:

$$w(t+1) = w(t) - \eta \widehat{g}(t), \tag{3.2}$$

where $\widehat{g}(t) = C(t)e^j$ is an unbiased estimate of the gradient vector on a single coordinate: $\mathbb{E}\left[\widehat{g}(t)\right] = g(t)$. In this algorithm the coordinates are selected according to the following distribution:

$$P(j) = \frac{|g_j(t)|}{\|g(t)\|_1}, \tag{3.3}$$

where $g(t) \in \partial f(w(t))$ is a sub-gradient of $f(w(t))$ and $g_j(t)$ is a sub-derivative of $f(w(t))$ w.r.t. the j-th coordinate. Considering that $\mathbb{E}\left[\widehat{g}(t)\right] = \mathbb{E}\left[C(t)e^j\right] = C(t)\sum_j \frac{|g_j(t)|}{\|g(t)\|_1}e^j$ while $\mathbb{E}[\mathrm{sgn}(g_j(t)) \cdot \|g(t)\|_1 \cdot e^j] = g(t)$, we need to set $C(t) = \|g(t)\|_1 \,\mathrm{sgn}(g_j(t))$. In this setup, we use constant step size $\eta = \frac{1}{\alpha L}$ where L is the maximum component-wise Lipschitz constant of $f(w)$ and α is a constant.

The pseudo-code for the centralized setup is given in Algorithm 4.

Algorithm 4 Centralized PSCD

```
Require: $\lambda$, $L$, $T$
  set $w_0 = 0$
  set $\eta = \frac{1}{\alpha L}$
  for $t = 1, 2, \ldots, T$ do
    calculate $g(t) \in \partial f(w(t))$
    select $j$ according to $P(j) = \frac{|g_j(t)|}{\|g(t)\|_1}$ for $j \in [d]$
    set $w(t+1) = w(t) - \eta(t)\,\|g(t)\|_1 \,\mathrm{sgn}(g_j(t))e^j$
  end for
  return $w(T+1)$
```

3.1.1 Convergence Analysis

Theorem 1. *Consider Algorithm 4 for solving problem* (3.1) *when f is convex with L-Lipschitz continuous component-wise gradients. With constant step size $\eta = \frac{1}{\alpha L}$ we have:*

$$\mathbb{E}\left[f(w(t)) - f(w^*)\right] \leq \frac{\alpha\left(\Psi(w_0) - \Psi(w^*)\right)}{T}, \tag{3.4}$$

where

$$\Psi(w) = f(w) + \frac{L}{2}\|w - w^*\|^2. \tag{3.5}$$

Proof. To prove this theorem we need the following Lemma, which is a corollary of Theorem 2.1.5 in the book of Nesterov [21].

Lemma 1. *Suppose that function $f(w)$ has component-wise Lipschitz continuous gradient:*

$$|\nabla_j f(w + he^j) - \nabla_j f(w)| \leq L^j |h|,$$

Then we have

$$f(w+he^j)-f(w)\le \left\langle \nabla f(w), he^j\right\rangle + \frac{L^j}{2}|h|^2. \tag{3.6}$$

To find an upper bound on the optimality gap in the centralized setup, we will use the preceding Lemma. Following the approach taken by Shalev-Shwartz and Tewari [2], define the potential function $\Psi(w)$ in (3.5), where $w^* = \operatorname{argmin}_w f(w)$ is the minimizer of the objective function $f(w)$ and L is the maximum component-wise Lipschitz constant of $f(w)$. Using this potential function we will prove that under some condition for λ our suggested method for updating $w(t)$ will converge to the optimal solution. Define $\gamma(t) = \eta C(t) = \eta \left\|g(t)\right\|_1 \operatorname{sgn}(g_j(t))$ so that the update is $\eta\widehat{g}(t) = \gamma(t)e^j$. Consider the difference of the potential across one iteration:

$$\begin{aligned}
\Psi(w(t))-\Psi(w(t+1)) =& f(w(t))-f(w(t+1))+\frac{L}{2}(\|w(t)-w^*\|^2-\|w(t+1)-w^*\|^2)\\
\overset{(a)}{\ge}& -g(t)^\top\left(w(t+1)-w(t)\right)-\frac{L\gamma(t)^2}{2}\\
&+\frac{L}{2}\left(w(t)-w(t+1)\right)^\top\left(w(t)+w(t+1)-2w^*\right)\\
=& -g(t)^\top\left(-\gamma(t)e^j\right)-\frac{L\gamma(t)^2}{2}+\frac{L}{2}\left(\gamma(t)e^j\right)^\top\left(2w(t)-2w^*-\gamma(t)e^j\right)\\
=&\gamma(t)g_j(t)-\frac{L\gamma(t)^2}{2}+\frac{L}{2}\left(\gamma(t)(2w_j(t)-2w_j^*-\gamma(t)^2\right)\\
=&\gamma(t)g_j(t)-L\gamma(t)^2+L\gamma(t)\left(w_j(t)-w_j^*\right),
\end{aligned} \tag{3.7}$$

where (a) follows from Lemma 1.

Let $\mathcal{F}(t)$ be the σ-algebra generated by the random coordinate choices up to time t. If we take the conditional expectation of both sides, we will have the following inequality, which is averaged over the choice $J(t)$ at time t:

$$\begin{aligned}
\mathbb{E}\left[\Psi(w(t))-\Psi(w(t+1))|\mathcal{F}(t)\right] &\ge \sum_{j=1}^{d}\frac{\gamma(t)|g_j(t)|g_j(t)}{\|g(t)\|_1}-L\gamma(t)^2+L\sum_{j=1}^{d}\frac{\gamma(t)|g_j(t)|\left(w_j(t)-w_j^*\right)}{\|g(t)\|_1}\\
&=\sum_{j=1}^{d}\frac{|g_j(t)|^2}{\alpha L}-\frac{\|g(t)\|_1^2}{\alpha^2 L}+\sum_{j=1}^{d}\frac{g_j(t)\left(w_j(t)-w_j^*\right)}{\alpha}\\
&=\frac{\|g(t)\|_2^2}{\alpha L}-\frac{\|g(t)\|_1^2}{\alpha^2 L}+\frac{1}{\alpha}g(t)^\top(w(t)-w^*).
\end{aligned} \tag{3.8}$$

We can see that if $\frac{\|g(t)\|_2^2}{\alpha L}-\frac{\|g(t)\|_1^2}{\alpha^2 L}$ has a non-negative value, then we will have the following inequality:

$$\mathbb{E}\left[\Psi(w(t)) - \Psi(w(t+1)) | \mathcal{F}(t)\right] \geq \frac{1}{\alpha} g(t)^\top (w(t) - w^*), \tag{3.9}$$

meaning that α must satisfy the following condition:

$$\frac{\|g(t)\|_2^2}{\alpha L} - \frac{\|g(t)\|_1^2}{\alpha^2 L} \geq 0 \Rightarrow \alpha \geq \frac{\|g(t)\|_1^2}{\|g(t)\|_2^2}. \tag{3.10}$$

Since we have the bound $\frac{\|g(t)\|_1^2}{\|g(t)\|_2^2} \leq d$, it suffices to set $\alpha \geq d$.

By taking the expectation with respect to the entire history up to time t, we have

$$\mathbb{E}\left[\Psi(w(t)) - \Psi(w(t+1))\right] \geq \frac{1}{\alpha} \mathbb{E}\left[g(t)^\top (w(t) - w^*)\right]. \tag{3.11}$$

The convexity of f implies

$$\mathbb{E}\left[f(w(t)) - f(w^*)\right] \leq \alpha \mathbb{E}\left[\Psi(w(t)) - \Psi(w(t+1))\right]. \tag{3.12}$$

Considering the fact that $f(w(t)) - f(w^*)$ is a monotonically non-increasing sequence with respect to t, summing over t gives us

$$\begin{aligned} T\mathbb{E}\left[f(w(T)) - f(w^*)\right] &\leq \mathbb{E}\left[\sum_{t=0}^{T-1} (f(w(t)) - f(w^*))\right] \\ &\leq \alpha \mathbb{E}\left[\sum_{t=0}^{T-1} (\Psi(w(t)) - \Psi(w(t+1)))\right] \\ &\leq \alpha \left(\Psi(w_0) - \Psi(w(T))\right). \end{aligned} \tag{3.13}$$

Therefore:

$$\begin{aligned} \mathbb{E}\left[f(w(T)) - f(w^*)\right] &\leq \frac{\alpha \left(\Psi(w_0) - \Psi(w(T))\right)}{T} \\ &\leq \frac{\alpha \left(\Psi(w_0) - \Psi(w*)\right)}{T}. \end{aligned} \tag{3.14}$$

□

3.2 Distributed Proportional Stochastic Gradient descent

We propose that the proportional gradient sampling method can be adopted in some distributed setups, namely shared memory systems and complete networks where all of the nodes directly communicate with each other.

3.2.1 DPSCD for Shared Memory System

Our proportional sampling scheme extends naturally to shared-memory models for distributed optimization. In these models, a common memory element holding the current iterate $w(t)$ is accessed by a collection of p processors. The goal of such a system is to minimize a function which can be written as sum of n functions:

$$\min_{w \in \mathcal{R}^d} f_S(w) = \frac{1}{n} \sum_{i=1}^{n} f^i(w), \tag{3.15}$$

where $\{f^i(w)\}_{i=1}^n$ are strongly convex functions with Lipschitz continuous gradients. In this setup, a central node has a memory that keeps a shared estimate vector and all other nodes have access to this node to read or update the estimate. The nodes also have access to functions $\{f^i\}$ and choose one of these functions uniformly at random when they update decision vector. If the nodes operate synchronously, the algorithm will be essentially performing the conventional unbiased stochastic gradient descent on $f_S(w)$.

Here, we focus on the more challenging asynchronous setup, which can be considered as a modified version of HOGWILD! [8]. Our proposed method for this setting, called asynchronous distributed PSCD, assumes that each node reads the shared vector at arbitrary times and updates the estimate using its local gradient information.The update rule in this method is

$$w(t) = w(t) - \eta\, \widehat{g}^i(\theta), \tag{3.16}$$

where η is a constant step size and $\widehat{g}^i(\theta) = \left\|g^i(\theta)\right\|_1 \cdot \operatorname{sgn}\left(g^i_j(\theta)\right) \cdot e^j$ is computed at the value of the decision vector at time θ and is used at time t. The delay $t - \theta \leq \tau$ is sum of two delays, namely, the computation time of $\widehat{g}^i(\theta)$ and communication delay between node i and the central node. During this period, the estimate vector may be updated by other nodes. In fact, we assume that t keeps track of the number of updates to the shared vector by any node. Therefore, τ is essentially an upper bound on the number of updates by the other nodes while a certain node is computing and transmitting its update to the central node. Since in our algorithm only one random coordinate is updated by each node per iteration, the updates do not get overwritten by the other nodes too often. The pseudo-code for this algorithm is demonstrated in Algorithm 5.

Algorithm 5 DPSCD for Shared Memory Systems (at an individual node)

```
Require: $\eta$
  set $w_0 = 0$
  loop
    select $i$ uniformly at random from $[n]$
    read $w(\theta)$ from the shared memory
    calculate $g^i(\theta) \in \partial f^i(w(\theta))$
    select $j$ according to $P(j) = \frac{|g^i_j(\theta)|}{\|g^i(\theta)\|_1}$ for $j \in [d]$
    calculate $\widehat{g}^i(\theta) = \left\|g^i(\theta)\right\|_1 \operatorname{sgn}(g^i_j(\theta)) e^j$
    read w(t)
    write $w(t) = w(t) - \eta\widehat{g}(\theta)$ to the shared memory
  end loop
```

Convergence Analysis

The analysis of this method follows a similar procedure to that of HOGWILD! [8]. Theorem (2) presents the asymptotic analysis of this algorithm.

Theorem 2. *Suppose that we want to solve the optimization problem* (3.15) *when* $\{f^i\}_{i=1}^n$ *and* f_D *are* λ*-strongly convex and have* L*-Lipschitz continuous gradients. Moreover,* $\left\|\nabla f^i(w)\right\|^2 \leq M^2$*. In the algorithm, suppose that* $t - \theta \leq \tau$ *and we use constant step size* $\eta(t) = \eta$*. Then, we have*

$$\mathbb{E}\left[f_S(w(T)) - f_S(w^*)\right] \leq (1 - \eta\Lambda)^T L \left\|w_0 - w^*\right\|^2 + \frac{\eta L K M^2}{\lambda}, \tag{3.17}$$

where $\Lambda = \frac{\sqrt{1+2\sqrt{d}}-1}{\sqrt{1+2\sqrt{d}}+1}\lambda \leq \frac{1}{\eta}$ *and* $K = \left(\tau + \sqrt{d + 6\tau + 2\tau^2\sqrt{d}}\right)^2$.

Proof. The proof of this theorem follows a similar method as that of the HOGWILD! method. First, we expand the distance between the iterate and the optimum:

$$\begin{aligned}\left\|w(t+1) - w^*\right\|^2 &= \left\|w(t) - w^*\right\|^2 + \eta^2 \left\|\widehat{g}^i(\theta)\right\|^2 - 2\eta \left(w(t) - w^*\right)^\top \widehat{g}^i(\theta) \\ &= \left\|w(t) - w^*\right\|^2 + \eta^2 \left\|\widehat{g}^i(\theta)\right\|^2 - 2\eta \left(w(t) - w(\theta)\right)^\top \widehat{g}^i(t) \\ &\quad - 2\eta \left(w(t) - w(\theta)\right)^\top \left(\widehat{g}^i(\theta) - \widehat{g}^i(t)\right) \\ &\quad - 2\eta \left(w(\theta) - w^*\right)^\top \widehat{g}^i(\theta). \end{aligned} \tag{3.18}$$

In the next step, we take expectation of both sides to find the expected error. Note that

$\mathbb{E}\left[\widehat{g}^i(t)\right] = \mathbb{E}\left[\mathbb{E}\left[\widehat{g}^i(t)|\mathcal{F}(t)\right]\right]$ and $\mathbb{E}\left[\widehat{g}^i(t)|\mathcal{F}(t)\right] = \mathbb{E}_i\mathbb{E}_j\left[\widehat{g}^i(t)|\mathcal{F}(t)\right] = \nabla f_S(w(t))$, where j represents a random coordinate of the local gradient vector $g^i(t)$ and i is a function index selected uniformly at random from $[n]$ and $\mathcal{F}(t)$ is the σ-algebra generated by the random coordinate and node choices up to time t. Therefore, taking expectation of both sides with respect to all choices of coordinates local gradient vectors and choices of functions $\{f^i\}$ at each step results in

$$\begin{aligned}
&\mathbb{E}\left[\|w(t+1)-w^*\|^2\right] - \mathbb{E}\left[\|w(t)-w^*\|^2\right] \\
&\qquad\qquad =\eta^2\,\mathbb{E}\left[\left\|\widehat{g}^i(\theta)\right\|^2\right] - 2\eta\,\mathbb{E}\left[(w(t)-w(\theta))^\top \nabla f_S\left(w(t)\right)\right] \\
&\qquad\qquad\quad - 2\eta\,\mathbb{E}\left[(w(t)-w(\theta))^\top \left(\nabla f_S\left(w(\theta)\right) - \nabla f_S\left(w(t)\right)\right)\right] \\
&\qquad\qquad\quad - 2\eta\,\mathbb{E}\left[(w(\theta)-w^*)^\top \nabla f_S\left(w(\theta)\right)\right] \\
&\qquad\qquad \le\eta^2 d\, M^2 - 2\eta\,\mathbb{E}\left[f_S\left(w(t)\right) - f_S\left(w(\theta)\right)\right] - \eta\lambda\,\mathbb{E}\left[\|w(t)-w(\theta)\|^2\right] \\
&\qquad\qquad\quad - 2\eta\,\mathbb{E}\left[(w(t)-w(\theta))^\top \left(\nabla f_S\left(w(\theta)\right) - \nabla f_S\left(w(t)\right)\right)\right] \\
&\qquad\qquad\quad - \eta\lambda\,\mathbb{E}\left[\|w(\theta)-w^*\|^2\right]. \qquad (3.19)
\end{aligned}$$

The inequality above results from λ-strong convexity of f_S and the fact that $f_S\left(w(\theta)\right) - f_S\left(w^*\right) \ge 0$. Now, for the second term in the r.h.s of (3.19) we have:

$$\begin{aligned}
-2\eta\,\mathbb{E}\left[f_S\left(w(t)\right) - f_S\left(w(\theta)\right)\right] &= \frac{2\eta}{n}\sum_{i=1}^{n}\sum_{t'=\theta}^{t-1}\mathbb{E}\left[f^i\left(w(t')\right) - f^i\left(w(t'+1)\right)\right] \\
&\le \frac{2\eta\tau}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(g^i(t)\right)^\top \left(w(t)-w(t+1)\right)\right] \\
&\le \frac{2\eta^2\tau}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(g^i(t)\right)^\top \widehat{g}^i(t)\right] \\
&= \frac{2\eta^2\tau}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|g^i(t)\right\|_1 \left(g^i(t)\right)^\top e^j\right] \\
&= \frac{2\eta^2\tau}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|g^i(t)\right\|_2^2\right] \\
&\le 2\eta^2\tau M^2. \qquad (3.20)
\end{aligned}$$

Similarly, we also get that

$$
\begin{aligned}
&- 2\eta \ \mathbb{E}\left[(w(t) - w(\theta))^\top \left(\nabla f_S\left(w(\theta)\right) - \nabla f_S\left(w(t)\right)\right)\right] \\
&\qquad = \frac{2\eta}{n} \sum_{i=1}^{n} \sum_{t'=\theta}^{t-1} \mathbb{E}\left[(w(t') - w(t'+1))^\top \left(\nabla f^i\left(w(\theta)\right) - \nabla f^i\left(w(t)\right)\right)\right] \\
&\qquad \leq 4\eta^2 \tau M^2. \qquad (3.21)
\end{aligned}
$$

Plugging the this bounds in inequality (3.19) and rearranging the terms results in the following:

$$
\begin{aligned}
&\mathbb{E}\left[\|w(t+1) - w^*\|^2\right] - \mathbb{E}\left[\|w(t) - w^*\|^2\right] \\
&\leq \eta^2 d \ M^2 + 2\eta^2 \tau M^2 + 4\eta^2 \tau M^2 \\
&\quad - \eta\lambda\left(\mathbb{E}\left[\|w(\theta) - w^*\|^2\right] + \mathbb{E}\left[\|w(t) - w(\theta)\|^2\right]\right) \\
&= \eta^2 M^2 \left(d + 6\tau\right) \\
&\quad - \eta\lambda\left(\mathbb{E}\left[\|w(t) - w^*\|^2\right] - 2\mathbb{E}\left[(w(t) - w(\theta))^\top (w(\theta) - w^*)\right]\right), \qquad (3.22)
\end{aligned}
$$

where the equality results from the fact that for any two (random) vectors u and v in $\mathcal{R}^d$, we have that $\|u\|^2 + \|v\|^2 = \|u+v\|^2 - 2u^\top v$. Therefore:

$$\begin{aligned}
&\mathbb{E}\left[\|w(t+1) - w^*\|^2\right] - (1-\eta\lambda)\,\mathbb{E}\left[\|w(t) - w^*\|^2\right] \\
&\leq \eta^2 M^2 (d + 6\tau) + 2\eta\lambda \mathbb{E}\left[(w(t) - w(\theta))^\top (w(\theta) - w^*)\right] \\
&= \eta^2 M^2 (d + 6\tau) + 2\eta\lambda \sum_{t'=\theta}^{t-1} \mathbb{E}\left[(w(t'+1) - w(t'))^\top (w(\theta) - w^*)\right] \\
&= \eta^2 M^2 (d + 6\tau) + 2\eta^2\lambda \sum_{t'=\theta}^{t-1} \mathbb{E}\left[\left(\widehat{g}^i(t')\right)^\top (w(\theta) - w^*)\right] \\
&= \eta^2 M^2 (d + 6\tau) + 2\eta^2\lambda \sum_{t'=\theta}^{t-1} \mathbb{E}\left[\mathbb{E}\left[\left(\widehat{g}^i(t')\right)^\top (w(\theta) - w^*)\,|\mathcal{F}(t')\right]\right] \\
&= \eta^2 M^2 (d + 6\tau) + 2\eta^2\lambda \sum_{t'=\theta}^{t-1} \mathbb{E}\left[\mathbb{E}\left[\widehat{g}^i(t')|\mathcal{F}(t')\right]^\top (w(\theta) - w^*)\right] \\
&= \eta^2 M^2 (d + 6\tau) + 2\eta^2\lambda \sum_{t'=\theta}^{t-1} \mathbb{E}\left[\nabla f_S(w(t'))^\top (w(\theta) - w^*)\right] \\
&\overset{(a)}{\leq} \eta^2 M^2 (d + 6\tau) + 2\eta^2\lambda \sum_{t'=\theta}^{t-1} \mathbb{E}\left[\|\nabla f_S(w(t'))\| \cdot \|w(\theta) - w^*\|\right] \\
&\leq \eta^2 M^2 (d + 6\tau) + 2\eta^2\lambda \sum_{t'=\theta}^{t-1} M\ \mathbb{E}\left[\|w(\theta) - w^*\|\right] \\
&\overset{(b)}{\leq} \eta^2 M^2 (d + 6\tau) + 2\tau M\eta^2\lambda\ \mathbb{E}\left[\|w(t) - w(\theta)\| + \|w(t) - w^*\|\right] \\
&\overset{(c)}{\leq} \eta^2 M^2 (d + 6\tau) + 2\tau M\eta^2\lambda\ \mathbb{E}\left[\tau\eta\sqrt{d}M + \|w(t) - w^*\|\right] \\
&\overset{(d)}{\leq} \eta^2 M^2 (d + 6\tau) + 2\tau M\eta^2\lambda \left(\tau\eta\sqrt{d}M + \sqrt{\mathbb{E}\left[\|w(t) - w^*\|^2\right]}\right),
\end{aligned} \tag{3.23}$$

where (a) follows from Cauchy-Schwarz inequality, (b) follows from the triangle inequality, (c) follows from the triangle inequality and the fact that $\|g^i(t')\|_1 \leq \sqrt{d}\,\|g^i(t')\|_2 \leq \sqrt{d}M$ and inequality (d) is an application of Jensen's inequality. For a more compact proof, define $y(t) = \mathbb{E}\left[\|w(t) - w^*\|^2\right]$. Then, we have:

$$\begin{aligned}
y(t+1) &\leq (1-\eta\lambda)\,y(t) + 2\tau M\eta^2\lambda\ \sqrt{y(t)} \\
&\quad + \eta^2 M^2 \left(d + 6\tau + 2\tau^2\sqrt{d}\eta\lambda\right).
\end{aligned} \tag{3.24}$$

To find the fixed point of this recursion, we study the asymptotic behavior of the algorithm. Define $\alpha = (1-\eta\lambda)$, $\beta = 2\tau M\eta^2\lambda$, and $\gamma = \eta^2 M^2 \left(d + 6\tau + 2\tau^2\sqrt{d}\eta\lambda\right)$. As $t \to \infty$, we have

$$y_\infty = \alpha\ y_\infty + \beta\ \sqrt{y_\infty} + \gamma. \tag{3.25}$$

Therefore,

$$y_\infty = \left(\frac{-\beta - \sqrt{\beta^2 - 4\left(\alpha - 1\right)\gamma}}{2\left(\alpha - 1\right)}\right)^2$$

$$= \left(\tau M\eta + \frac{\sqrt{\tau^2 M^2\eta^2\lambda^2 + \eta\lambda M^2\left(d + 6\tau + 2\tau^2\sqrt{d}\eta\lambda\right)}}{\lambda}\right)^2. \tag{3.26}$$

Now, assuming that $\eta\lambda \le 1$, we have

$$y_\infty \le \frac{M^2}{\eta\lambda}\left(\tau\eta + \frac{\sqrt{\tau^2\eta^2\lambda^2 + \eta^2\lambda^2\left(d + 6\tau + 2\tau^2\sqrt{d}\eta\lambda\right)}}{\lambda}\right)^2$$

$$\le \frac{\eta M^2}{\lambda}\left(\tau + \sqrt{\tau^2 + \left(d + 6\tau + 2\tau^2\sqrt{d}\eta\lambda\right)}\right)^2$$

$$\le \frac{\eta M^2}{\lambda}\left(\tau + \sqrt{d + 6\tau + \tau^2\left(1 + \eta\lambda\sqrt{d}\right)}\right)^2 \tag{3.27}$$

Also, from 3.26 we have that

$$y_\infty \ge \left(\tau M\eta + \frac{\sqrt{\tau^2 M^2\eta^2\lambda^2 + \eta\lambda M^2\left(2\tau^2\sqrt{d}\eta\lambda\right)}}{\lambda}\right)^2 = \tau^2 M^2\eta^2\left(1 + \sqrt{1 + 2\sqrt{d}}\right)^2. \tag{3.28}$$

Taking into account that $y_\infty \le y(t)$ for any t, from (3.24) we have

$$y(t+1) \le \alpha\ y(t) + \beta\ \frac{y(t)}{\sqrt{y_\infty}} + \gamma$$

$$= \alpha\ \left(y(t) - y_\infty\right) + \frac{\beta}{\sqrt{y_\infty}}\left(y(t) - y_\infty\right) + \alpha y_\infty + \beta\sqrt{y_\infty} + \gamma$$

$$= \left(\alpha + \frac{\beta}{\sqrt{y_\infty}}\right)\left(y(t) - y_\infty\right) + y_\infty \tag{3.29}$$

or

$$y(t+1) - y_\infty \le \left(\alpha + \frac{\beta}{\sqrt{y_\infty}}\right)\left(y(t) - y_\infty\right). \tag{3.30}$$

Expanding the recursion and plugging in the results of (3.27) and (3.28) gives us the following closed-form formula:

$$\begin{aligned}
y(t) &\leq \left(\alpha + \frac{\beta}{\sqrt{y_\infty}}\right)^T (y_0 - y_\infty) + y_\infty \\
&= \left(1 - \eta\lambda + \frac{2\tau M\eta^2\lambda}{\sqrt{y_\infty}}\right)^T (y_0 - y_\infty) + y_\infty \\
&\overset{(a)}{\leq} \left(1 - \eta\lambda + \frac{2\tau M\eta}{\sqrt{y_\infty}}\eta\lambda\right)^T (y_0 - y_\infty) + \frac{KM^2}{\lambda}\eta \\
&\overset{(b)}{\leq} \left(1 - \left(1 - \frac{2\tau M\eta}{\tau M\eta\left(1 + \sqrt{1 + 2\sqrt{d}}\right)}\right)\eta\lambda\right)^T (y_0 - y_\infty) + \frac{KM^2}{\lambda}\eta \\
&\leq \left(1 - \left(1 - \frac{2}{1 + \sqrt{1 + 2\sqrt{d}}}\right)\eta\lambda\right)^T y_0 + \frac{KM^2}{\lambda}\eta \\
&= \left(1 - \frac{\sqrt{1 + 2\sqrt{d}} - 1}{\sqrt{1 + 2\sqrt{d}} + 1}\eta\lambda\right)^T y_0 + \frac{KM^2}{\lambda}\eta, \qquad (3.31)
\end{aligned}$$

where $K = \left(\tau + \sqrt{d + 6\tau + 2\tau^2\sqrt{d}}\right)^2$ and (a) is a result of (3.27) and (b) is a result of (3.28). Hence,

$$\begin{aligned}
&\mathbb{E}\left[\|w(t+1) - w^*\|^2\right] \\
&\leq \left(1 - \frac{\sqrt{1 + 2\sqrt{d}} - 1}{\sqrt{1 + 2\sqrt{d}} + 1}\eta\lambda\right)^T \|w_0 - w^*\|^2 + \frac{KM^2}{\lambda}\eta. \qquad (3.32)
\end{aligned}$$

Due to the Lipschitz continuity of the gradients (1.5) and considering that $\nabla f_S(w^*) = 0$, we have

$$\mathbb{E}\left[f_S(w(t)) - f_S(w^*)\right] \leq \frac{L}{2}\ \|w(t) - w^*\|^2. \qquad (3.33)$$

Therefore,

$$\mathbb{E}\left[f_S(w(T)) - f_S(w^*)\right] \leq \frac{L}{2}\left(1 - \frac{\sqrt{1 + 2\sqrt{d}} - 1}{\sqrt{1 + 2\sqrt{d}} + 1}\eta\lambda\right)^T \|w_0 - w^*\|^2 + \frac{LKM^2}{\lambda}\eta. \qquad (3.34)$$

□

3.2.2 DPSCD for Complete networks

Consider the setup where all nodes are directly connected to each other and cooperate to solve the following problem:

$$\min_{w \in \mathcal{R}^d} f_D(w) = \frac{1}{n} \sum_{i=1}^{n} f^i(w), \tag{3.35}$$

where $\{f^i(w)\}_{i=1}^n$ are convex functions with Lipschitz continuous gradients. If all nodes are connected to each other, every node has the gradient information of all nodes. However, since the nodes are not sharing their estimate values $\{w^i(t)\}$, the gradient information is not useful unless all the nodes have the same iterate value at every iteration. In order to satisfy this requirement without synchronizing the iterate values at every iteration, nodes can start the optimization algorithm with the same initial value. The update rule in this model is

$$w^i(t+1) = w^i(t) - \eta \sum_{k=1}^{n} \frac{\widehat{g}^k(t)}{n}, \tag{3.36}$$

where $\eta(t) = \frac{1}{\lambda t}$ and $\widehat{g}^k(t) = \left\|g^k(t)\right\|_1 \cdot \operatorname{sgn}\left(g^k_{J^k_t}\right) \cdot e^{J^k(t)} = \gamma^k(t) e^{J^k_t}$. In the end, the algorithm outputs the time average $\widetilde{w^i}(T) = \frac{1}{T} \sum\limits_{t=1}^{T} w^i(t)$.

Algorithm 6 DPSCD for Complete Networks

```
Require: $n$, $T$, $\lambda$
  set $w_0 = 0$
  for $t = 1, 2, \ldots, T$ do
    for all $i \in [n]$ do
      calculate $g^i(t) \in \partial f^i(w(t))$
      select $J^i_t$ according to $P(j) = \frac{|g^i_j(t)|}{\|g^i(t)\|_1}$ from $j \in [d]$
      calculate $\widehat{g}^i(t) = \left\|g^i(t)\right\|_1 \operatorname{sgn}(g^i_{J^i_t}) e^j$
      send $\widehat{g}^i(t)$ to all other nodes in the network
    end for
    for all $i \in [n]$ do
      set $\eta_t = \frac{1}{\lambda t}$
      set $w^i(t+1) = w^i(t) - \eta(t) \sum\limits_{k=1}^{n} \frac{\widehat{g}^k(t)}{n}$
    end for
  end for
  return $W(T+1)$
```

Convergence Analysis

If all nodes start from the same initial point, the problem basically reduces to the central case with the slight difference that the objective function is in the form of $f(w) = \frac{1}{n}\sum_{i=1}^{n} f^i(w)$

and we estimate the gradient by $\widehat{g}(t) = \sum_{i=1}^{n} \frac{\widehat{g}^i(t)}{n}$, where $\widehat{g}^i(t)$ is an unbiased estimate of $g^i(t) = \nabla f^i(w(t))$ computed using PSCD.

Theorem 3. *Consider Algorithm (3.35) for solving problem (1.11) where every f^i is convex with L-Lipschitz continuous gradients. If every node uses step size $\eta = \frac{1}{\alpha L}$, where $L = \max_i L^i$, we have*

$$\mathbb{E}\left[f(w(T)) - f(w^*)\right] \leq \frac{\alpha\left(\Psi(w(0) - \Psi(w^*)\right)}{T}, \tag{3.37}$$

where

$$\Psi(w) = f(w) + \frac{L}{2}\|w - w^*\|^2. \tag{3.38}$$

Proof. Here, we assume that all nodes start with the same starting point $w(0)$. Since they all use the same gradient estimate $\widehat{g}_t$ at each step, the iterates w^i are the same among all nodes at each iteration. We denote the common iterate by $w(t) = w^i(t)$ for all $i \in [n]$. Following the proof for centralized PSCD, if we substitute $\widehat{g}(t) = \sum_{i=1}^{n} \frac{\widehat{g}^i(t)}{n}$, we get

$$\begin{aligned}
\Psi(w(t)) - \Psi(w(t+1)) =& f(w(t)) - f(w(t+1)) \\
&+ \frac{L}{2}(\|w(t) - w^*\|^2 - \|w(t+1) - w^*\|^2) \\
\geq& - g(t)^\top \left(w(t+1) - w(t)\right) - \frac{L}{2}\left\|\eta\ \widehat{g}(t)\right\|^2 \\
&+ \frac{L}{2}\left(w(t) - w(t+1)\right)^\top \left(w(t) + w(t+1) - 2w^*\right) \\
=& - \eta\ g(t)^\top \widehat{g}(t) - \frac{L}{2}\left\|\eta\ \widehat{g}(t)\right\|^2 \\
&+ \frac{L}{2}\eta\ \widehat{g}(t)^\top \left(2w(t) - 2w^* - \eta\ \widehat{g}(t)\right) \\
=& \frac{1}{n}\sum_{k=1}^{n}\gamma^k(t) g_{J_t^k} - \frac{L}{2}\left\|\frac{1}{n}\sum_{k=1}^{n}\left(\gamma^k(t)e^{J_t^k}\right)\right\|^2 \\
&+ \frac{1}{n}\sum_{k=1}^{n}\frac{L}{2}\gamma^k(t)(2w_{J^k(t)} - 2w^*_{J_t^k}) - \frac{L}{2}\left\|\frac{1}{n}\sum_{k=1}^{n}\left(\gamma^k(t)e^{J_t^k}\right)\right\|^2 \\
=& \frac{1}{n}\sum_{k=1}^{n}\gamma^k(t) g_{J_t^k} - L\left\|\frac{1}{n}\sum_{k=1}^{n}\left(\gamma^k(t)e^{J_t^k}\right)\right\|^2 \\
&+ L\frac{1}{n}\sum_{k=1}^{n}\gamma^k(t)(w_{J^k(t)} - w^*_{J_t^k}).
\end{aligned} \tag{3.39}$$

Then, taking conditional expectation of both sides w.r.t to the entire history up to time t results in

$$\begin{aligned}
\mathbb{E}\left[\Psi(w(t)) - \Psi(w(t+1))|\mathcal{F}(t)\right] \geq & \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\gamma^k(t) g_{J_t^k}|\mathcal{F}(t)\right] - L\left\|\frac{1}{n}\sum_{k=1}^{n}\left(\gamma^k(t)e^{J_t^k}\right)\right\|^2 \\
& + L\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\gamma^k(t)\left(w_{J^k(t)} - w^*_{J_t^k}\right)|\mathcal{F}(t)\right] \\
= & \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\eta\left\|g^k(t)\right\|_1 \cdot \operatorname{sgn}\left(g^k_{J_t^k}\right)\cdot g_{J_t^k}|\mathcal{F}(t)\right] \\
& - L\left\|\frac{1}{n}\sum_{k=1}^{n}\left(\gamma^k(t)e^{J_t^k}\right)\right\|^2 \\
& + L\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\eta\left\|g^k(t)\right\|_1 \cdot \operatorname{sgn}\left(g^k_{J_t^k}\right)\cdot\left(w_{J^k(t)} - w^*_{J_t^k}\right)|\mathcal{F}(t)\right] \\
= & \frac{1}{n}\sum_{k=1}^{n}\sum_{J_t^k=1}^{d}\frac{\eta\left\|g^k(t)\right\|_1 \cdot \operatorname{sgn}\left(g^k_{J_t^k}\right).g^k_{J_t^k}\cdot g_{J_t^k}}{\left\|g^k(t)\right\|_1} \\
& - L\left\|\frac{1}{n}\sum_{k=1}^{n}\left(\gamma^k(t)e^{J_t^k}\right)\right\|^2 \\
& + L\frac{1}{n}\sum_{k=1}^{n}\sum_{J_t^k=1}^{d}\frac{\eta\left\|g^k(t)\right\|_1 \cdot \operatorname{sgn}\left(g^k_{J_t^k}\right).g^k_{J_t^k}\cdot\left(w_{J^k(t)} - w^*_{J_t^k}\right)}{\left\|g^k(t)\right\|_1} \\
= & \frac{1}{\alpha L n}\sum_{k=1}^{n} g^k(t)^\top g(t) \\
& - L\left\|\frac{1}{n}\sum_{k=1}^{n}\left(\eta\left\|g^k(t)\right\|_1 \cdot \operatorname{sgn}\left(g^k_{J_t^k}\right)\cdot e^{J_t^k}\right)\right\|^2 \\
& + \frac{1}{\alpha n}\sum_{k=1}^{n} g^k(t)^\top\left(w(t) - w^*\right) \\
= & \frac{1}{\alpha L} g(t)^\top g(t) - \frac{1}{\alpha^2 n^2 L}\left\|\sum_{k=1}^{n}\left(\left\|g^k(t)\right\|_1 \cdot \operatorname{sgn}\left(g^k_{J_t^k}\right)\cdot e^{J_t^k}\right)\right\|^2 \\
& + \frac{1}{\alpha} g(t)\top\left(w(t) - w^*\right) \\
\geq & \frac{\|g(t)\|_2^2}{\alpha L} - \frac{1}{\alpha^2 n^2 L}\left|\sum_{k=1}^{n}\left\|g^k(t)\right\|_1\right|^2 + \frac{1}{\alpha} g(t)\top\left(w(t) - w^*\right) \\
= & \frac{\|g(t)\|_2^2}{\alpha L} - \frac{1}{\alpha^2 n^2 L}\left|n\cdot\|g(t)\|_1\right|^2 + \frac{1}{\alpha} g(t)\top\left(w(t) - w^*\right) \\
= & \frac{\|g(t)\|_2^2}{\alpha L} - \frac{\|g(t)\|_1^2}{\alpha^2 L} + \frac{1}{\alpha} g(t)\top\left(w(t) - w^*\right)
\end{aligned} \tag{3.40}$$

The first inequality above follows from L-smoothness of the objective function and the second inequality results from repeatedly applying the triangle inequality. Therefore, if $\alpha \geq \max_i \frac{\|g^i(t)\|_1^2}{\|g^i(t)\|_2^2}$

we will have the following inequality:

$$\mathbb{E}\left[\Psi(w(t)) - \Psi(w(t+1))|\mathcal{F}(t)\right] \geq \frac{1}{\alpha} g(t)^\top (w(t) - w^*), \tag{3.41}$$

Again, since we have $\frac{\|u\|_1^2}{\|u\|_2^2} \leq d$ for any vector u, it suffices to set $\alpha \geq d$. Following the same arguments as in the centralized method, we get

$$\mathbb{E}\left[f(w(T)) - f(w^*)\right] \leq \frac{\alpha\left(\Psi(w(0) - \Psi(w*)\right)}{T}. \tag{3.42}$$

$\square$

3.3 Discussion

In this chapter the proportional stochastic coordinate descent algorithm was discussed for three different setups: centralized, shared memory system, and complete network. This method requires computation of the entire gradient vector at each step which seems wasteful since optimization algorithms have faster convergence if they use the true gradient vector instead of an estimate of it. However, in distributed setups, computing the true gradient of each local function is cheaper, so for this setups this method seems more promising. Also, under some conditions, such as smoothness, etc., the algorithm might still achieve comparable convergence rates with less frequent computation of the true gradient. A future direction of this work can be the study of these conditions and the relationship between the frequency of the true gradient computations and the convergence rate.

Moreover, for the complete network setup we observe that DPSCD algorithm has the same convergence rate as its centralized counterpart. Because in this model the gradient computation task is distributed among n nodes, we can achieve linear speedup (in n) in runtime (measured in number of CPU ycles) compared to the centralized setup. For the shared memory system, however, the speedup depends on the relation between the delay τ and number of processors p.

Chapter 4
A Semi-Stochastic Method

4.1 Stochastic Variance Reduced Coordinate Descent

Although stochastic optimization methods seem attractive due to their small per-iteration complexity, they show high variance in performance which arises from the high variance in the estimates of the gradient vectors. In order to address this issue we propose that one can make use of the previous gradient estimates. The idea is that if the objective function is smooth enough so that gradient values do not change drastically between two nearby iterates, the previous gradient estimate is still a good estimate of the current gradient vector. In this section, we consider the centralized minimization problem

$$\min_{w \in \mathcal{R}^d} f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w), \tag{4.1}$$

where $f : \mathbb{R}^d \to \mathcal{R}$ is a λ-strongly convex objective function with L-Lipschitz continuous gradients. For this family of problems, first-order methods such as gradient descent (GD) achieve linear convergence rates (if adopting a fixed step-size). however, their fast convergence is only practical for problems with moderate size because their per-iteration cost increases linearly with the size of the data set [21]. In the recent years, with the rise of large scale problems, the focus has been on stochastic methods that use some inexpensive approximation of the gradient instead of taking a full gradient at each iteration in order to reduce the complexity. However, these methods achieve slower convergence rates and show much higher variance in performance compared to their more complex deterministic counterparts. In order to fill in this gap, Le Roux, et al. in [9] have suggested the stochastic average gradient (SAG) method that achieves linear convergence rate of GD while enjoying little iteration cost of stochastic gradient descent. This algorithm, at each iteration, computes the value of the subgradients at only a random subset of data points and updates those values, while the subgradient values for other data points are the same as those at time $t-1$. The major drawback of this algorithm is that it requires a large amount of memory to keep the values of subgradients at all data points.

The coordinate-wise method we present here addresses the memory requirement issue of SAG while maintaining both fast convergence rate and little per-iteration complexity. This method updates a random coordinate of the gradient vector (as opposed to the subgradient at a data point) at every iteration while keeping the old values for the other coordinates. We call our method *Stochastic Variance Reduced Coordinate descent* (SVRC). In SVRC, the memory required to save the gradient vector is proportional to the dimension of the data d (and not the size the data set nd, which is the case in SAG). Also, as we will see, the convergence rate is improved compared to SAG. However, this is achieved at the cost of improvement in the run-time as in SAG, the complexity per-iteration is improved by a factor of n, while the improvement in SVRC is relative to d.

In this section, we define $g_t = \nabla f(w_t)$ the true gradient of function f at iteration t and $\widehat{g}_t$ the approximate of the gradient vector at time t. Further, let $\theta_t = \begin{pmatrix} \widehat{g}_t \\ w_t \end{pmatrix}$ be the vertical concatenation of the approximate gradient vector and the iterate (decision vector) at time t and $\theta^* = \begin{pmatrix} g^* \\ w^* \end{pmatrix} = \begin{pmatrix} 0 \\ w^* \end{pmatrix}$ where w^* is the minimizer of $f(w)$ and g^* is the gradient vector at minimum which equals the zero vector since $f(w)$ is convex.

The Stochastic Variance Reduced Coordinate descent (SVRC) method we present here updates a random coordinate of the gradient vector at every iteration while keeping the old values for the other coordinates. Thus, the update rule is the following:

$$w_{t+1} = w_t - \eta \widehat{g}_t, \tag{4.2}$$

where η is the fixed step-size which depends on the dimension of the data and the Lipschitz constant. The approximate gradient vector $\widehat{g}_t$ is computed at each iteration according to the following rule:

$$\widehat{g}_t = \sum_{j=1}^{d} \widehat{g}_j(t) e^j, \tag{4.3}$$

where e^j is the jth standard coordinate vector and

$$\widehat{g}_j(t) = \begin{cases} \nabla_j f(w_t) & \text{if } \ j \text{ is selected,} \\ \widehat{g}_j(t-1) & \text{otherwise.} \end{cases} \tag{4.4}$$

is the jth coordinate of $\widehat{g}_t$ and j is the randomly selected coordinate to be updated with probability $p = \frac{1}{d}$.

In order to write equation (4.4) in a more compact form, we introduce a random vector z_t. The elements of this vector are zero-mean random variables z_{i_t} which take the value $\left(1 - \frac{1}{d}\right)$ with probability $\frac{1}{d}$ and $-\frac{1}{d}$ with probability $\left(1 - \frac{1}{d}\right)$. Now, the equation (4.4) can be re-written in the following compact form:

$$\widehat{g}_j(t) = \left(1 - \frac{1}{d}\right) \widehat{g}_j(t-1) + \left(1 - \frac{1}{d}\right) g_j(t-1) + z_t \left(g_j(t-1) - \widehat{g}_j(t-1)\right), \tag{4.5}$$

or

$$\widehat{g}_t = \left(1 - \frac{1}{d}\right) \widehat{g}_{t-1} + \left(1 - \frac{1}{d}\right) g_{t-1} + Z_t \left(g_{t-1} - \widehat{g}_{t-1}\right), \tag{4.6}$$

with $Z_t = \operatorname{diag}(z_t)$.

4.2 Convergence Analysis

The following theorem provides the convergence result for SVRC method proposed in this section.

Theorem 4. *Consider using SVRC for solving optimization problem 4.1 with L-smooth λ-strongly convex objective function. With constant step-size $\eta = \frac{1}{2dL}$ we have*

$$\mathbb{E}\left[f(w_T) - f(w^*)\right] < L \, \|w^*\|^2 \left(1 - \frac{\lambda}{8dL}\right)^T. \tag{4.7}$$

Proof. In order to find the convergence rate of the Stochastic Variance Reduced Coordinate descent (SVRC), we first define a quadratic function $Q(\theta_t) = \left(\theta_t - \theta^*\right)^T P \left(\theta_t - \theta^*\right)$ for some value of P to be specified later and prove the convergence of this function to zero. Then, we show that Q is a Lyapunov function for $\|w_t - w^*\|^2$. In other words, it always upper-bounds $\|w_t - w^*\|^2$.

Lemma 2. *Assume that $Q(\theta_t) = \left(\theta_t - \theta^*\right)^T P \left(\theta_t - \theta^*\right)$ with $\theta_t = \begin{pmatrix} \widehat{g}_t \\ w_t \end{pmatrix}$ and $\theta^* = \begin{pmatrix} g^* \\ w^* \end{pmatrix} = \begin{pmatrix} 0 \\ w^* \end{pmatrix}$ and $P = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ with A, B and C being symmetric. Note that $w_t, \widehat{g}_t, z_t \in \mathbb{R}^d$ and*

$A, B, C \in \mathbb{R}^{d\times d}$ and . Then, we have

$$\begin{aligned}\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] =& \widehat{g}_{t-1}^T\left[\frac{1}{d}\operatorname{Diag}(S)+\left(1-\frac{2}{d}\right)S\right]\widehat{g}_{t-1}+\frac{1}{d}g_{t-1}^T\left[\operatorname{Diag}(S)\right]g_{t-1}\\ &+\frac{2}{d}\widehat{g}_{t-1}^T\left[S-\operatorname{Diag}(S)\right]g_{t-1}+2\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T\left[B-\eta C\right](w_{t-1}-w^*)\\ &+\frac{2}{d}g_{t-1}^T\left[B-\eta C\right](w_{t-1}-w^*)+(w_{t-1}-w^*)^T C\,(w_{t-1}-w^*)\,, \end{aligned} \tag{4.8}$$

where $S = A - 2\eta B + \eta^2 C$.

Proof.

$$\begin{aligned}\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[(\theta_t-\theta^*)^T\begin{pmatrix}A & B\\ B^T & C\end{pmatrix}(\theta_t-\theta^*)|\mathcal{F}_{t-1}\right]\\ &= \mathbb{E}\left[\widehat{g}_t^T A\widehat{g}_t+2\widehat{g}_t^T B\,(w_t-w^*)+(w_t-w^*)^T C\,(w_t-w^*)\,|\mathcal{F}_{t-1}\right]. \end{aligned} \tag{4.9}$$

Plugging in the values of w_t and $\widehat{g}_t$ from (4.2) and (4.6) gives us

$$\begin{aligned}\widehat{g}_t^T A\widehat{g}_t =& \left(1-\frac{1}{d}\right)^2\widehat{g}_{t-1}^T A\widehat{g}_{t-1}+\frac{1}{d^2}g_{t-1}^T A g_{t-1}\\ &+\frac{1}{d}\left(g_{t-1}-\widehat{g}_{t-1}\right)^T\left[\operatorname{Diag}(A)-\frac{1}{d}A\right](g_{t-1}-\widehat{g}_{t-1})\\ &+\frac{2}{d}\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T g_{t-1}, \end{aligned} \tag{4.10}$$

$$\begin{aligned}\widehat{g}_t^T B\,(w_t-w^*) =& \left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T B\,(w_{t-1}-w^*)+\frac{1}{d}g_{t-1}^T B\,(w_{t-1}-w^*)\\ &-\eta\left(1-\frac{1}{d}\right)^2\widehat{g}_{t-1}^T A\widehat{g}_{t-1}-\frac{2\eta}{d}\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T B g_{t-1}\\ &-\frac{\eta}{d^2}g_{t-1}^T B g_{t-1}\\ &-\frac{\eta}{d}\left(g_{t-1}-\widehat{g}_{t-1}\right)^T\left[\operatorname{Diag}(B)-\frac{1}{d}B\right](g_{t-1}-\widehat{g}_{t-1}) \end{aligned} \tag{4.11}$$

and

$$\begin{aligned}
\left(w_t - w^*\right)^T C \left(w_t - w^*\right) = &\left(w_{t-1} - w^*\right)^T C \left(w_{t-1} - w^*\right) + \eta^2 \left(1 - \frac{1}{d}\right)^2 \widehat{g}_{t-1}^T C \widehat{g}_{t-1} \\
&+ \frac{\eta^2}{d^2} g_{t-1}^T C g_{t-1} - 2\eta \left(1 - \frac{1}{d}\right) \widehat{g}_{t-1}^T C \left(w_{t-1} - w^*\right) \\
&- \frac{2\eta}{d} g_{t-1}^T C \left(w_{t-1} - w^*\right) + \frac{2\eta^2}{d} \left(1 - \frac{1}{d}\right) \widehat{g}_{t-1}^T C g_{t-1} \\
&+ \frac{\eta^2}{d} \left(g_{t-1} - \widehat{g}_{t-1}\right)^T \left[\operatorname{Diag}(C)) - \frac{1}{d} C\right] \left(g_{t-1} - \widehat{g}_{t-1}\right). \qquad (4.12)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[Q(\theta_t) | \mathcal{F}_{t-1}\right] = &\widehat{g}_{t-1}^T \left[\left(1 - \frac{1}{d}\right)^2 A - 2\eta \left(1 - \frac{1}{d}\right)^2 B + \eta^2 \left(1 - \frac{1}{d}\right)^2 C\right] \widehat{g}_{t-1} \\
&+ g_{t-1}^T \left[\frac{1}{d^2} A - \frac{2\eta}{d^2} B + \frac{\eta^2}{d^2} C\right] g_{t-1} \\
&+ \left(g_{t-1} - \widehat{g}_{t-1}\right)^T U \left(g_{t-1} - \widehat{g}_{t-1}\right) \\
&+ \widehat{g}_{t-1}^T \left[\frac{2}{d}\left(1 - \frac{1}{d}\right) A - \frac{4\eta}{d}\left(1 - \frac{1}{d}\right) B + \frac{2\eta^2}{d}\left(1 - \frac{1}{d}\right) C\right] g_{t-1} \\
&+ \widehat{g}_{t-1}^T \left[2\left(1 - \frac{1}{d}\right) B - 2\eta \left(1 - \frac{1}{d}\right) C\right] \left(w_{t-1} - w^*\right) \\
&+ g^T \left[\frac{2}{d} B - \frac{2\eta}{d} C\right] \left(w_{t-1} - w^*\right) \\
&+ \left(w_{t-1} - w^*\right)^T C \left(w_{t-1} - w^*\right). \qquad (4.13)
\end{aligned}$$

where $U = \left[\frac{1}{d}\operatorname{Diag}(A) - \frac{1}{d^2} A - \frac{2\eta}{d}\operatorname{Diag}(B) + \frac{2\eta}{d^2} B + \frac{\eta^2}{d}\operatorname{Diag}(C) - \frac{\eta^2}{d^2} C\right]$.

If we rearrange the terms, we will have

$$\begin{aligned}
\mathbb{E}\left[Q(\theta_t) | \mathcal{F}_{t-1}\right] = &\left(1 - \frac{1}{d}\right)^2 \widehat{g}_{t-1}^T S \widehat{g}_{t-1} + \frac{1}{d^2} g_{t-1}^T S g_{t-1} \\
&+ \frac{1}{d} \left(g_{t-1} - \widehat{g}_{t-1}\right)^T \left[\operatorname{Diag}(S) - \frac{1}{d} S\right] \left(g_{t-1} - \widehat{g}_{t-1}\right) \\
&+ \frac{2}{d}\left(1 - \frac{1}{d}\right) \widehat{g}_{t-1}^T S g_{t-1} + 2\left(1 - \frac{1}{d}\right) \widehat{g}_{t-1}^T \left[B - \eta C\right] \left(w_{t-1} - w^*\right) \\
&+ \frac{2}{d} g_{t-1}^T \left[B - \eta C\right] \left(w_{t-1} - w^*\right) + \left(w_{t-1} - w^*\right)^T C \left(w_{t-1} - w^*\right). \qquad (4.14)
\end{aligned}$$

Finally, by expanding the third term above we have

$$\begin{aligned}\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] =& \widehat{g}_{t-1}^T\left[\frac{1}{d}\,\mathrm{Diag}(S)+\left(1-\frac{2}{d}S\right)\right]\widehat{g}_{t-1}+g_{t-1}^T\left[\frac{1}{d}\,\mathrm{Diag}(S)\right]g_{t-1}\\ &+\frac{2}{d}\widehat{g}_{t-1}^T\left[S-\mathrm{Diag}(S)\right]g_{t-1}+2\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T\left[B-\eta C\right]\left(w_{t-1}-w^*\right)\\ &+\frac{2}{d}g_{t-1}^T\left[B-\eta C\right]\left(w_{t-1}-w^*\right)+\left(w_{t-1}-w^*\right)^T C\left(w_{t-1}-w^*\right). \qquad (4.15)\end{aligned}$$

□

Now, we use Lemma 2 with $A=\left[3d\eta^2+\frac{\eta^2}{d}-2\eta^2\right]I$, $B=-\eta\left(1-\frac{1}{d}\right)I$, and $C=\frac{1}{d}I$ for the case of $\eta=\frac{1}{2dL}$. We note that in this case

$$B-\eta C=-\eta I$$

and

$$S=\left[3d\eta^2+\frac{\eta^2}{d}-2\eta^2\right]I+2\eta^2 I-\left(\frac{2\eta^2}{d}\right)I+\left(\frac{\eta^2}{d}\right)I=3d\eta^2 I.$$

Therefore, we have

$$\begin{aligned}\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right]=&\left(1-\frac{1}{d}\right)3d\eta^2\widehat{g}_{t-1}^T\widehat{g}_{t-1}+3\eta^2 g_{t-1}^T g_{t-1}-2\eta\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T\left(w_{t-1}-w^*\right)\\ &-\frac{2\eta}{d}g_{t-1}^T\left(w_{t-1}-w^*\right)+\frac{1}{d}\left(w_{t-1}-w^*\right)^T\left(w_{t-1}-w^*\right). \qquad (4.16)\end{aligned}$$

Nesterov in [21] proves the following lemma that we will use in the sequel.

Lemma 3. *Let $h:\mathcal{R}^d\to\mathcal{R}$ be a function with L-Lipschitz continuous gradients. Then, for any vectors u and v we have*

$$\|\nabla h(u)-\nabla h(v)\|^2\le L\left\langle\nabla h(u)-\nabla h(v),u-v\right\rangle. \qquad (4.17)$$

Applying Lemma 3 to the second term in equation (4.16) results in

$$3\eta^2 g_{t-1}^T g_{t-1}\le 3\eta^2 L g_{t-1}^T\left(w_{t-1}-w^*\right). \qquad (4.18)$$

Thus, we get

$$\begin{aligned}\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] \leq &\left(1-\frac{1}{d}\right)3d\eta^2\widehat{g}_{t-1}^T\widehat{g}_{t-1} - 2\eta\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T\left(w_{t-1}-w^*\right) \\ &+\left(3\eta^2L-\frac{2\eta}{d}\right)g_{t-1}^T\left(w_{t-1}-w^*\right)+\frac{1}{d}\left(w_{t-1}-w^*\right)^T\left(w_{t-1}-w^*\right). \end{aligned} \tag{4.19}$$

On the other hand, from the definition of $Q(\theta)$ we have

$$\begin{aligned}Q(\theta_{t-1}) =&\widehat{g}_{t-1}^T A\widehat{g}_{t-1} + 2\widehat{g}_{t-1}^T B\left(w_{t-1}-w^*\right) + \left(w_{t-1}-w^*\right)^T C\left(w_{t-1}-w^*\right) \\ =&\widehat{g}_{t-1}^T\left[3d\eta^2+\frac{\eta^2}{d}-2\eta^2\right]I\widehat{g}_{t-1} \\ &-2\widehat{g}_{t-1}^T\left[\eta\left(1-\frac{1}{d}\right)I\right]\left(w_{t-1}-w^*\right) \\ &+\frac{1}{d}\left(w_{t-1}-w^*\right)^T\left(w_{t-1}-w^*\right). \end{aligned} \tag{4.20}$$

Therefore,

$$\begin{aligned}&\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right]-\left(1-\delta\right)Q(\theta_{t-1}) \\ &\qquad\leq\left[3d\eta^2-3\eta^2-3d\eta^2-\frac{\eta^2}{d}+2\eta^2+3d\delta\eta^2+\frac{\eta^2\delta}{d}-2\delta\eta^2\right]\widehat{g}_{t-1}^T\widehat{g}_{t-1} \\ &\qquad-2\eta\delta\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T\left(w_{t-1}-w^*\right) \\ &\qquad+\left(3\eta^2L-\frac{2\eta}{d}\right)g_{t-1}^T\left(w_{t-1}-w^*\right)+\frac{\delta}{d}\left(w_{t-1}-w^*\right)^T\left(w_{t-1}-w^*\right). \end{aligned} \tag{4.21}$$

For some value of δ that we specify later Rearranging the terms inside the first bracket on the right-hand side results in

$$\begin{aligned}\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right]-\left(1-\delta\right)Q(\theta_{t-1}) \leq &\left[3d\eta^2\left(\delta-\frac{1}{d}\right)+(1-\delta)\eta^2\left(2-\frac{1}{d}\right)\right]\widehat{g}_{t-1}^T\widehat{g}_{t-1} \\ &-2\eta\delta\left(1-\frac{1}{d}\right)\widehat{g}_{t-1}^T\left(w_{t-1}-w^*\right) \\ &+\left(3\eta^2L-\frac{2\eta}{d}\right)g_{t-1}^T\left(w_{t-1}-w^*\right) \\ &+\frac{\delta}{d}\left(w_{t-1}-w^*\right)^T\left(w_{t-1}-w^*\right). \end{aligned} \tag{4.22}$$

Let $M=\left[3d\eta^2\left(\delta-\frac{1}{d}\right)+(1-\delta)\eta^2\left(2-\frac{1}{d}\right)\right]I=cI$. We know that if c is negative, for any vector $v \in \mathbb{R}^d$, $v^T M v < 0$. Therefore:

$$\begin{aligned}
&\left(\widehat{g}_{t-1} + \frac{1}{2}M^{-1}t\right)^T M \left(\widehat{g}_{t-1} + \frac{1}{2}M^{-1}t\right) < 0 \\
\Rightarrow \quad & \widehat{g}_{t-1}^T M \widehat{g}_{t-1} + \widehat{g}_{t-1}^T t + \frac{1}{4} t^T M^{-1} t < 0 \\
\Rightarrow \quad & \widehat{g}_{t-1}^T M \widehat{g}_{t-1} + \widehat{g}_{t-1}^T t < -\frac{1}{4} t^T M^{-1} t,
\end{aligned} \tag{4.23}$$

where $t = -2\eta\delta\left(1 - \frac{1}{d}\right)(w_{t-1} - w^*)$. This gives an upper bound on the first two terms in (4.22). Therefore, we have

$$\begin{aligned}
\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] - (1-\delta)\,Q(\theta_{t-1}) \leq & -\frac{\frac{1}{4}\left(2\eta\delta\left(1-\frac{1}{d}\right)\right)^2}{3d\eta^2\left(\delta - \frac{1}{d}\right) + (1-\delta)\eta^2\left(2 - \frac{1}{d}\right)}\|w_{t-1} - w^*\|^2 \\
& + \left(3\eta^2 L - \frac{2\eta}{d}\right) g_{t-1}^T\left(w_{t-1} - w^*\right) \\
& + \frac{\delta}{d}\left(w_{t-1} - w^*\right)^T\left(w_{t-1} - w^*\right) \\
= & \frac{-\delta^2\left(1 - \frac{1}{d}\right)^2}{3d\delta - 1 + \frac{\delta - 1}{d} - 2\delta}\|w_{t-1} - w^*\|^2 \\
& + \left(3\eta^2 L - \frac{2\eta}{d}\right) g_{t-1}^T\left(w_{t-1} - w^*\right) \\
& + \frac{\delta}{d}\left(w_{t-1} - w^*\right)^T\left(w_{t-1} - w^*\right) \\
= & \left(\frac{\delta}{d} - \frac{-\delta^2\left(1 - \frac{1}{d}\right)^2}{3d\delta - 1 + \frac{\delta - 1}{d} - 2\delta}\right)\|w_{t-1} - w^*\|^2 \\
& + \left(3\eta^2 L - \frac{2\eta}{d}\right) g_{t-1}^T\left(w_{t-1} - w^*\right)
\end{aligned} \tag{4.24}$$

In order for M to be negative definite, we need the following condition:

$$\begin{aligned}
&\left[3d\eta^2\left(\delta - \frac{1}{d}\right) + (1-\delta)\eta^2\left(2 - \frac{1}{d}\right)\right] I \prec 0 \\
\Rightarrow \quad & 3d\eta^2\left(\delta - \frac{1}{d}\right) + (1-\delta)\eta^2\left(2 - \frac{1}{d}\right) < 0 \\
\Rightarrow \quad & 3d\delta - 3 + 2 - 2\delta\frac{1}{d} + \frac{\delta}{d} < 0 \\
\Rightarrow \quad & \delta\left(3d - 2 + \frac{1}{d}\right) < 1 + \frac{1}{d} \\
\Rightarrow \quad & \delta < \frac{1 + \frac{1}{d}}{\left(3d - 2 + \frac{1}{d}\right)}.
\end{aligned} \tag{4.25}$$

Therefore it suffices that

$$\delta < \frac{1}{3d}. \tag{4.26}$$

Following from λ-strong convexity of f, we have

$$\|w_{t-1} - w^*\|^2 \leq \frac{1}{\lambda} g_{t-1}^T \left(w_{t-1} - w^*\right). \tag{4.27}$$

Hence,

$$\begin{aligned}&\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] - (1-\delta)\, Q(\theta_{t-1}) \leq \\ &\qquad\left(3\eta^2 L - \frac{2\eta}{d} + \frac{\delta}{\lambda d} - \frac{\delta^2\left(1-\frac{1}{d}\right)^2}{\lambda\left(3d\delta - 1 + \frac{\delta-1}{d} - 2\delta\right)}\right)\|w_{t-1} - w^*\|^2.\end{aligned} \tag{4.28}$$

If we set $\delta = \frac{\lambda}{8dL}$ and $\eta = \frac{1}{2dL}$, assuming $\lambda \leq L$, we have

$$\begin{aligned}3\eta^2 L - \frac{2\eta}{d} + \frac{\delta}{\lambda d} - \frac{\delta^2\left(1-\frac{1}{d}\right)^2}{\lambda\left(3d\delta - 1 + \frac{\delta-1}{d} - 2\delta\right)} =& \frac{3}{4d^2L} - \frac{1}{d^2L} + \frac{1}{8d^2L} \\ &- \frac{\delta^2\left(1-\frac{1}{d}\right)^2}{\lambda\left(3d\delta - 1 + \frac{\delta-1}{d} - 2\delta\right)} \\ =& -\frac{1}{8d^2L} + \frac{\delta^2\left(1-\frac{1}{d}\right)^2}{\lambda\left(1 - 3d\delta + 2\delta - \frac{\delta-1}{d}\right)} \\ \leq& -\frac{1}{8d^2L} + \frac{\delta^2}{\lambda\left(1-3d\delta\right)} \\ =& -\frac{1}{8d^2L} + \frac{\lambda}{64d^2L^2\left(1-\frac{3\lambda}{8L}\right)} \\ \leq& -\frac{1}{8d^2L} + \frac{\lambda}{64d^2L^2\left(1-\frac{3}{8}\right)} \\ \leq& -\frac{1}{8d^2L} + \frac{1}{40d^2L} \\ <& 0.\end{aligned} \tag{4.29}$$

Note that $\delta = \frac{\lambda}{8dL}$ satisfies (4.26). Inequality (4.29) implies

$$\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] - (1-\delta)\, Q(\theta_{t-1}) < 0 \tag{4.30}$$

or

$$\mathbb{E}\left[Q(\theta_t)|\mathcal{F}_{t-1}\right] < (1-\delta)\, Q(\theta_{t-1}). \tag{4.31}$$

Taking expectation on both sides with respect to $\theta_0, \theta_1, \ldots, \theta_{t-1}$ and substituting the value for

δ, we get

$$\mathbb{E}\left[Q(\theta_t)\right] < \left(1 - \frac{\lambda}{8dL}\right) \mathbb{E}\left[Q(\theta_{t-1})\right]. \tag{4.32}$$

Therefore,

$$\mathbb{E}\left[Q(\theta_t)\right] < \left(1 - \frac{\lambda}{8dL}\right)^t Q(\theta_0). \tag{4.33}$$

Initializing $\widehat{g_0} = 0$ and $w_0 = 0$ results in

$$Q(\theta_0) = \widehat{g_0}^T A \widehat{g_0} + 2\widehat{g_0}^T B \left(w_0 - w^*\right) + \left(w_0 - w^*\right)^T C \left(w_0 - w^*\right) = \frac{1}{d}\|w^*\|^2. \tag{4.34}$$

Hence,

$$\mathbb{E}\left[Q(\theta_t)\right] < \left(1 - \frac{\lambda}{8dL}\right)^t \frac{1}{d}\|w^*\|^2. \tag{4.35}$$

We now need to prove that $Q(\theta_t)$ (the Lyapunov function) dominates $\|w_t - w^*\|^2$. If $P' = P - \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2d}I \end{pmatrix} = \begin{pmatrix} A & B \\ B^T & C - \frac{1}{2d}I \end{pmatrix}$ is positive definite, then $x^T P x \succ x^T \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{2d}I \end{pmatrix} x$ for any $x \in \mathcal{R}^d$. Therefore, we have $Q(\theta_t) > \frac{1}{2d}\|w_t - w^*\|^2$. We know that P' is positive definite if and only if both A and the Schur complement of P', i.e. $S' = (C - \frac{1}{2d}I) - B^T A^{-1} B$, are positive definite [27]. $A = \left[3d\eta^2 + \frac{\eta^2}{d} - 2\eta^2\right] I$ is obviously positive definite for all acceptable values of d. We now check S' to determine if it is positive definite.

$$\begin{aligned} S' &= \left(\frac{1}{d}I - \frac{1}{2d}I\right) - \frac{\eta^2\left(1 - \frac{1}{d}\right)^2}{3d\eta^2 + \frac{\eta^2}{d} - 2\eta^2} I \\ &= \left[\frac{1}{2d} - \frac{\eta^2\left(1 - \frac{1}{d}\right)^2}{3d\eta^2 + \frac{\eta^2}{d} - 2\eta^2}\right] I \\ &\succ \left[\frac{1}{2d} - \frac{1}{3d-2}\right] I \\ &\succeq 0 \end{aligned} \tag{4.36}$$

for $d \geq 2$. Therefore, P' is positive definite for $d \geq 2$. Then,

$$\frac{1}{2d}\|w_t - w^*\|^2 < Q(\theta_t) \tag{4.37}$$

Therefore,

$$\mathbb{E}\left[\|w_t - w^*\|^2\right] < 2d\mathbb{E}\left[Q(\theta_t)\right]. \tag{4.38}$$

From this and (4.35) we get

$$\mathbb{E}\left[\|w_T - w^*\|^2\right] < 2\|w^*\|^2 \left(1 - \frac{\lambda}{8dL}\right)^T. \tag{4.39}$$

Finally, from smoothness of $f(w)$ we have

$$\mathbb{E}\left[f(w_T) - f(w^*)\right] < L\ \|w^*\|^2 \left(1 - \frac{\lambda}{8dL}\right)^T. \tag{4.40}$$

□

which shows an exponential rate of convergence for this method.

4.2.1 Discussion

The SVRC method presented here provides linear convergence rate for strongly convex smooth functions. Instead of computing the true gradient vector at every iteration, our method only updates one element of the coordinate vector. It also addresses the memory issue of the SAG method and improves its convergence rate (compare $\left(1 - \frac{\lambda}{8dL}\right)^T$ of SVRC to $\left(1 - \frac{\lambda}{8mL}\right)^T$ of SAG), at the cost of higher per-iteration complexity. A future direction of research in this area can be using second order (Hessian) information to achieve faster convergence rates.

As the name of this method suggests, SVRC has reduced variance in results compared to the standard stochastic coordinate descent (SCD) method. The variance reduction analysis for SVRC falls within the framework for variance reduction presented in [10] and mentioned in Section 2.2.3 of Chapter 2. The idea behind the methods following this framework is that if X is a random variable and we are interested in estimating $\mathbb{E}[X]$, we can introduce have random variable Y which is highly correlated with X for which we can easily compute its expectation. Now, define $\theta_\alpha = \alpha(X - Y) + \mathbb{E}[Y]$ as an estimator of $\mathbb{E}[X]$. We observe that $\mathbb{E}[\theta_\alpha] = \alpha\mathbb{E}[X] + (1-\alpha)\mathbb{E}[Y]$ and $Var(\theta_\alpha) = \alpha^2\left[Var(X) + Var(Y) - 2\ Cov(X,Y)\right]$. If the covariance is large enough, $Var(\theta_\alpha)$ can be smaller than $Var(X)$.

If we rewrite the update rule of SVRC as:

$$w_{t+1} = w_t - \eta_t\left[\nabla_j f(w_t) - \widehat{g}_j(t-1) + \widehat{g}(t-1)\right], \tag{4.41}$$

we can see that here $X = d\nabla_j f(w_t)$ and $Y = d\widehat{g}_j(t-1)$, $\theta = \widehat{g}(t)$, $\alpha = \frac{1}{d}$, and the expectation of Y is taken over the choice of j.

Chapter 5

Distributed Optimization in Networks with Limited Communication Resources

In a general connected network, we aim to minimize the average of the local objective functions associated with the nodes of the network:

$$\min_{w \in \mathcal{R}^d} f_D(w) = \frac{1}{n} \sum_{i=1}^{n} f^i(w), \tag{5.1}$$

where $\{f^i(w)\}$ are strongly convex functions.

It is tempting to directly apply proportional gradient sampling to the general network setting for distributed optimization. A naïve adoption of our centralized method for distributed problems would involve communicating proportionally sampled estimates of the local gradients with the neighbors in order to converge to a common optimal point of the global objective function. In Chapter 3 we showed that this method actually works if the nodes are fully connected where all of the nodes start with the same initial value. However, sending gradient information to neighbors in the network does not necessarily help them reach the global minimizer because at any instant t, various nodes have different values of estimates w_t^i, so the gradient values from the neighbors might be totally irrelevant. Hence, for general connected networks we suggest that the nodes exchange partial information about their current local estimates $\{w_t^i\}_{i=1}^n$ instead of communicating gradient information.

5.1 Distributed Coordinate-wise Primal Averaging

The method in this section is motivated by the problem of limiting communication during the iterations. Based on two existing works, [12] and [28], on consensus-based distributed optimization and the social sampling protocol suggested by Sarwate and Javidi [29], we consider a communication scheme based on sending partial information about the current iterate w_t^i: each node only sends an estimate (sample) $\widehat{w}_i^t$ of their current belief w_i^t. In particular, $\widehat{w}_i^t$ only contains information about a single (random) coordinate of w_t^i.

5.1.1 Synchronous Coordinate-wise Primal Averaging

A simple model is for an oracle to select a coordinate of $\{w_t^i\}$ and have all nodes synchronized to transmit information about that same coordinate. This method can be seen as a coordinate-wise variant of the distributed primal averaging algorithm of Nedić and Ozdaglar [12]. In this algorithm, at every iteration some oracle selects a random coordinate j from $\{1, \cdots, n\}$ and orders all nodes to send the j-th coordinate of w_t^i to their neighbors. Each node i updates its j-th coordinate with a weighted average of their neighbor's coordinates. Subsequently, node i updates all coordinates of w_t^i using its full local gradient $g_t^i = \nabla f^i(w_t^i)$. Therefore, the update rule can be written as

$$w_{t+1}^i = \sum_{j=1}^{d} \sum_{k \in \widehat{\mathcal{N}}^i} Q_{ik}^j(t) D^j w_t^k - \eta_t^i g_t^i \qquad \text{for } j \in [d], \tag{5.2}$$

where D^j is a diagonal matrix with 1 on its jth diagonal element and zero otherwise and

$$Q^j(t) = \begin{cases} Q & \text{if } j \text{ is selected at time } t, \\ I & \text{otherwise.} \end{cases} \tag{5.3}$$

5.1.2 Asynchronous Coordinate-wise Primal Averaging

The synchronous algorithm requires an oracle to select a common coordinate j so that all nodes average on the same coordinate at each time. In a more realistic scenario, however, we would like to allow each node i to select its own coordinate j^i at random. It can then send the pair $\left(j^i, \left(w_t^i\right)_{j^i}\right)$ to its neighbors. Then, node i will receive a collection of *requests* $\left\{\left(j^k, \left(w_t^i\right)_{j^k}\right) : k \in \mathcal{N}^i\right\}$. For each $k \in \mathcal{N}^i$, it will send $\left(j^k, \left(w_t^i\right)_{j^k}\right)$ to node k. This preserves the bidirectionality of the links. Each node, for every coordinate, computes a convex combination of its own belief and those of the neighbors who have sent their information about the same coordinate. For different coordinates, the assigned weights to the neighbors need not be the same.

For this algorithm the update rule is the same as (5.2), except that for every coordinate j, matrix $Q^j(t)$ is chosen i.i.d across time according to $\mathcal{P}_j^{\mathcal{G}}$, which is a probability distribution over $\mathcal{Q}_j^{\mathcal{G}}$, a set of doubly stochastic matrices comfortant to subgraphs of $\mathcal{G}$ each of which include all of the self-loops.

Algorithm 7 Asynchronous Coordinate-wise Primal Averaging

```
Require: $N$, $T$, graph $\mathcal{G}$, step size sequence $\{\eta_t\}$, matrix $Q$
  arbitrarily select $w_1^i \in \mathcal{R}^d$ for all $i \in [n]$.
  for $t = 1, 2, \ldots T$ do
    for all $i \in [n]$ do
      compute $g_t^i \in \partial f^i(w_t^i)$
      select $j^i$ uniformly in $[d]$
      send $\left(j^i, \left(w_t^i\right)_{j^i}\right)$ to all nodes in $\mathcal{N}^i$
    end for
    for all $i \in [n]$ do
      send $\left(j^k, \left(w_t^i\right)_{j^k}\right)$ to each node $k \in \mathcal{N}^i$
    end for
    for all $i \in [n]$ do
      for all $j \in [d]$ do
        if $j \in \{j^k : k \in \bar{\mathcal{N}}^i\}$ then
          $(v_{t+1}^i)_j = \sum_{k \in \bar{\mathcal{N}}^i} Q_{ik}^j(t)(w_t^k)_j$
        else
          $(v_{t+1}^i)_j = (w_t^i)_j$
        end if
      end for
      $w_{t+1}^i = v_{t+1}^i - \eta_t g_t^i$
    end for
  end for
  return for each $i \in [n]$ the average

      $$\widetilde{w}_T^i = \frac{1}{T} \sum_{t=1}^{T} w_t^i.$$

  for each $i \in [n]$
```

5.1.3 Convergence Rate

Theorem 5 provides the convergence analysis for the case with deterministic Q^j across time for every coordinate. This provides the basis for analyzing both synchronous and asynchronous methods where the weight matrices are random. Our analysis relates several average quantities such as the time average

$$\widetilde{w}_T^i = \frac{1}{T}\sum_{t=1}^{T} w_t^i, \tag{5.4}$$

the network average

$$\bar{w}_t = \frac{1}{n}\sum_{i=1}^{n} w_t^i, \tag{5.5}$$

and the time-and-network average

$$\widetilde{w} = \sum_{i=1}^{n}\frac{\widetilde{w}^i}{n} = \sum_{t=1}^{T}\frac{\bar{w}_t}{T}. \tag{5.6}$$

Theorem 5. *Consider solving problem* (5.1)*. Suppose that every node uses update rule* (5.2) *with the same step size $\eta_t = \frac{1}{\lambda t}$ across the network and $Q^j(t) = Q^j$ for all $j \in [d]$. Furthermore, assume that our objective functions are λ-strongly convex and have bounded gradients, that is for any vector $w \in \mathcal{R}^d$ we have $\|\nabla f^i(w)\| \leq M$ and $|\nabla_j f^i(w)| \leq M_j$. Then, for $T \geq \max_{j\in[d]}\{-2e\log(\sqrt{\lambda_2(Q^j)})\}$,*

$$\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] \leq (C_1 + C_2\log(T))\,\frac{\log(T)}{T}, \tag{5.7}$$

where

$$\begin{aligned} C_1 &= \frac{M^2}{2n\lambda} \\ C_2 &= \frac{18MM'\sqrt{n}}{\lambda} \\ M' &= \sum_{j=1}^{d}\frac{M_j}{-\log(\sqrt{\lambda_2(Q^j)})}. \end{aligned}$$

Proof. We take an approach similar to Nedić and Ozdaglar [12]. We first find a bound on the expected distance of the network average (5.5) from the optimal point w^*. Note that since Q^j

is doubly stochastic for all $j \in [d]$, we have the following recursive relation:

$$\bar{w}_{t+1} = \bar{w}_t - \eta_t \sum_{i=1}^{n} \frac{g_t^i}{n}. \tag{5.8}$$

This implies the following recursion:

$$\begin{aligned}
\|\bar{w}_{t+1} - w^*\|^2 &= \left\| \bar{w}_t - w^* - \eta_t \sum_{i=1}^{n} \frac{g_t^i}{n} \right\|^2 \\
&= \|\bar{w}_t - w^*\|^2 + \left\| \eta_t \sum_{i=1}^{n} \frac{g_t^i}{n} \right\|^2 - 2\eta_t (\bar{w}_t - w^*)^\top \sum_{i=1}^{n} \frac{g_t^i}{n} \\
&\overset{(a)}{\leq} \|\bar{w}_t - w^*\|^2 + \eta_t^2 \frac{M^2}{n} - 2\eta_t \sum_{i=1}^{n} \frac{(\bar{w}_t - w^*)^\top g_t^i}{n},
\end{aligned} \tag{5.9}$$

where in (a) we made use of the finite form of Jensen's inequality. For the the summands of the third term above we have

$$\begin{aligned}
(\bar{w}_t - w^*)^\top g_t^i &= (\bar{w}_t - w^*)^\top \nabla f^i(w_t^i) \\
&= (\bar{w}_t - w_t^i)^\top \nabla f^i(w_t^i) + (w_t^i - w^*)^\top \nabla f^i(w_t^i) \\
&\overset{(a)}{\geq} - \left\|\nabla f^i(w_t^i)\right\| \|\bar{w}_t - w^*\| + f^i(w_t^i) - f^i(w^*) \\
&\qquad + \frac{\lambda}{2} \left\|w_t^i - w^*\right\|^2 \\
&= - \left\|\nabla f^i(w_t^i)\right\| \|\bar{w}_t - w^*\| + f^i(w_t^i) - f^i(\bar{w}_t) \\
&\qquad + \frac{\lambda}{2} \left\|w_t^i - w^*\right\|^2 + f^i(\bar{w}_t) - f^i(w^*) \\
&\overset{(b)}{\geq} - \left\|\nabla f^i(w_t^i)\right\| \left\|\bar{w}_t - w_t^i\right\| + \nabla f^i(\bar{w}_t)^\top (w_t^i - \bar{w}_t) \\
&\qquad + \frac{\lambda}{2} \left\|w_t^i - w^*\right\|^2 + f^i(\bar{w}_t) - f^i(w^*) \\
&\overset{(c)}{\geq} - \left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\| \\
&\qquad + \frac{\lambda}{2} \left\|w_t^i - w^*\right\|^2 + f^i(\bar{w}_t) - f^i(w^*),
\end{aligned} \tag{5.10}$$

where (a) follows from Cauchy-Shwartz inequality and strong convexity of f^i, (b) is a result of convexity of f^i and (c) also results from Cauchy-Shwartz inequality. Using inequality (5.10) we can upper-bound the third term in the r.h.s of (5.9):

$$\begin{aligned}
-2\eta_t \sum_{i=1}^{n} \frac{\langle (\bar{w}_t - w^*), g_t^i \rangle}{n} \leq{}& 2\eta_t \sum_{i=1}^{n} \frac{\left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\|}{n} \\
&- \lambda\eta_t \sum_{i=1}^{n} \frac{\left\|w_t^i - w^*\right\|^2}{n} - 2\eta_t \sum_{i=1}^{n} \frac{f^i(\bar{w}_t) - f^i(w^*)}{n} \\
\overset{(a)}{\leq}{}& 2\eta_t \sum_{i=1}^{n} \frac{\left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\|}{n} \\
&- \lambda\eta_t \left\|\bar{w}_t - w^*\right\|^2 - 2\eta_t \left(f_D(\bar{w}_t) - f_D(w^*)\right), \qquad (5.11)
\end{aligned}$$

where (a) results from finite form Jensen's Inequality as well as the definition of f_D. Substituting this result in (5.9) and taking expectation w.r.t. the entire history up to time t we get

$$\begin{aligned}
\mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right] \leq{}& \mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] + \eta_t^2 \frac{M^2}{n} \\
&+ 2\eta_t \sum_{i=1}^{n} \frac{\mathbb{E}\left[\left(\left\|\nabla f^i(w_t^i)\right\| + \left\|\nabla f^i(\bar{w}_t)\right\|\right) \left\|\bar{w}_t - w_t^i\right\|\right]}{n} \\
&- \lambda\eta_t \mathbb{E}\left[\left\|\bar{w}_t - w^*\right\|^2\right] - 2\eta_t \mathbb{E}\left[f(\bar{w}_t) - f_D(w^*)\right] \\
\leq{}& (1 - \lambda\eta_t)\, \mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] + \eta_t^2 \frac{M^2}{n} \\
&+ 4\eta_t M \mathbb{E}\left[\left\|\bar{w}_t - w_t^i\right\|\right] - 2\eta_t \mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right]. \qquad (5.12)
\end{aligned}$$

By rearranging the terms we get

$$\begin{aligned}
\mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right] \leq{}& \frac{1 - \lambda\eta_t}{2\eta_t} \mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] - \frac{1}{2\eta_t} \mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right] \\
&+ \frac{\eta_t}{2n} M^2 + 2M \mathbb{E}\left[\left\|\bar{w}_t - w_t^i\right\|\right]. \qquad (5.13)
\end{aligned}$$

The following Lemma provides us with a bound on $\mathbb{E}\left[\left\|\bar{w}_t - w_t^i\right\|\right]$.

Lemma 4. *Suppose all the assumptions in Theorem 5 hold. For any node i at any time t the network average $\bar{w}_t$ in* (5.5) *satisfies*

$$\mathbb{E}\left[\left\|\bar{w}_{t+1} - w_{t+1}^i\right\|\right] \leq \frac{2\sqrt{n}}{\lambda} \sum_{j=1}^{d} M_j \frac{\log(2b_j e\ t^2)}{b_j\ t}, \qquad (5.14)$$

where $b_j = -\log(\sqrt{\lambda_2(Q^j)})$.

Proof. we define the $n \times d$ matrices W_t, G_t, and $\bar{W}_t$ whose i-th rows are w_t^i, g_t^i, and $\bar{w}_t$. We further define $n \times 1$ vectors $W_t^j = (W_t)_{:,j}$, $G_t^j = (G_t)_{:,j}$, and $\bar{W}_t^j = (\bar{W}_t)_{:,j}$. Recall that Q^j

is $n \times n$. Using this notation, we have the following update rule for the network for every coordinate:

$$W_{t+1}^j = Q^j W_t^j - \eta_t G_t^j, \tag{5.15}$$

Therefore,

$$\begin{aligned}
\bar{W}_{t+1}^j - W_{t+1}^j &= \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) W_{t+1}^j \\
&= \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) \left(Q^j W_t^j - \eta_t G_t^j\right) \\
&= \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) Q^j \left(Q^j W_{t-1} - \eta_{t-1} G_{t-1}^j\right) \\
&\quad - \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) \eta_t G_t^j \\
&= \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) (Q^j)^2 W_{t-1}^j \\
&\quad - \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) Q^j \eta_{t-1} G_{t-1}^j - \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) \eta_t G_t^j.
\end{aligned} \tag{5.16}$$

Assuming $W_1 = \mathbf{0}$, if we continue the process we get

$$\begin{aligned}
\bar{W}_{t+1}^j - W_{t+1}^j &= -\sum_{s=1}^{t} \eta_s \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - I\right) \left(Q^j\right)^{t-s} G_s^j \\
&= -\sum_{s=1}^{t} \eta_s \left(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - \left(Q^j\right)^{t-s}\right) G_s,
\end{aligned} \tag{5.17}$$

where the last line results from the fact that for any doubly stochastic matrix A, we have that $\mathbf{1}\mathbf{1}^\top A = \mathbf{1}\mathbf{1}^\top$.

Let $\eta_t = \frac{1}{\lambda t}$. Taking the absolute value of the i-th elements of both sides of (5.17) and using Jensen's inequality results in

$$\begin{aligned}
\left|(\bar{w}_{t+1})_j - (w_{t+1}^i)_j\right| &= \left|\sum_{s=1}^{t} \frac{1}{\lambda s} \left(\frac{1}{n}\mathbf{1}^\top - \left(Q^j\right)_{i,:}^{t-s}\right)^\top G_s^j\right| \\
&\leq \sum_{s=1}^{t} \frac{1}{\lambda s} \left|\left(\frac{1}{n}\mathbf{1}^\top - \left(Q^j\right)_{i,:}^{t-s}\right)^\top G_s^j\right| \\
&\leq \sum_{s=1}^{t} \frac{M_j}{\lambda s} \left\|\frac{1}{n}\mathbf{1}^\top - \left(Q^j\right)_{i,:}^{t-s}\right\|_1 .
\end{aligned} \tag{5.18}$$

Now, we get the following bound for the summation in the right-hand side of (5.18) [30]:

$$\sum_{s=1}^{t} \frac{M_j}{\lambda s} \left\| \frac{1}{n}\mathbf{1}^\top - \left(Q^j\right)_{i,:}^{t-s} \right\|_1 \leq \frac{M_j\sqrt{n}}{\lambda} \sum_{s=1}^{t} \frac{1}{s} \left(\sqrt{\lambda_2(Q^j)}\right)^{t-s}$$
$$\leq \frac{2M_j\sqrt{n}}{\lambda} \frac{\log(2b_j e\ t^2)}{b_j\ t}, \tag{5.19}$$

where $\lambda_2(Q^j) \leq 1$ is the second eigenvalue of matrix Q^j and $b_j = -\log(\sqrt{\lambda_2(Q^j)})$. Summing over all $j \in [d]$ and taking expectation of both sides of (5.20) with respect to the entire history results in

$$\begin{aligned}\mathbb{E}\left[\left\|\bar{w}_{t+1} - w^i_{t+1}\right\|\right] \leq & \mathbb{E}\left[\left\|\bar{w}_{t+1} - w^i_{t+1}\right\|_1\right] \\ = & \sum_{j=1}^{d} \mathbb{E}\left[\left|(\bar{w}_{t+1})_j - (w^i_{t+1})_j\right|\right] \\ \leq & \frac{2\sqrt{n}}{\lambda} \sum_{j=1}^{d} M_j \frac{\log(2b_j e\ t^2)}{b_j\ t}. \end{aligned} \tag{5.20}$$

□

Substituting the result of Lemma 4 and $\eta_t = \frac{1}{\lambda t}$ in (5.13):

$$\begin{aligned}\mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right] \leq & \frac{\lambda(t-1)}{2}\mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] - \frac{\lambda t}{2}\mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right] \\ & + \frac{M^2}{2n\lambda t} + \frac{4M\sqrt{n}}{\lambda} \sum_{j=1}^{d} M_j \frac{\log(2b_j e\ t^2)}{b_j\ t}. \end{aligned} \tag{5.21}$$

This provides a bound on $\mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right]$. However, in order to analyze the asymptotic behavior of our algorithms, we are interested in $\mathbb{E}\left[f_D(\widetilde{w}^i) - f_D(w^*)\right]$. Consider the time-and-network average (5.6). From convexity of f_D and Jensen's inequality we have

$$\begin{aligned}\mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*)\right] \leq & \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right] \\ \leq & \sum_{t=1}^{T} \frac{\lambda(t-1)\mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] - \lambda t \mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right]}{2T} \\ & + \frac{1}{T}\sum_{t=1}^{T} \frac{M^2}{2n\lambda t} + \frac{1}{T}\sum_{t=1}^{T} \frac{4M\sqrt{n}}{\lambda} \sum_{j=1}^{d} M_j \frac{\log(2b_j e\ t^2)}{b_j\ t}. \end{aligned} \tag{5.22}$$

Using convexity, Jensen's inequality, and some algebra we can establish that for $T > \max_{j\in[d]}\{2b_j e\}$,

$$\begin{aligned}\mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*)\right] &\le \frac{1}{T}\sum_{t=1} -\frac{\lambda T}{2}\mathbb{E}\left[\|\bar{w}_{T+1} - w^*\|^2\right] \\ &\quad + \frac{1}{T}\sum_{t=1}^{T}\left(\frac{M^2}{2n\lambda} + \frac{12MM'\sqrt{n}}{\lambda}\log(T)\right)\frac{1}{t} \\ &\le \frac{C_T}{T}\sum_{t=1}^{T}\frac{1}{t} \\ &\le C_T\frac{\log(T)}{T}, \end{aligned} \tag{5.23}$$

where $M' = \sum_{j=1}^{d}\frac{M_j}{b_j}$ and $C_T = \frac{M^2}{2n\lambda} + \frac{12MM'\sqrt{n}}{\lambda}\log(T)$. This lets us relate the time average $\widetilde{w}_T^i$ at a node to the network time average:

$$\begin{aligned}\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] &\le \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*) + \nabla f_D(\widetilde{w}_t^i)^\top(\widetilde{w}_t^i - \widetilde{w}_t)\right] \\ &\overset{(a)}{\le} \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*) + \left\|\nabla f_D(\widetilde{w}_T^i)\right\|\left\|\widetilde{w}_t^i - \widetilde{w}_T\right\|\right] \\ &\overset{(b)}{\le} \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*)\right] + \mathbb{E}\left[\left\|\nabla f_D(\widetilde{w}_T^i)\right\|\sum_{t=1}^{T}\frac{\left\|w_t^i - \bar{w}_t\right\|}{T}\right] \\ &\le \mathbb{E}\left[f_D(\widetilde{w}_T) - f_D(w^*)\right] + \frac{M}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|w_t^i - \bar{w}_t\right\|\right], \end{aligned} \tag{5.24}$$

where in the inequality (a) we used Cauchy-Shwartz and the (b) results from Jensen's inequality and the definition of $\widetilde{w}_T^i$ and $\widetilde{w}_T$.

Combining the results from (5.24), (5.23), and Lemma 4 gives us the desired upper bound on the loss:

$$\begin{aligned}&\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] \\ &\le C_T\frac{\log(T)}{T} + \frac{2M\sqrt{n}}{\lambda T}\sum_{t=1}^{T}\sum_{j=1}^{d}\frac{M_j\log(2b_j e\ t^2)}{b_j\ t} \\ &\overset{(a)}{\le} C_T\frac{\log(T)}{T} + \frac{6MM'\sqrt{n}\log(T)}{\lambda T}\sum_{t=1}^{T}\frac{1}{t} \\ &\le \left(\frac{M^2}{2n\lambda} + \frac{12MM'\sqrt{n}}{\lambda}\log(T)\right)\frac{\log(T)}{T} \\ &\quad + \frac{6MM'\sqrt{n}\log(T)}{\lambda}\frac{\log(T)}{T} \\ &= \left(\frac{M^2}{2n\lambda} + \frac{18MM'\sqrt{n}}{\lambda}\log(T)\right)\frac{\log(T)}{T}. \end{aligned} \tag{5.25}$$

where (a) results from the assumption $T > \max_{j\in[d]}\{2b_j e\}$, the fact $t < T$ and the definition $M' = \sum_{j=1}^{d} \frac{M_j}{b_j}$. □

Now that we have established the results for the case with time-invariant weight matrices, we go on to study the case where $Q^j(t)$ is selected i.i.d. from distribution $\mathcal{P}_j^{\mathcal{G}}$. Note that for every coordinate j, the quantity $\mathcal{P}_j^{\mathcal{G}}$ is a function of $\{\mathcal{P}_i^C\}_{i=1}^n$ that are probability distributions over coordinate indexes at each node. The following lemma is used in the sequel to analyze the behavior of our methods.

Lemma 5. *Suppose the same assumptions as Theorem (5) hold except that in (5.2), $Q^j(t)$ is an i.i.d sequence of doubly stochastic matrices drawn from distribution $\mathcal{P}_j^{\mathcal{G}}$. Then, for w_t^i and $\bar{w}_t$ we have that*

$$\mathbb{E}\left[\left\|\bar{w}_{t+1} - w_{t+1}^i\right\|\right] \leq \sum_{j=1}^{d} \frac{2M_j\sqrt{n}}{\lambda}\frac{\log(2b_j e\ t^2)}{b_j\ t}, \tag{5.26}$$

where $b_j = -\log\left(\mathbb{E}\left[\sigma_2\left(Q^j(t)\right)\right]\right)$ and $\sigma_2\left(Q^j\right)$ is the second largest singular value of $Q^j(t)$.

Proof. Define $\mathbb{Q}^j(s,t) = Q^j(t)Q^j(t-1)\cdots Q^j(s+1)$. Following the procedure in Lemma 4, we get

$$\begin{aligned}
\left|(\bar{w}_{t+1})_j - (w_{t+1}^i)_j\right| &\leq \sum_{s=1}^{t} \frac{M_j}{\lambda s}\left\|\frac{1}{n}\mathbf{1}^\top - \mathbb{Q}^j(s,t)_{i,:}\right\|_1 \\
&\leq \sum_{s=1}^{t-1} \frac{M_j}{\lambda s}\left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top\mathbb{Q}^j(s,t)\right\|_1 + \frac{2M^j}{\lambda t} \\
&\leq \sum_{s=1}^{t-1} \frac{M_j\sqrt{n}}{\lambda s}\left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top\mathbb{Q}^j(s,t)\right\|_2 + \frac{2M^j}{\lambda t}.
\end{aligned} \tag{5.27}$$

For the summand of the first term here we have

$$\begin{aligned}
\left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top \mathbb{Q}^j(s,t)\right\|_2^2 &= \left(\frac{\mathbf{1}^\top}{n} - (e^i)^\top \mathbb{Q}^j(s+1,t)Q^j(s+1)\right) \\
&\qquad \left(\frac{\mathbf{1}^\top}{n} - (e^i)^\top \mathbb{Q}^j(s+1,t)Q^j(s+1)\right)^\top \\
&= \frac{\mathbf{1}^\top\mathbf{1}}{n^2} + (e^i)^\top \mathbb{Q}^j(s+1,t)Q^j(s)Q^j(s+1)^\top \mathbb{Q}^j(s+1,t)^\top e^i \\
&\quad - \frac{2}{n}\mathbf{1}^\top Q^j(s+1)^\top \mathbb{Q}^j(s+1,t)^\top e^i \\
&= \frac{\mathbf{1}^\top Q^j(s+1)Q^j(s+1)^\top \mathbf{1}}{n^2} \\
&\quad + (e^i)^\top \mathbb{Q}^j(s+1,t)Q^j(s+1)Q^j(s+1)^\top \mathbb{Q}^j(s+1,t)^\top e^i \\
&\quad - \frac{2}{n}\mathbf{1}^\top Q^j(s+1)Q^j(s+1)^\top \mathbb{Q}^j(s+1,t)^\top e^i \\
&= \left(\frac{\mathbf{1}^\top}{n} - (e^i)^\top \mathbb{Q}^j(s+1,t)\right) Q^j(s+1) \\
&\qquad Q^j(s+1)^\top \left(\frac{\mathbf{1}^\top}{n} - (e^i)^\top \mathbb{Q}^j(s+1,t)\right)^\top \\
&\leq \lambda_2\left(Q^j(s+1)Q^j(s+1)^\top\right) \left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top \mathbb{Q}^j(s+1,t)\right\|_2^2. \qquad (5.28)
\end{aligned}$$

Continuing this recursive procedure results in

$$\begin{aligned}
\left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top \mathbb{Q}^j(s,t)\right\|^2 &\leq \sigma_2^2\left(Q^j(s)\right) \left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top \mathbb{Q}^j(s+1,t)\right\|^2 \\
&\leq \left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top\right\|^2 \prod_{l=s+1}^{t} \sigma_2^2\left(Q^j(l)\right) \qquad (5.29)
\end{aligned}$$

Taking expectation of both sides w.r.t. the entire history and keeping in mind that $\{Q^j(l)\}_{l=s+1}^t$ are i.i.d. over time results in

$$\begin{aligned}
\mathbb{E}\left[\left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top \mathbb{Q}^j(s,t)\right\|\right] &\leq \left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top\right\| \mathbb{E}\left[\prod_{l=s+1}^{t} \sigma_2\left(Q^j(l)\right)\right] \\
&= \left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top\right\| \prod_{l=s+1}^{t} \mathbb{E}\left[\sigma_2\left(Q^j(l)\right)\right] \\
&= \left\|\frac{1}{n}\mathbf{1}^\top - (e^i)^\top\right\| \mathbb{E}\left[\sigma_2\left(Q^j(l)\right)\right]^{t-s} \\
&\leq \mathbb{E}\left[\sigma_2\left(Q^j(\tau)\right)\right]^{t-s}. \qquad (5.30)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}\left[\left\|\bar{w}_{t+1} - w^i_{t+1}\right\|\right] \le \mathbb{E}\left[\left\|\bar{w}_{t+1} - w^i_{t+1}\right\|_1\right] &= \sum_{j=1}^{d} \mathbb{E}\left[\left|(\bar{w}_{t+1})_j - (w^i_{t+1})_j\right|\right] \\
&\le \sum_{j=1}^{d}\left[\sum_{s=1}^{t-1} \frac{M_j\sqrt{n}}{\lambda s}\mathbb{E}\left[\sigma_2\left(Q^j(\tau)\right)\right]^{t-s} + \frac{2M^j}{\lambda t}\right] \\
&\le \sum_{j=1}^{d} \frac{2M^j\sqrt{n}}{\lambda}\frac{\log(2b_j e t^2)}{b_j t},
\end{aligned} \tag{5.31}$$

□

Now, we are ready to find the upper bound on $\mathbb{E}\left[f_D(\widetilde{w}^i_T) - f_D(w^*)\right]$ in the setup with random weight matrices.

Theorem 6. *Assume that all conditions in Lemma 5 hold. We have the following upper bound on the loss of coordinate-wise primal averaging algorithm with update rule* (5.2) *for the optimization problem* (5.1) *if* $T \ge -2\ e\log\left(\mathbb{E}\left[\sigma_2\left(Q^j(t)\right)\right]\right)$*:*

$$\mathbb{E}\left[f_D(\widetilde{w}^i_T) - f_D(w^*)\right] \le \left(\frac{M^2}{2n\lambda} + \frac{18Mc\sqrt{n}}{\lambda}\log(T)\right)\frac{\log(T)}{T}, \tag{5.32}$$

where $c = \sum_{j=1}^{d} \frac{M_j}{-\log(\mathbb{E}[\sigma_2(Q^j(t))])}$ *and* M *and* M_j *are as defined in Theorem* (5)*.*

Proof. By applying Lemma 5 to (5.13) and following the same procedure as that of the proof of Theorem 5, we get the stated result. □

We remark that for both synchronous and asynchronous distributed algorithms proposed in this chapter, the analysis in Theorem 6 holds. The synchronous algorithm is a special case where Q^j is a random matrix which takes value from the set $\{Q, I\}$. In the asynchronous method the sample space is larger, i.e. the entire $\mathcal{Q}^{\mathcal{G}}_j$ which is defined in Subsection 5.1.2.

5.2 Distributed Coordinate-wise Primal Averaging with Stochastic Local Gradients

We claim that our coordinate-wise primal averaging method also works for a setup in which the nodes use unbiased estimates of their gradient vectors instead of the true gradient. The update rule here is

$$w^i_{t+1} = \sum_{j=1}^{d}\sum_{k\in\widehat{\mathcal{N}}^i} Q^j_{ik}(t)D^j w^k_t - \eta^i_t\widehat{g}^i_t \qquad \text{for } j\in[d], \tag{5.33}$$

or

$$\bar{w}_{t+1} = \bar{w}_t - \eta_t \sum_{i=1}^{n} \frac{\widehat{g}_t^i}{n}. \tag{5.34}$$

where $\widehat{g}_t^i$ is an unbiased estimate of the true local gradient vector: $\mathbb{E}\left[\widehat{g}_t^i | \mathcal{F}_t\right] = g_t^i = \nabla f^i(w_t^i)$.

5.2.1 Convergence Analysis

Theorem 7 provides the convergence result in case of using approximate gradients. The result stated in this theorem uses the following lemma which provides us with a bound on $\mathbb{E}\left[\left\|\bar{w}_t - w_t^i\right\|\right]$.

Lemma 6. *Suppose all the assumptions in Theorem 7 hold. For any node i at any time t the network average $\bar{w}_t$ in* (5.5) *satisfies*

$$\mathbb{E}\left[\left\|\bar{w}_{t+1} - w_{t+1}^i\right\|\right] \leq \frac{2\sqrt{n}}{\lambda} \sum_{j=1}^{d} \widehat{M}_j \frac{\log(2b_j e\ t^2)}{b_j\ t}, \tag{5.35}$$

where $\mathbb{E}\left[\left| \left(\widehat{g}_t^i\right)_j \right| |\mathcal{F}_t\right] \leq \widehat{M}_j$ *and* $b_j = -\log(\sqrt{\lambda_2(Q^j)})$.

Proof. Following the same procedure as the proof in Lemma 4 we get the result stated here. □

Now, we are ready to state the main result of the coordinate-wise primal averaging with approximate local gradient.

Theorem 7. *Consider solving problem* (5.1). *Suppose that every node uses update rule* (5.33) *with the same step size $\eta_t = \frac{1}{\lambda t}$ across the network and $Q^j(t) = Q^j$ for all $j \in [d]$. Furthermore, assume that our objective functions are λ-strongly convex and have bounded gradients, that is for any vector $w \in \mathcal{R}^d$ we have $\|\nabla f^i(w)\| \leq M$, $\mathbb{E}\left[|\widehat{g}_t^i|^2\ |\mathcal{F}_t\right] \leq \widehat{M}^2$ and $\mathbb{E}\left[\left| \left(\widehat{g}_t^i\right)_j \right| |\mathcal{F}_t\right] \leq \widehat{M}_j$. Then, for $T \geq \max_{j\in[d]}\{-2e\log(\sqrt{\lambda_2(Q^j)})\}$,*

$$\mathbb{E}\left[f_D(\widetilde{w}_T^i) - f_D(w^*)\right] \leq \left(C_1' + C_2' \log(T)\right) \frac{\log(T)}{T}, \tag{5.36}$$

where

$$C_1' = \frac{\widehat{M}^2}{2n\lambda}, \tag{5.37}$$

$$C_2' = \frac{18M\widehat{M}'\sqrt{n}}{\lambda}, \tag{5.38}$$

$$\widehat{M}' = \sum_{j=1}^{d} \frac{\widehat{M}_j}{-\log(\sqrt{\lambda_2(Q^j)})}. \tag{5.39}$$

Proof. Here, we follow a procedure similar to the previous proofs in this section.

$$\|\bar{w}_{t+1} - w^*\|^2 = \|\bar{w}_t - w^*\|^2 + \left\|\eta_t \sum_{i=1}^{n} \frac{\widehat{g}_t^i}{n}\right\|^2 - 2\eta_t(\bar{w}_t - w^*)^\top \sum_{i=1}^{n} \frac{\widehat{g}_t^i}{n}. \tag{5.40}$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2 \,|\mathcal{F}_t\right] &= \|\bar{w}_t - w^*\|^2 + \mathbb{E}\left[\left\|\eta_t \sum_{i=1}^{n} \frac{\widehat{g}_t^i}{n}\right\|^2\right] - 2\eta_t(\bar{w}_t - w^*)^\top \sum_{i=1}^{n} \frac{g_t^i}{n} \\ &\le \|\bar{w}_t - w^*\|^2 + \eta_t^2 \frac{\widehat{M}^2}{n} - 2\eta_t \sum_{i=1}^{n} \frac{(\bar{w}_t - w^*)^\top g_t^i}{n}. \end{aligned} \tag{5.41}$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[f_D(\bar{w}_t) - f_D(w^*)\right] \le & \frac{1 - \lambda\eta_t}{2\eta_t}\mathbb{E}\left[\|\bar{w}_t - w^*\|^2\right] - \frac{1}{2\eta_t}\mathbb{E}\left[\|\bar{w}_{t+1} - w^*\|^2\right] \\ &+ \frac{\eta_t}{2n}\widehat{M}^2 + 2M\mathbb{E}\left[\|\bar{w}_t - w_t^i\|\right]. \end{aligned} \tag{5.42}$$

Following similar steps as in the proof of Theorem 5 we get the result stated in this theorem.

□

Chapter 6
Conclusion

In this work, we used social sampling to limit the communication in cooperative multi-agent optimization settings. For centralized and shared memory systems, we proposed a new nonuniform variant of stochastic coordinate descent and provided upper bounds on the expected suboptimality gap. This method requires full knowledge of local gradient vectors, which seems computationally wasteful. However, this method may be useful for shared memory systems with limited communication resources where computing local gradient vectors by each node is inexpensive and we are more concerned about the amount of communication or contention among nodes rather than the computation cost.

We also proposed a semi stochastic method with reduced variance compared to stochastic coordinate descent. This algorithm makes use of old gradient values as well as random updates on components of the gradient vector. We showed that this method has linear convergence in a setup where stochastic coordinate descent methods show sublinear convergence, while having lower variance.

In distributed models, we argued that sharing gradient information is not necessarily beneficial; which suggested that nodes should share samples of their current estimates $\{w_t^i\}$. We proposed a stochastic coordinate-wise consensus-based optimization method that requires nodes to share random coordinates of their estimates with their neighbors. We provided convergence analysis and explicit error bounds in expectation for this method.

An interesting question raised in the centralized model is using that how using "stale" gradient values from previous iterations would affect the convergence rate of the algorithm. Less frequent full gradient evaluation drastically reduces the computational cost of the algorithm, therefore a *delayed* PSCD method might solve the intrinsic issue of PSCD that it requires evaluation of the full gradient at every iteration. Analyzing such a scheme would build on recent results of [9] and also the SVRC method presented here. Finally, an empirical evaluation of our methods on typical objective functions, especially in machine learning, may shed more light on when nonuniform sampling can help in practice.

Bibliography

[1] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008. doi: 10.1145/1327452.1327492. URL `http://dl.acm.org/citation.cfmdoid=1327452.1327492`.

[2] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011. URL `http://www.jmlr.org/papers/v12/shalev-shwartz11a.html`.

[3] Stephen J. Wright. Coordinate descent algorithms. Technical Report arXiv:1502.04759 [math.OC], ArXiV, February 2015. URL `http://arxiv.org/abs/1502.04759`.

[4] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems read more: http://epubs.siam.org/doi/abs/10.1137/100802001. *SIAM Journal on Optimization*, 22(2):341–362, 2012. doi: 10.1137/100802001. URL `http://dx.doi.org/10.1137/100802001`.

[5] Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. Technical Report arXiv:1310.3438 [stat.ML], ArXiV, October 2013. URL `http://arxiv.org/abs/1310.3438`.

[6] Zheng Qu, Peter Richtárik, and Tong Zhang. Randomized dual coordinate ascent with arbitrary sampling. Technical Report arXiv:1411.5873 [math.OC], ArXiV, November 2014. URL `http://arxiv.org/abs/1411.5873`.

[7] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling I: Algorithms and complexity. Technical Report arXiv:1412.8060 [math.OC], ArXiV, December 2014. URL `http://arxiv.org/abs/1412.8060`.

[8] Benjamin Recht, Christopher Ré, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011. URL `http://papers.nips.cc/paper/4390-hogwild-a-lock-free-approach-to-parallelizing-stochastic-gradient-descent.pdf`.

[9] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013. URL `http://arxiv.org/abs/1309.2388`.

[10] Aaron Defazio, Francis Bach, and Simon 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. URL `http://papers.nips.cc/paper/5258-saga-a-fast-incremental-gradient-method-with-support-for-non-strongly-convex-composite-objectives.pdf`.

[11] Anand D Sarwate and Tara Javidi. Distributed learning of distributions via social sampling. *IEEE Transactions on Automatic Control*, 60(1):34–45, 2015. doi: 10.1109/TAC.2014.2329611. URL `http://dx.doi.org/10.1109/TAC.2014.2329611`.

[12] Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, January 2009. doi: 10.1109/TAC.2008.2009515. URL `http://dx.doi.org/10.1109/TAC.2008.2009515`.

[13] Angelia Nedić. Asynchronous broadcast-based convex optimization over a network. *IEEE Transactions on Automatic Control*, 56(6):1337–1351, 2011. doi: 10.1109/TAC.2010.2079650. URL `http://dx.doi.org/10.1109/TAC.2010.2079650`.

[14] Ilan Lobel and Asuman Ozdaglar. Distributed subgradient methods for convex optimization over random networks. *IEEE Transactions on Automatic Control*, 56(6):1291–1306, 2011. doi: 10.1109/TAC.2010.2091295. URL `http://dx.doi.org/10.1109/TAC.2010.2091295`.

[15] Kunal Srivastava and Angelia Nedić. Distributed asynchronous constrained stochastic optimization. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):772–790, 2011. doi: 10.1109/JSTSP.2011.2118740. URL `http://dx.doi.org/10.1109/JSTSP.2011.2118740`.

[16] S Sundhar Ram, A Nedić, and Venugopal V Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optimization Theory and Applications*, 147(3):516–545, 2010. doi: 10.1007/s10957-010-9737-7. URL `http://dx.doi.org/10.1007/s10957-010-9737-7`.

[17] Angelia Nedić, Asuman Ozdaglar, and Pablo Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010. doi: 10.1109/TAC.2010.2041686. URL `http://dx.doi.org/10.1109/TAC.2010.2041686`.

[18] John N Tsitsiklis, Dimitri P Bertsekas, Michael Athans, et al. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986. doi: 10.1109/TAC.1986.1104412. URL `http://dx.doi.org/10.1109/TAC.1986.1104412`.

[19] Vivek S Borkar. Asynchronous stochastic approximations. *SIAM Journal on Control and Optimization*, 36(3):840–851, 1998. doi: 10.1137/S0363012995282784. URL `http://dx.doi.org/10.1137/S0363012995282784`.

[20] Dušan Jakovetić, Joao Xavier, and José M. F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014. doi: 10.1109/TAC.2014.2298712. URL `http://dx.doi.org/10.1109/TAC.2014.2298712`.

[21] Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004. URL `http://www.springer.com/us/book/9781402075537`.

[22] Michael Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.

[23] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. *Mathematical Programming, Series B*, 127(1):3–30, October 2011. doi: 10.1007/s10107-010-0420-4. URL `http://dx.doi.org/10.1007/s10107-010-0420-4`.

[24] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. extended version of ICML paper arXiv:1109.5647 [cs.LG], ArXiV, 2012. URL `http://arxiv.org/abs/1109.5647`.

[25] Ji Liu and Stephen J Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015. doi: 10.1137/140961134. URL `http://epubs.siam.org/doi/abs/10.1137/140961134`.

[26] Tuncer Can Aysal, Mehmet Ercan Yildiz, Anand D Sarwate, and Anna Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing*, 57(7): 2748–2761, 2009. doi: 10.1109/TSP.2009.2016247.

[27] Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

[28] Avleen S Bijral, Anand D Sarwate, and Nathan Srebro. On data dependence in distributed stochastic optimization. *arXiv preprint arXiv:1603.04379*, 2016. URL `http://arxiv.org/abs/1603.04379`.

[29] Anand D Sarwate and Tara Javidi. Distributed learning from social sampling. In *2012 46th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2012. doi: 10.1109/CISS.2012.6310767.

[30] Konstantinos I Tsianos and Michael G Rabbat. Distributed strongly convex optimization. *arXiv preprint arXiv:1207.3031*, 2012. URL `http://arxiv.org/abs/1207.3031`.