Fast Distributed First-Order Methods

by

I-An Chen

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Author .....
Department of Electrical Engineering and Computer Science
May 11, 2012

Certified by .....
Asuman Ozdaglar
Class of 1943 Associate Professor
Thesis Supervisor

Accepted by .....
Professor Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Fast Distributed First-Order Methods

by

I-An Chen

Submitted to the Department of Electrical Engineering and Computer Science
on May 11, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

This thesis provides a systematic framework for the development and analysis of distributed optimization methods for multi-agent networks with time-varying connectivity. The goal is to optimize a global objective function which is the sum of local objective functions privately known to individual agents. In our methods, each agent iteratively updates its estimate of the global optimum by optimizing its local function and exchanging estimates with others in the network. We introduce distributed proximal-gradient methods that enable the use of a gradient-based scheme for non-differentiable functions with a favorable structure. We present a convergence rate analysis that highlights the dependence on the step size rule. We also propose a novel fast distributed method that uses Nesterov-type acceleration techniques and multiple communication steps per iteration. Our method achieves exact convergence at the rate of $O(1/t)$ (where t is the number of communication steps taken), which is superior than the rates of existing gradient or subgradient algorithms, and is confirmed by simulation results.

Thesis Supervisor: Asuman Ozdaglar
Title: Class of 1943 Associate Professor

Acknowledgments

This thesis owes its very existence to my advisor, Asu Ozdaglar. I am honored to be under her guidance in an area of her expertise, and I have learned a lot from her in many different ways. Her passion and dedication to research have been an inspiration to me. Our discussions have been enlightening and fruitful— especially when I'm "stuck," which happens to be more often than not, her insight gives me a fresh perspective to the problem and helps me keep a steady course. I also appreciate her kindness and encouragement even while correcting, and her patience in coaching me on critical thinking and technical writing. She has been an incredible advisor, and I would like to thank her for all the growth that resulted from working with her.

I am also grateful for the support of other LIDS members. Special thanks to Ermin, for all the discussions and mentorship, both academically and personally; to Ali, Ozan, Kimon and Elie, for their helpful advice and generous encouragement, and for being such considerate officemates; to Christina, Dawsen, Henghui, Mitra, and the rest of the 6F lunch bunch, for the conversations and laughters; and to every professor, staff member and student at LIDS that I've interacted with, for making my past two years here such a wonderful experience.

Many thanks go to friends who have walked with me through these first two years of graduate school. I have been greatly blessed by brothers and sisters in the MIT Graduate Christian Fellowship and the Northwest Small Group, with whom I could share not only joys but also burdens. I am also thankful for the friendship of other Taiwanese students, and neighbors in the Sidney-Pacific residential community. Grad school may be an exciting place in itself, but it is the people encountered therein that make life here so meaningful and memorable.

Last but definitely not least, this thesis is dedicated to my parents, brothers, and the rest of my family and friends in Taiwan, who are light at the end of the tunnel for me this summer and beyond; to Jerome, for the unfailing support in word and in deed, for the unwavering patience in both good times and bad, for everything. To the only God, our Savior, through Jesus Christ our Lord, be glory, majesty, dominion, and authority, before all time and now and forever.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

We live in an age with an exploding amount of information. With advances in technology, we have been able to collect, store, and process data at an increasing rate and a decreasing cost. When used effectively, comprehensive information gives us a better understanding of the world and helps us improve the quality of life.

However, processing data with a single processor is often restrictive. For example, if we rely on one sensor to make measurements, it may take a long time to gather information about a vast terrain, because no matter how intricate the equipment is, it is still only able to make one measurement at a time. In such cases where information is distributed, it is much more effective to use multiple processing units, and then aggregate the message they gather.

Another limitation is in computing power. As data grow in volume, they may become computationally difficult or time-consuming to process, not to mention the additional effort required for storage and retrieval. For instance, in machine learning, a large number of training samples or features may prevent a problem from being solved effectively on a single machine. Instead, it would be desirable to leverage the power of multiple machines, each processing a portion of the data, and then combine their results.

In view of these considerations, there has recently been a growing interest in

developing distributed methods for solving optimization problems where information is decentralized among multiple agents. These methods usually involve a large number of agents connected through a network. The agents cooperatively solving a global (convex) optimization problem through local computations and information exchange over the network. More specifically, distributed optimization methods seek to solve

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

where m is the number of agents in the network, and for each $i = 1, ..., m$, $f_i(x)$ is a convex function determined by their private information. The agent i maintains x_i, an estimate of the global optimum x^* that minimizes the overall objective $f(x)$.

This general form has many applications. For instance, a team of m sensors exploring an unknown environment are collaboratively solving for the parameter x, which describes the environment, by optimizing

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x) = \frac{1}{m}\sum_{i=1}^{m} \|A_i x - b_i\|^2,$$

where b_i is the measurement taken at agent i and $A_i x$ is the corresponding linear transformation from the parameter space to the measurement space. Note that each agent has access to only one of the terms in the above sum.

As another example, regularized logistic regression in machine learning looks for the optimal parameter x in

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x) = \frac{1}{m}\sum_{i=1}^{m}\left[\frac{1}{|N_i|}\sum_{j\in N_i}\log\left(1+\exp(-b_j\langle a_j, x\rangle)\right) + \lambda\|x\|_1\right],$$

where N_i is the training dataset of agent i, corresponding to $\{a_j \mid j \in N_i\}$, the set of feature vectors, and $\{b_j \mid j \in N_i\}$, the set of associated labels.

The objective in such problems is for each agent to obtain an estimate of the global optimal solution. In reality, several constraints may prevent the agents from sharing their local function $f_i(x)$, since it encodes private information at the agent,

and may take up additional resources to transmit, process, and store. Therefore, a standard approach is to only allow agents to share their estimates x_i of the optimal solution without giving away private local information contained in $f_i(x)$.

This thesis provides a systematic framework for the development and performance analysis of distributed algorithms for multi-agent optimization problems that can operate on a network with time-varying connectivity. Our development will rely on first-order methods (i.e., methods that use gradient or subgradient information), which are low-complexity alternatives to second order methods.

1.2 Related Literature

In this section, we briefly review existing algorithms and convergence rate results that are relevant to our work. Since the literature is broad, we organize our discussion into three sections: centralized methods, parallel and incremental methods, and distributed methods. This is an active field with an extensive literature, and the following is not a comprehensive list, but an overview of the most relevant works.

Centralized Methods. Centralized first-order optimization methods dates back to Cauchy, who proposed the gradient method in 1847 [1]. Today, they are widely used in practice, especially in large-scale problems where higher order methods (such as Newton's method) are computationally expensive. For an objective function $F(x) : \mathbb{R}^d \to \mathbb{R}$ that is convex and continuously differentiable, the basic gradient method generates a sequence of iterates $\{x^k\}_{k=1}^{\infty}$ that approach x^*, the optimum of $F(x)$, by moving the most recent iterate along the direction of steepest descent in function value, which is opposite to the direction of the gradient. With a constant step size, it converges to the optimal solution with $F(x^n) - F^* = O(1/n)$ [1]. [2]. Nesterov [3] proposed an acceleration technique that uses two previous estimates to make a prediction before performing the next gradient step. This leads to an im-

[1] We write $g(n) = O(h(n))$ if and only if there exists a positive real number M and a real number n_0 such that $|g(n)| \leq M|h(n)|$ for all $n \geq n_0$. Convergence rate notion are discussed in more detail in Section 2.5.

proved convergence rate of $O(1/n^2)$, which was also shown to be the best achievable convergence rate [4].

When the objective function is not differentiable, the iterates can be updated using a subgradient direction instead of the gradient direction. This is called the subgradient method. Unlike the gradient, the subgradient direction is not guaranteed to be a direction of descent, due to the fact that for nondifferentiable functions, the value of two iterates can be drastically different despite the iterates being very close. As a result, its performance is worse than that of the gradient method: with a constant step size, it converges at the best-achievable rate of $O(1/\sqrt{n})$ [4].

For certain non-differentiable functions that have favorable structures allowing simple computation of the proximal operator(for example, $F(x) = \|x\|_1$, a common choice for regularization), the proximal-point method [5] can be applied. It utilizes desirable properties of the non-differentiable objective function to solve a minimization problem at each iteration. The convergence rate of the proximal-point method is $O(1/n^2)$, and it is not sensitive to step size choices. There is also the combination of both the gradient and the proximal-point method, called the proximal-gradient method, which is applicable to functions that have both a differentiable part and a non-differentiable part that have a desirable structure. It has been shown to converge at the optimal rate of $O(1/n^2)$ [6] with a constant step size.

Recently, there has also been a growing interest in stochastic approximation [7] and inexact methods [8,9]. These methods are applicable when there is error, introduced by uncertainty or noise, in obtaining the gradient, subgradient, or in the proximal operation. The performance of these methods can be characterized in terms of the error, and techniques for such error analysis are often useful for methods that are not centralized.

Parallel and Incremental Methods. There are several ways to divide the computation load among multiple agents. [10,11] studied the case where every agent had access to the same global objective function that is differentiable. They proposed a generic network communication model, analyzed its consensus properties,

and showed that a gradient-type method converged to the global optimum under this setting.

While we inherit the network communication model in [11], the parallel approach is different from our distributed setting, where the agents have private objective functions that are unknown to others.

Incremental methods, on the other hand, considers agents with private objectives who are connected via a well-structured network, and updates the iterate by passing it around the network and updating it according to each agent's private objective. [12] surveys combinations of gradient, subgradient, and proximal methods. These methods achieve exact convergence only with diminishing step sizes, and there are currently no known techniques for acceleration in incremental gradient methods.

Although they also involve agents with private objectives, incremental methods are different from distributed methods studied in this work. In incremental methods, only one agent updates at a time, whereas in distributed methods, every agent operates in every iteration and maintains an estimate of the global optimum, thereby utilizing full distributed computation power. Moreover, incremental methods rely on cyclic or uniformly random order of passing the iterate, while distributed methods consider more generic communication networks in which agents pass iterates to multiple neighboring agents, and also combines the estimates received from different agents.

Distributed Methods. Our work is closely related to [13], which studied the distributed subgradient method with a constant step size, with convergence rate $O\left(1/\sqrt{n}\right)$. Under a similar framework, [14, 15] considered constrained consensus and optimization, [16] incorporated communication link failures, [17] considered asynchronous updates with stochastic errors, and [18] studied the effect of graph topology on the convergence rate. Other extensions of gradient- or subgradient-based methods include [19], which considers the case where the local functions are time varying but related; and [20], which considers allows the agents to exchange gradient information rather than just estimates.

Our work extends the analysis of [13] in several ways. First, we characterize the convergence results explicitly in terms of the step size choice, giving rates for exact convergence when diminishing step sizes are used. Secondly, and more importantly, we consider proximal gradient methods instead of the subgradient method. In their centralized counterparts, as mentioned above, it has been shown that for the subgradient method, the convergence rate of $O\left(1/\sqrt{n}\right)$ cannot be improved, while Nesterov's techniques can applied to the proximal gradient method to accelerate the convergence rate from $O\left(1/n\right)$ to $O\left(1/n^2\right)$. We were able to use this acceleration technique to obtain an improvement from the subgradient method.

During the preparation of of this work, [21] independently gave a distributed gradient method similar to our setting. Under a static communication network with the same weight for each neighbor, they presented a method that uses diminishing step sizes and converges exactly to the optimum at rate $O\left(\log n/n\right)$. In contrast, our network communication model is time-varying, and we give a method with a constant step size that converges exactly to the optimum at rate $O\left(1/t\right)$ (where t is the number of communication steps taken.)

There are various other distributed methods which are extensions of other centralized optimization methods that utilizes the dual in addition to the primal function. For example, [22] presents the dual averaging subgradient method, which allows for an explicit characterization of how the convergence rate depends on network topology. Also, [23] considers the distributed augmented Lagrangian dual method, and [24] uses an alternating direction method of multipliers (ADMM) for distributed linear regression. These dual methods often involve more complicated computation, but may be more directly applicable in the context of specific problems.

1.3 Contributions

This thesis presents novel distributed methods for solving cooperative optimization problems among multiple agents connected through a potentially time-varying network. The goal is to optimize a global objective function which is the sum of local

objective functions, and each local objective is known by an individual agent only. We have two sets of contributions:

First, we introduce distributed proximal-gradient methods that offer flexibility in exploiting the special structure of local objective functions, enabling the use of a gradient-based scheme for non-differentiable functions with a favorable structure. We present a convergence rate analysis for such methods that highlights the dependence on the step size sequence.

We next propose a *fast distributed gradient method* that uses Nesterov-type acceleration techniques and multiple communication steps per iteration. Our method achieves exact convergence at the rate of $O(1/t)$ (where t is the computation time), superior than the rate achieved by existing gradient or subgradient algorithms.

1.4 Outline

The class of methods considered is outlined in Chapter 2, along with preliminary results that are important for our anaylsis.

In Chapter 3, we consider methods with a single communication step, and study the convergence properties under both constant and diminishing step sizes. For a constant step size rule, it extends current results for subgradient methods to proximal-gradient methods, highlighting the effect of the distributed nature of the problem on the rate of consensus. For diminishing step size rules, convergence results is characterized for a class of diminishing step sizes, providing an optimal choice of diminishing step sizes within this class.

Chapter 4 offers a novel distributed gradient method that uses a constant step size, and converges with a rate of $O(1/t)$, where t is the number of communication steps. We explain the Nesterov-type optimization acceleration technique, as well as the effect of using multiple communication steps per iteration, both of which are crucial to achieving this result.

In Chapter 5, we present results for numerical experiments on a machine learning benchmark dataset, verifying our theoretical analysis. Chapter 6 closes with conclu-

sions.

Chapter 2

Model

This chapter sets the stage for our distributed first-order methods. In Section 2.1, we review some basics in convex analysis, including a discussion on the proximal operator. Section 2.2 describes the class of distributed methods that is the subject of our studies, followed by Sections 2.3 and 2.4 with details on its optimization and consensus aspects, respectively. In Section 2.5, we explain the convergence rate notions that will be used to characterize the performance of our methods.

2.1 Preliminaries

In this section, we gather some notations, definitions, properties and concepts that are important to our work.

Vectors and Matrices

We begin by explaining our notations and recalling some basic definitions in linear algebra.

Superscripts are used for the iteration number of vectors in our methods, for example, $\overline{x}^k, \overline{w}^k, e^k$. Subscripts are used for the iteration number of scalars, such as $\alpha_k, \beta_k, \varepsilon_k$.

The standard inner product of two vectors $x, y \in \mathbb{R}^d$ is denoted $\langle x, y \rangle = x'y$. For

$x \in \mathbb{R}^d$, its Euclidean norm is $\|x\| = \sqrt{\langle x, x \rangle}$, and its 1-norm is $\|x\|_1 = \sum_{l=1}^{d} |x(l)|$, where $x(l)$ is its l-th entry.

For a matrix A, we denote its entry at the i-th row and j-th column as $[A]_{ij}$. We also write $[a_{ij}]$ to represent a matrix with $[A]_{ij} = a_{ij}$. A matrix is said to be *stochastic* if the entries in each row sum up to 1, and it is *doubly stochastic* if A and A' are both stochastic.

Properties of Functions

We now list some standard definitions and properties pertaining to the class of functions of our interest. Details and proofs can be found in [2, Appendices A-B].

- A function $f : \mathbb{R}^d \to (-\infty, \infty]$ is *convex* if, for any two points $x, y \in \mathbb{R}^d$, and any $t \in [0, 1]$, we have

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

- The effective domain of a function $f : \mathbb{R}^d \to [-\infty, \infty]$, denoted $\mathrm{dom}(f)$, is defined as

$$\mathrm{dom}(f) = \{x \in \mathbb{R}^d \mid f(x) < \infty\}.$$

  f is said to be *proper* $\mathrm{dom}(f)$ is nonempty and the restriction of f to $\mathrm{dom}(f)$ never attains $-\infty$. In other words, $f : \mathbb{R}^d \to [-\infty, \infty]$ is proper if $f(x) < \infty$ for at least one $x \in \mathbb{R}^d$ and $f(x) > -\infty$ for all $x \in \mathbb{R}^d$.

- A function $g : \mathbb{R}^d \to \mathbb{R}$ is *continuously differentiable* if its derivative exists and is continuous. It is *smooth* if it has derivatives of all orders. However, in the context of convex optimization, *nonsmooth* functions, usually refer to functions that do not even have a first-order derivative.

- A function $h : \mathbb{R}^d \to (-\infty, \infty]$ is *lower semi-continuous* if the set

$$\{x \in \mathbb{R}^d \mid h(x) \leq a\}$$

is closed for every $a \in \mathbb{R}$.

- A function $\nabla : \mathbb{R}^d \to \mathbb{R}^{d'}$ is called *Lipschitz-continuous* if there exists a constant $L > 0$, call the Lipschitz constant, such that

$$\|\nabla(x) - \nabla(y)\| \leq L\|x - y\|$$

for all $x, y \in \mathbb{R}^d$.

- If $f : \mathbb{R}^d \to (-\infty, \infty]$ is a proper convex function, then a vector $z \in \mathbb{R}^d$ is called the *subgradient* of f at point x if

$$f(y) \geq f(x) + \langle z, y - x \rangle$$

for all $y \in \mathbb{R}^d$. The set of all subgradients of f at x is called the *subdifferential* and is denoted as $\partial f(x)$.

For a continuously differentiable real-valued convex function, the subgradient coincides with its gradient, and we have the following characterization of convexity [2, Proposition B.3]:

Proposition 1. *(Convexity of Continuously Differentiable Functions)*

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function. Then f is convex if and only if

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle$$

for all $x, y \in \mathbb{R}^d$.

For a function with a Lipschitz-continuous gradient, we have the following well-known result [2, Proposition A.24]:

Proposition 2. *(Descent Lemma)*

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuously differentiable function whose gradient is Lipschitz-continuous with Lipschitz constant $L(f) > 0$, i.e.

$$\|\nabla f(x) - \nabla f(y)\| \leq L(f)\|x - y\|.$$

for every $x, y \in \mathbb{R}^d$. *Then for every* $L \geq L(f)$ *and* $x, y \in \mathbb{R}^d$,

$$f(x) \leq f(y) + \langle \nabla f(y), x - y \rangle + \frac{L}{2}\|x - y\|^2$$

Putting together both propositions above, we obtain the following useful expression:

Proposition 3. *(Convexity and Lipschitz-Continuous Gradient)*

Let $f : \mathbb{R}^d \to \mathbb{R}$ *be a continuously differentiable function whose gradient is Lipschitz-continuous with Lipschitz constant* L. *Then for every* $x, y, u \in \mathbb{R}^d$,

$$f(u) \geq f(x) + \langle \nabla f(y), u - x \rangle - \frac{L}{2}\|x - y\|^2.$$

Proof. This follows directly by summing up the following expressions:

$$f(u) \geq f(y) + \langle \nabla f(y), u - y \rangle \quad \text{(by Proposition 1)}$$
$$f(y) \geq f(x) - \langle \nabla f(y), x - y \rangle - \frac{L}{2}\|x - y\|^2 \quad \text{(by Proposition 2)}$$

□

The Proximal Operator

The proximal operator with respect to a function $f : \mathbb{R}^d \to (-\infty, \infty]$ is defined as

$$\text{prox}_f^\alpha(x) = \underset{z \in \mathbb{R}^d}{\text{argmin}} \left\{ f(z) + \frac{1}{2\alpha}\|z - x\|^2 \right\}.$$

In other words, the proximal operator represents the minimizer of the function f localized around the operand x. Note that $\text{prox}_f^\alpha(x) = \text{prox}_{\alpha f}^1(x)$, so when α is not specified, it is understood that $\text{prox}_f(x) = \text{prox}_f^1(x)$.

As an example, consider the case where $f(x) = \mathbf{1}_X(x)$ is the indicator function of

the set X, with $\mathbf{1}_X(x) = 0$ if $x \in X$ and $\mathbf{1}_X(x) = \infty$ if $x \notin X$. Then

$$\text{prox}_f^\alpha(x) = \begin{cases} x, & x \in X \\ \text{argmin}_{z \in X} \|z - x\|^2, & x \notin X \end{cases}$$

which is exactly the projection of x onto the set X. From this perspective, the proximal operator can be seen as a generalization of the projection operator. This is also why we're allowing the function f to be an extended real-value function.

The proximal operator has the following properties:

Proposition 4. *(Basic properties of the proximal operator)*

Let $y = prox_f^\alpha(x)$ for some function $f : \mathbb{R}^d \to (-\infty, \infty]$. Then

(a) $\frac{1}{\alpha}(x - y) \in \partial f(y)$.

(b) y can be written as $y = x - \alpha z$, where $z \in \partial f(y)$.

(c) For all $u \in \mathbb{R}^d$, $f(u) \geq f(y) + \frac{1}{\alpha}\langle x - y, u - y \rangle$.

Proof. (a) is clear from the observation that $0 \in \partial f(y) + \frac{1}{\alpha}(y - x)$ by the definition of the proximal operator, and (b) is simply an alternative expression of (a). (c) follows from (a) and the definition of a subgradient. □

Another useful property of the proximal operator is that, like the projection operator, it is nonexpansive:

Proposition 5. *(Nonexpansiveness of the proximal operator)*

Let $y = prox_f(x)$ and $\hat{y} = prox_f(\hat{x})$ for some function $f : \mathbb{R}^d \to (-\infty, \infty]$. Then

$$\|y - \hat{y}\| \leq \|x - \hat{x}\|.$$

Proof. By Proposition 4, we have

$$f(\hat{y}) \geq f(y) + \langle x - y, \hat{y} - y \rangle$$
$$f(y) \geq f(\hat{y}) + \langle \hat{x} - \hat{y}, y - \hat{y} \rangle$$

whose sum is

$$\langle x - y - \hat{x} + \hat{y}, \hat{y} - y \rangle \leq 0$$

which, by the Cauchy-Schwarz inequality, further implies

$$\|\hat{y} - y\|^2 \leq \langle x - \hat{x}, y - \hat{y} \rangle \leq \|x - \hat{x}\| \cdot \|y - \hat{y}\|.$$

Cancelling the nonnegative term $\|\hat{y} - y\|$ yields the desired result.

□

The proximal operator provides an alternative to the use of subgradients when optimizing nonsmooth functions. It involves finding the minimum argument $z \in \mathbb{R}^d$ for the expression $f(z) + \frac{1}{2\alpha}\|z - x\|^2$, which may be difficult to calculate. However, there are certain well-structured nonsmooth functions for which this can be computed easily, and in such cases, proximal methods may outperform subgradient methods. We illustrate this by an example; more examples and applications in signal processing can be found in [25, 2.2.2].

Consider the function $f : \mathbb{R} \to \mathbb{R}$, $f(x) = \lambda|x|$, with $\lambda > 0$, and let $y = \text{prox}_f^1(x)$. Then, by Proposition 4, we have

$$x - y \in \partial f(y) = \begin{cases} \{\lambda\}, & y > 0 \\ [-\lambda, \lambda], & y = 0 \\ \{-\lambda\}, & y < 0 \end{cases}$$

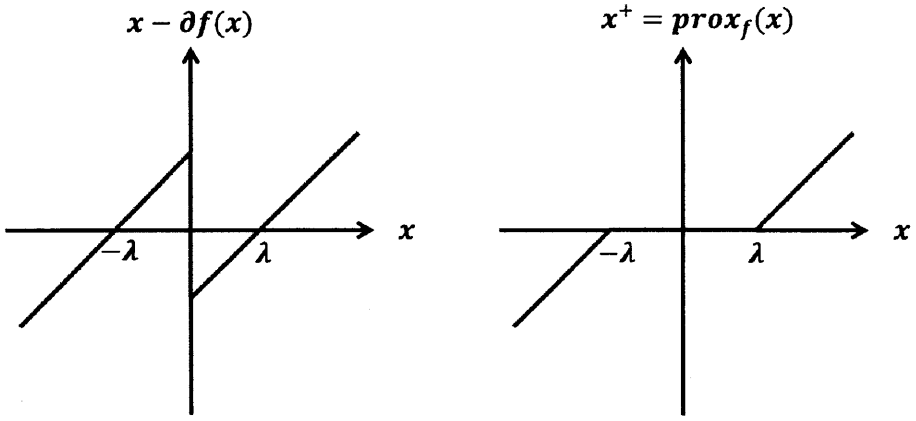
We can see that the first case, $x - y = \lambda$, happens only if $y > 0$; in other words, $y = x - \lambda$ if and only if $x - \lambda > 0$. Similarly, $y = x + \lambda$ if and only if $x + \lambda < 0$. Finally, $y = 0$ if and only if $x \in [-\lambda, \lambda]$. In summary,

$$\text{prox}_f^1(x) = \begin{cases} x - \lambda, & x > \lambda \\ 0, & -\lambda < x < \lambda \\ x + \lambda, & x < \lambda \end{cases}$$

which is simple to compute. It can be easily extended to the multi-dimensional one-norm $f : \mathbb{R}^d \to \mathbb{R}, f(x) = \lambda\|x\|_1$, which is a popular choice for regularization.

It is more effective to optimize the one-norm using the proximal operator rather than the subgradient. This is clear from comparing the following two methods for minimizing $f(x) = \lambda|x|$:

$$\begin{aligned} \text{(Subgradient method)} \quad & x^+ = x - z, z \in \partial f(x) \\ \text{(Proximal-point method)} \quad & x^+ = \text{prox}_f^1(x) \end{aligned}$$



(a) Subgradient Method (b) Proximal-point Method

Figure 2-1: Comparison between subgradient and proximal-point methods

Figure 2-1 illustrates the update map from the previous iterate x to the next iterate x^+. When the distance from x to the minumum 0 is greater than λ, both methods take a unit step towards 0. However, when x is within $[-\lambda, \lambda]$, the subgradient method overshoots and in some cases may not converge to 0, while the proximal-point method brings the iterate to 0 in the following iteration and keeps it there. For this reason, the proximal operator is preferred over the subgradient when optimizing well-structured functions like the one-norm.

The Proximal-Gradient Method

This thesis focuses on the proximal-gradient method, which combines the use of the gradient for the continuously differentiable part of the function, and the proximal operator for the nonsmooth part. Again, these methods are particularly favorable when the proximal operator for the nonsmooth part is easy to compute.

Formally, the proximal-gradient method seeks to optimize

$$\min_{x \in \mathbb{R}^d} f(x) = g(x) + h(x)$$

where $g : \mathbb{R}^d \to \mathbb{R}$ is convex, continuously differentiable, and has Lipschitz-continuous gradients with Lipschitz constant $L > 0$, and $h : \mathbb{R}^d \to (-\infty, \infty]$ is convex, proper, and lower-semicontinuous, but may not be differentiable.

The proximal-gradient method performs the following iterative updates:

$$x^k = \text{prox}_h^\alpha\{y^{k-1} - \alpha \nabla g(y^{k-1})\}$$

It is referred to as *basic* if $y^k = x^k$, and *accelerated* if $y^k = x^k + \frac{k-1}{k+2}(x^k - x^{k-1})$. α is the constant step size, chosen such that $\alpha \leq \frac{1}{L}$. The basic method converges with $f(x^k) - f(x^*) = O(1/k)$, while the accelerated method converges with rate $O(1/k^2)$ [25].

We highlight two properties of the proximal-gradient method that are useful for our anaylsis:

Proposition 6. *(Proximal-Gradient Method)*

Let

$$x^+ = prox_h^\alpha\{x - \alpha \nabla g(x)\}$$

with $\alpha \leq \frac{1}{L}$ *and* $f(x) = g(x) + h(x)$. *Then*

(a) x^+ *can be written as*

$$x^+ = x - \alpha\left(\nabla g(x) + z\right),$$

where $z \in \partial h(x^+)$.

(b) For every $u \in \mathbb{R}^d$,

$$f(x^+) - f(u) \leq \frac{1}{2\alpha}\left[\|x-u\|^2 - \|x^+ - u\|^2\right].$$

Proof. Part (a) follows directly from Proposition 4(b). As for (b), by Proposition 3, we have

$$g(u) \geq g(x^+) + \langle \nabla g(x), u - x^+ \rangle - \frac{L}{2}\|x^+ - x\|^2$$

and by Proposition 4(c), we have

$$h(u) \geq h(x^+) + \frac{1}{\alpha}\langle x - \alpha\nabla g(x) - x^+, u - x^+ \rangle$$

Summing up the two expressions above, and noting that $\frac{1}{2\alpha} \geq \frac{L}{2}$,

$$f(u) \geq f(x^+) + \frac{1}{\alpha}\langle x - x^+, u - x^+ \rangle - \frac{1}{2\alpha}\|x^+ - x\|^2$$

Recall that for any two vectors $a, b \in \mathbb{R}^d$, $2\langle a, b\rangle - \|a^2\| = \|b^2\| - \|a-b\|^2$. Therefore,

$$f(u) \geq f(x^+) + \frac{1}{2\alpha}\left[\|u - x^+\|^2 - \|u - x\|^2\right].$$

Rearranging terms yields the desired expression. □

2.2 Distributed Proximal-Gradient Methods

Consider a network of m agents. For each $i = 1, ..., m$, agent i has the local objective function

$$f_i(x) = g_i(x) + h_i(x)$$

where $g_i : \mathbb{R}^d \to \mathbb{R}$ is a convex, continuously differentiable function whose gradient is Lipschitz-continuous, and $h_i : \mathbb{R}^d \to (-\infty, \infty]$ is a lower-semicontinuous proper convex function that is not necessarily differentiable.

The goal of our method is to solve the optimization problem

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{m}\sum_{i=1}^{m} f_i(x) \tag{2.1}$$

$f(x)$ is called the global objective function. Its optimal value, denoted by f^*, is assumed to be finite and attained at a unique x^*.

We propose a class of first-order methods that solves the optimization problem in a distributed fashion, where each agents in the network maintains an iterate that is an estimation of the global optimum x^*, using only its private objective, and the iterates of its neighbors in the network. Agent i updates its iterate as follows:

$$\begin{cases} x_i^k = \text{prox}_{h_i}^{\alpha_k}\left\{w_i^{k-1} - \alpha_k \nabla g_i(w_i^{k-1})\right\} & \text{(optimization)} \\ w_i^k = \sum_{j=1}^m \lambda_{ij}^k y_j^k & \text{(consensus)} \end{cases} \tag{2.2}$$

where $x_i^k, y_i^k, w_i^k \in \mathbb{R}^d$ are vectors at agent i in iteration k, $\alpha_k > 0$ is the step size, and $\lambda_{ij}^k \in [0,1]$ are weights.

There are two stages in this method:

- The *optimization stage* performs a standard proximal gradient step with the chosen step size α_k. Note that if $h_i(x) = 0$ for each i, then the method reduces to the gradient method.

  When $y_i^k = x_i^k$, it is called the *basic* method, and when $y_i^k = x_i^k + \frac{k-1}{k+2}(x_i^k - x_i^{k-1})$, it is called the *accelerated* method.

- In the *consensus stage*, each agent communicates through the network to exchange iterates with its neighbors. Communication may happen either only once, which we call the *single-step consensus* scheme, or many times, which we call *multi-step consensus.*

  The result is modelled as a linear combination of its own iterate and that of its neighbors, and the weights λ_{ij}^k are determined both by the communication network and by the single- or multi-step consensus scheme.

In practice, each agent only needs to maintains either y_i^k or w_i^k at any given time; however, for clarity of analysis, we distinguish results of the optimization and consensus stages by denoting them as y_i^k and w_i^k, respectively.

2.3 Conditions on Objective Functions

We assume that the functions of interest have the following properties:

Assumption 1. *For every i,*

(a) $g_i : \mathbb{R}^d \to \mathbb{R}$ *is convex, continuously differentiable, and has a Lipschitz-continuous gradient with Lipschitz constant $L > 0$, i.e.* $\|\nabla g_i(x) - \nabla g_i(y)\| \leq L\|x - y\|$.

(b) $h_i : \mathbb{R}^d \to (-\infty, \infty]$ *is convex, proper, and lower-semicontinuous.*

(c) (Bounded gradients and subgradients) There exists a scalar G such that for all x, $\|\nabla g_i(x)\| < G$, *and* $\|z\| < G$ *for each subgradient $z \in \partial h(x)$.*

(d) (Uniqueness of optimum) $f(x) = \frac{1}{m}\sum_{i=1}^m f_i(x) = \frac{1}{m}\sum_{i=1}^m g_i(x) + h_i(x)$ *attains its minimum at a unique x^*.*

Note that Assumption 1 implies that $g(x) = \frac{1}{m}\sum_{i=1}^m g_i(x)$ and $h(x) = \frac{1}{m}\sum_{i=1}^m h_i(x)$ also satisfy the same assumptions.

While $(a), (b)$ and (d) are standard assumptions also common to centralized first-order methods, (c) requires more justification. We now illustrate, by an example, that distributed first-order methods given in (2.2) may not converge when the (sub)gradient is not bounded. In particular, we construct a case with unbounded gradients for the distributed gradient method, in which the estimates $||x_i^k|| \to \infty$ as $k \to \infty$ for each agent i, while y^k stays at the minimum of $g(x) = \frac{1}{m}\sum_i g_i$. This occurs despite using a diminishing step size $\alpha_k = \frac{1}{k}$.

We consider the most simple case with only two agents i, j, whose private functions are differentiable and symmetric to each other with respect of the y-axis, i.e. $g_i(x) = g_j(-x), h_i(x) = h_j(x) = 0$. If we also choose a symmetric weight matrix A and symmetric initial points $x_i^0 = -x_j^0$, then we would have $x_i^k = -x_j^k$ and average iterate

$\overline{x}^k = \frac{x_i^k + x_j^k}{2} = 0$ at every iteration k. We choose $[\lambda_{ij}^k = \begin{bmatrix} 1/3 & 2/3 \\ 2/3 & 1/3 \end{bmatrix}$ and $x_i^0 = -x_j^0 = -1$. Then

$$\begin{bmatrix} w_i^k \\ w_j^k \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} x_i^k \\ x_j^k \end{bmatrix} = \begin{bmatrix} -\frac{1}{3}x_i^k \\ -\frac{1}{3}x_j^k \end{bmatrix} \text{ for each } k \geq 1.$$

Next, we construct a function for which x_i decreases with each step at a fixed interval. In particular, we aim at $x_i^1 = -2, x_i^2 = -3, ..., x_i^k = -(k+1)$, which, by the the previous formula, implies $w_i^{k-1} = \frac{k}{3}$. We wish to have $x_i^k = w_i^{k-1} - \frac{1}{k}\nabla g_i(w_i^{k-1})$, or $\nabla g_i(\frac{k}{3}) = k(\frac{k}{3} + (k+1))$. Therefore, if we set

$$\nabla g_i(x) = 3x(4x+1) = 12x^2 + 3x \text{ for } x \geq 1$$

and calculate g_j correspondingly, then we would end up with the desired divergent sequence $x_i^k = -(k+1) \to -\infty, x_j^k \to \infty$ as $k \to \infty$.

This gives $\nabla^2 g_i(x) = 24x + 3$, so it is convex on $[-\frac{1}{8}, \infty]$. On the other hand, integration gives $g_i(x) = 4x^3 + \frac{3}{2}x^2$, where we have taken the integration constant to be zero for simplicity.

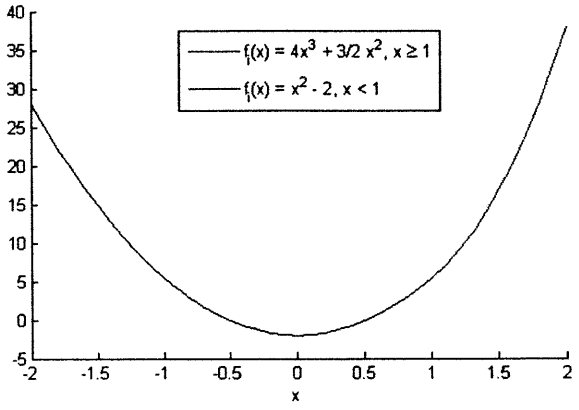
Finally, choose $g_i(x), x < 1$ and $g_j(x), x > -1$ so that they are convex and $y^k = 0$ is the minimum of the global function $g_i + g_j$. We simply paste a quadratic function of the form $ax^2 + c$ to the left of $x = 1$ to ensure that the minimum is at 0. Using zero- and first-order information at $x = 1$,

$$\begin{aligned} \nabla g_i(1) &= 15 = 2a \\ g_i(1) &= \frac{11}{2} = a + c \\ \Rightarrow \quad a &= \frac{15}{2}, c = -2 \end{aligned}$$

Therefore,

$$g_i(x) = \begin{cases} 4x^3 + \frac{3}{2}x^2, & x \geq 1 \\ \frac{15}{2}x^2 - 2, & x < 1 \end{cases}$$

as plotted below. $g_j(x) = g_i(-x)$ can be constructed easily.



We have thus obtained smooth convex functions for which the distributed gradient method does not converge with diminishing step sizes $\alpha_k = \frac{1}{k}$. Indeed, without a bound on the gradients, there are instances where the distributed method fails to converge, because there are functions whose (sub)gradients grow faster than the rate at which the step size diminishes. The most straightforward fix to this is to assume that the gradients are bounded.

2.4 Network Communication and Consensus

For the consensus stage of (2.2), we adopt the information exchange model developed in [10, 13], which we summarize in this section. In the model, the agents form a communication network, and in the communication step at time t, every agent takes a linear combination of other agents' estimates according to a weight matrix $A(t) = [a_{ij}(t)]$. While the weight matrix may change over time, the following conditions should always be satisfied:

Assumption 2. *(Communication Network Requirements)*

Consider the weight matrices $A(t) = [a_{ij}(t)], t = 1, 2, ...$

(a) (Significant weights) For every t, $A(t)$ *is doubly stochastic. Moreover, there exists a scalar* $\eta \in (0, 1)$ *such that for all* i, $a_{ii}(t) \geq \eta$, *and for* $j \neq i$, *either* $a_{ij}(t) = 0$,

in which case j is not a neighbor of i at time t, or $a_{ij}(t) \geq \eta$, in which case j is a neighbor of i and receives the estimate of i at time t.

(b) (Connectivity and bounded intercommunication interval) Let

$$E_t = \{(j,i) \mid j \text{ receives the estimate of } i \text{ at time } t\},$$
$$E_\infty = \{(j,i) \mid j \text{ receives the estimate of } i \text{ for infinitely many } t\}.$$

Then E_∞ is connected. Moreover, there exists an integer $B \geq 1$ such that if $(j,i) \in E_\infty$, then $(j,i) \in E_t \cup E_{t+1} \cup ... \cup E_{t+B-1}$.

In this assumption, part (a) ensures that each agent maintains an equal influence on and by others in the network. It also guarantees the significance of every estimate received by an agent. On the other hand, part (b) states that the overall communication network is capable of passing information from an agent to any other agent in bounded time. As a result, if we take $\overline{B} = (m-1)B$, then $E_t \cup E_{t+1} \cup ... \cup E_{t+\overline{B}-1} = E_\infty$.

We then have the following result from [13]:

Lemma 1. *[13, Proposition 1(b)] Let Assumption 2 hold, and let*

$$\Phi(t,s) = A(s)A(s+1)\cdots A(t-1)A(t).$$

Then the entries $[\Phi(t,s)]_{ij}$ converges to $\frac{1}{m}$ as $t \to \infty$ with a geometric rate uniformly with respect to i,j. Specifically, for all $i,j \in \{1,...,m\}$ and all t,s with $t \geq s$,

$$\left|[\Phi(t,s)]_{ij} - \frac{1}{m}\right| \leq 2\frac{1+\eta^{-\overline{B}}}{1-\eta^{\overline{B}}}\left(1-\eta^{\overline{B}}\right)^{\frac{t-s}{\overline{B}}}$$

For simplicity, we shall denote $\Gamma = 2\frac{1+\eta^{-\overline{B}}}{1-\eta^{\overline{B}}}$, $\gamma = \left(1-\eta^{\overline{B}}\right)^{\frac{1}{\overline{B}}}$, $0 < \beta < 1$, and quote this theorem as

$$\left|[\Phi(t,s)]_{ij} - \frac{1}{m}\right| \leq \Gamma\gamma^{k-s} \tag{2.3}$$

The above lemma implies that the distance between each local iterate and the

average iterate decreases geometrically with respect to the number of communication steps taken in the consensus stage. This result is crucial for our analysis later on, when showing convergence of the iterates to the average iterate.

2.5 Convergence Rate Notions

In this section, we clarify the distinction between *exact convergence* and *convergence to an error neighborhood*, and specify two possible convergence rate notions used to describe the latter.

Traditionally, the rate of convergence is characterized in terms of the number of iterations required to reach an ϵ-optimal solution. $x \in \mathbb{R}^d$ is said to be an *ϵ-optimal solution* of the function $f : \mathbb{R}^d \to (-\infty, \infty]$ if

$$f(x) - f^* \leq \epsilon.$$

For example, suppose that for a given method, the function values converge to the optimal value with a bound of

$$f(x^n) - f^* \leq \frac{D_0}{n} \tag{2.4}$$

for some scalar $D_0 > 0$, where n is the number of iterations. Setting the right-hand side to ϵ, we see that it takes $n = \frac{D_0}{\epsilon}$ iterations to find an ϵ-optimal solution. In this case, we say that the method *converges exactly*, and the convergence rate is $O(1/n)$.

Another type of convergence, which usually arises in methods that use a constant step size, involves terms that depend on the step size α and do not diminish as n increases. As an example, consider the bound

$$f(x^n) - f^* \leq \frac{D_1}{\alpha n^2} + D_2\alpha \tag{2.5}$$

for some positive scalars D_1, D_2. Under the traditional ϵ-optimality convergence rate characterization, if we fix the number of iterations n that this method is allowed to

run, then by choosing $\alpha = \sqrt{\frac{D_1}{n^2 D_2}}$ to minimize $\frac{D_1}{\alpha n^2} + D_2\alpha$, the bound becomes

$$f(x^n) - f^* \leq \frac{\sqrt{D_1 D_2}}{n}.$$

Therefore, with a budget of n iterations, the best solution we can achieve is an $\frac{\sqrt{D_1 D_2}}{n}$-optimal solution. In other words, to reach an ϵ-optimal solution, at least $n = \frac{\sqrt{D_1 D_2}}{\epsilon}$ iterations are required. In this case, since $\epsilon = \frac{\sqrt{D_1 D_2}}{n}$, we say that the convergence rate of this method is $O(1/n)$.

While this conventional convergence rate notion provides a common basis for comparison between methods that converge exactly and methods that converge to an error neighborhood, note that it does not differentiate between the rates of (2.4) and (2.5), although they are quite different. The expression (2.4) does not require fixing a budget of iterations in advance so as to find the optimal constant step size, and it approaches the optimal solution as the method continues to run for more iterations. On the other hand, once the constant step size is fixed, (2.5) does not reach the optimal solution even if the method continues for more iterations; however, in the early stages, the rate of decrease in function value is in effect $1/n^2$, and it may outperform than (2.4), until (2.4) decreases beyond the error neighborhood of (2.5).

Therefore, as an alternative interpretation of convergence rate in the latter case, it is often helpful explicitly state the rate at which the error neighborhood is being reached. For example, (2.5) is also said to (2.5) *converge to an error neighborhood* $D_2\alpha$ *with rate* $O(1/n^2)$. In the upcoming chapters, it should be clear from the context which interpretation is being used.

Chapter 3

Distributed First-Order Methods with Single-Step Consensus

In this chapter, we consider the distributed method where only one communication step is taken at each iteration. This is called the "single-step consensus" scheme, introduced in Section 3.1. In Section 3.2, we show that the basic method converges with rates $O(1/\sqrt{n})$ for a constant step size, and $O(\log n/\sqrt{n})$ for a class of diminishing step sizes. Section 3.3 outlines challenges for using Nesterov's technique to accelerate distributed methods with single-step consensus.

3.1 Introduction

Recall from Section 2.2 that the distributed proximal-gradient method solves

$$\min_{x\in\mathbb{R}^d} f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x) = \frac{1}{m}\sum_{i=1}^{m} g_i(x) + h_i(x)$$

by iteratively performing the following:

$$\begin{cases} x_i^k = \text{prox}_{h_i}^{\alpha_k}\left\{w_i^{k-1} - \alpha_k \nabla g_i(w_i^{k-1})\right\} & \text{(optimization)} \\ w_i^k = \sum_{j=1}^{m} a_{ij}^k y_j^k & \text{(consensus)} \end{cases} \tag{3.1}$$

where $y_i^k = x_i^k$ is call the *basic* method, and $y_i^k = x_i^k + \frac{k-1}{k+2}(x_i^k - x_i^{k-1})$ is called the *accelerated* method. With single-step consensus, communication only happens once in the consensus stage, and the weight matrices $[a_{ij}^k]$ satisfy Assumption 2. The method is initialized with $\{w_i^0\}_{i=1}^m$.

For a constant step size, [13] showed that the basic subgradient method (with $y_i^k = x_i^k$) converges with rate $O(1/\sqrt{k})$. We shall see that although the proximal-gradient method makes better use of function properties in the optimization stage, it suffers from the same bottleneck due to consensus. Therefore, the convergence rate is similar to that of the subgradient method. We also extend the analysis to arbitrary step sizes and characterize the result in terms of step size choices. In particular, for the class of diminishing step sizes $\alpha_k = 1/k^a, a > 0$, we show that the best exact convergence rate is achieved when $a = 1/2$.

On the other hand, in the accelerated case (with $y_i^k = x_i^k + \frac{k-1}{k+2}(x_i^k - x_i^{k-1})$), there are instances of this problem that my not converge, due to the distributed nature of this problem, and the sensitivity of the accelerated method to error. In order to take advantage of the acceleration technique, we have to control the quality of consensus by using multiple consensus steps, which will be discussed in the next chapter.

3.2 Convergence Rate of the Basic Method

In order to derive the convergence rate of the basic proximal-gradient method, we first provide an upper bound on consensus of iterates, using properties of network communication. This is then used to derive the convergence rate, given in terms of step sizes. Finally, we characterize the rate according to the step size rule.

3.2.1 Consensus of Iterates

Recall that the goal for each agent is to maintain an estimate of the global optimum x^*. If this is achieved, then the local estimates must be equal to each other, because by Assumption 1(d), x^* is unique. However, since each agent moves its iterate in a different direction according to the gradient and subgradient of its private function,

the question arises as to whether local iterates will actually converge to the same point. In other words, is the single communication step in the consensus stage strong enough to "pull" the iterates close to each other?

The following lemma addresses this question:

Lemma 2. *(Consensus of iterates, limited consensus) In Algorithm* (3.1),

$$\left\|x_i^k - \overline{x}^k\right\| \leq \Gamma\gamma^{k-1}\sum_{j=1}^{m}\left\|w_j^0\right\| + 2m\Gamma G\sum_{r=0}^{k-1}\gamma^{k-r-1}\alpha_r + 4\alpha_k G$$

where $\overline{x}^k = \frac{1}{m}\sum_{i=1}^{m} x_i^k$, Γ *and* γ *are given in Lemma 1, and* G *is the bound on the gradient of* g_i *and the subgradients of* h_i *for every* i, *as in Assumption 1.*

Proof. By Proposition 6, (3.1) can be written as

$$x_i^k = w_i^{k-1} - \alpha_k\left[\nabla g_i(w_i^{k-1}) + z_i^k\right] \tag{3.2}$$

where $z_i^k \in \partial h_i(x_i^k)$. Since $w_i^{k-1} = \sum_{j=1}^{m} a_{ij}^{k-1}x_i^{k-1}$ due to the consensus stage, we can write (3.2) recursively:

$$\begin{aligned}
x_i^k &= \sum_{j=1}^{m} a_{ij}^{k-1}x_i^{k-1} - \alpha_k\left[\nabla g_i(w_i^{k-1}) + z_i^k\right] \\
&= \sum_{j=1}^{m}[\Phi(k-1,k-2)]_{ij}x_i^{k-2} - \sum_{j=1}^{m} a_{ij}^{k-1}\alpha_{k-1}\left[\nabla g_j(w_j^{k-2}) + z_j^{k-1}\right] - \alpha_k\left[\nabla g_i(w_i^{k-1}) + z_i^k\right] \\
&= \cdots \\
&= \sum_{j=1}^{m}[\Phi(k-1,0)]_{ij}w_j^0 - \sum_{r=0}^{k-1}\sum_{j=1}^{m}[\Phi(k-1,r)]_{ij}\alpha_r\left[\nabla g_j(w_j^{r-1}) + z_j^r\right] - \alpha_k\left[\nabla g_i(w_i^{k-1}) + z_i^k\right]
\end{aligned} \tag{3.3}$$

Taking the average, we have

$$\begin{aligned}\overline{x}^k &= \frac{1}{m}\sum_{i=1}^{m} x_i^k \\ &= \sum_{j=1}^{m}\frac{1}{m}w_j^0 - \sum_{r=0}^{k-1}\sum_{j=1}^{m}\frac{1}{m}\alpha_r\left[\nabla g_j(w_j^{r-1}) + z_j^r\right]\frac{1}{m}\sum_{i=1}^{m}\alpha_k\left[\nabla g_i(w_i^{k-1}) + z_i^k\right] \qquad (3.4)\end{aligned}$$

where we used the fact that $\sum_{i=1}^{m}[\Phi(t,s)]_{ij} = 1$, since $\Phi(t,s)$ is doubly stochastic for all $s \leq t$.

Subtracting (3.4) from (3.3),

$$\begin{aligned}x_i^k - \overline{x}^k = &\sum_{j=1}^{m}\left[[\Phi(k-1,0)]_{ij} - \frac{1}{m}\right]w_j^0 - \sum_{r=0}^{k-1}\sum_{j=1}^{m}\left[[\Phi(k-1,r)]_{ij} - \frac{1}{m}\right]\alpha_r\left[\nabla g_j(w_j^{r-1}) + z_j^r\right] \\ &- \frac{m-1}{m}\alpha_k\left[\nabla g_i(w_i^{k-1}) + z_i^k\right] + \frac{1}{m}\sum_{j\neq i}\alpha_k\left[\nabla g_j(w_i^{k-1}) + z_j^k\right]\end{aligned}$$

Finally, taking the norm,

$$\begin{aligned}\left\|x_i^k - \overline{x}^k\right\| \leq &\sum_{j=1}^{m}\left\|[\Phi(k-1,0)]_{ij} - \frac{1}{m}\right\| \cdot \left\|w_j^0\right\| \\ &+ \sum_{r=0}^{k-1}\sum_{j=1}^{m}\left\|[\Phi(k-1,r)]_{ij} - \frac{1}{m}\right\|\alpha_r\left[\left\|\nabla g_j(w_j^{r-1})\right\| + \left\|z_j^r\right\|\right] \\ &+ \frac{m-1}{m}\alpha_k\left(\left\|\nabla g_i(w_i^{k-1})\right\| + \left\|z_i^k\right\|\right) + \frac{1}{m}\sum_{j\neq i}\alpha_k\left(\left\|\nabla g_j(w_i^{k-1})\right\| + \left\|z_j^k\right\|\right) \\ &\leq \Gamma\gamma^{k-1}\sum_{j=1}^{m}\left\|w_j^0\right\| + \sum_{r=0}^{k-1} m\Gamma\gamma^{k-r-1}2\alpha_r G + \frac{2(m-1)}{m}\cdot 2\alpha_k G\end{aligned}$$

where in the last line we used Lemma 1 and Assumption 1 (c), respectively, to bound terms of the form $\left\|[\Phi(k-1,r)]_{ij} - \frac{1}{m}\right\|$ and $\left\|\nabla g_j(w_j^{r-1})\right\| + \left\|z_j^r\right\|$. This implies the desired result.

□

The key idea behind this proof is that without the optimization stage, the algorithm is in effect an averaging algorithm, and the iterates $x_i^k = w_i^k$ converge to the mean $\overline{x}^k$. In light of this, we treat the optimization stage as a process that introduces

error to the averaging process, and study its affect on the averaging algorithm. By (3.2), the error that occurs at each stage is $\left\|w_i^{k-1} - x_i^k\right\| = \left\|\nabla g_i(w_i^{k-1}) + z_i^k\right\| \leq 2\alpha_k G$. From Lemma 2, it is clear that the error which happened $k-r$ iterations ago is attenuated geometrically by γ^{k-r}, in line with our understanding of the consensus algorithm. This leads to the bound above.

Clearly, the choice of step sizes has a significant effect on the accumulated error. For example, if we use a constant step size, then the average error over all iterations is actually finite:

Corollary 1. *(Average consensus for a constant step size)*

Let $\alpha_k = \alpha$ in Algorithm (3.1). Then there exists scalars C_1, C_2 such that for $n = 1, 2, ...$

$$\frac{1}{n}\sum_{k=1}^{n}\left\|x_i^k - \overline{x}^k\right\| \leq \frac{C_1'}{n} + C_2'$$

Proof. This is obtained simply by averaging the statement of Lemma 2 over $k = 1, ..., n$:

$$\begin{aligned}\frac{1}{n}\sum_{k=1}^{n}\left\|x_i^k - \overline{x}^k\right\| &\leq \frac{1}{n}\left[\Gamma\sum_{k=1}^{n}\gamma^{k-1}\sum_{j=1}^{m}\left\|w_j^0\right\| + 2\alpha m\Gamma G\sum_{k=1}^{n}\sum_{r=0}^{k-1}\gamma^{k-r-1} + \sum_{k=1}^{n}4\alpha G\right] \\ &\leq \frac{1}{n}\left(\frac{\Gamma}{1-\gamma}\sum_{j=1}^{m}\left\|w_j^0\right\|\right) + 2\alpha m\Gamma G\frac{1}{1-\gamma} + 4\alpha G\end{aligned}$$

Note that the given initial points $\{w_j^0\}_{j=1}^m$ can be treated as known constants. Then the corollary is obtained with

$$\begin{aligned}C_1' &= \frac{\Gamma}{1-\gamma}\sum_{j=1}^{m}\left\|w_j^0\right\| \\ C_2' &= 2\alpha m\Gamma G\frac{1}{1-\gamma} + 4\alpha G\end{aligned}$$

□

More generally, when the step size is not constant, the error can be expressed in terms of the step sizes. The following lemma gives the total error in n iterations, in

the form that will be applied directly in the following section.

Corollary 2. *(Accumulated consensus of iterates for arbitrary step sizes)*

In Algorithm (3.1)*, there exists scalars* C_1, C_2 *such that for all iterations* $n = 1, 2, ...,$

$$\sum_{k=1}^{n} \alpha_k \left\| x_i^k - \overline{x}^k \right\| \leq C_1 + C_2 \sum_{i=1}^{m} {\alpha_k}^2 \tag{3.5}$$

Proof. The accumulated consensus, weighed by step sizes α_k, is

$$\sum_{k=1}^{n} \alpha_k \left\| x_i^k - \overline{x}^k \right\| \leq \Gamma \left(\sum_{j=1}^{m} \left\| w_j^0 \right\| \right) \sum_{k=1}^{n} \alpha_k \gamma^{k-1} + 2m\Gamma G \sum_{k=1}^{n} \alpha_k \sum_{r=0}^{k-1} \gamma^{k-r-1} \alpha_r + 4G \sum_{k=1}^{n} {\alpha_k}^2 \tag{3.6}$$

To separate the product terms, we use the fact that $ab \leq \frac{1}{2}(a^2 + b^2)$. Therefore,

$$\begin{aligned} \sum_{k=1}^{n} \alpha_k \gamma^{k-1} &\leq \frac{1}{2} \sum_{k=1}^{n} {\alpha_k}^2 + \frac{1}{2} \sum_{k=1}^{n} \gamma^{2(k-1)} \\ &\leq \frac{1}{2} \sum_{k=1}^{n} {\alpha_k}^2 + \frac{1}{2(1-\gamma^2)} \end{aligned} \tag{3.7}$$

$$\begin{aligned} \sum_{k=1}^{n} \sum_{r=0}^{k-1} \gamma^{k-r-1} \alpha_k \alpha_r &\leq \sum_{k=1}^{n} \sum_{r=0}^{k-1} \gamma^{k-r-1} \frac{1}{2} \left({\alpha_k}^2 + \alpha_r^2 \right) \\ &= \frac{1}{2} \sum_{k=1}^{n} {\alpha_k}^2 \sum_{r=0}^{k-1} \gamma^{k-r-1} + \frac{1}{2} \sum_{r=0}^{n-1} \alpha_r^2 \sum_{k=r+1}^{n} \gamma^{k-r-1} \\ &\leq \frac{1}{2(1-\gamma)} \sum_{k=1}^{n} {\alpha_k}^2 + \frac{1}{2(1-\gamma)} \sum_{r=0}^{n-1} \alpha_r^2 = \frac{1}{1-\gamma} \sum_{k=1}^{n} {\alpha_k}^2 \end{aligned} \tag{3.8}$$

where in the second-to-last line we swapped the order of k and r, and in the last line we used the definition $\alpha_k = 0$ for $k \leq 0$ and $\alpha_k > 0$ for $k > 0$.

Substituting (3.7) and (3.8) into (3.6),

$$\sum_{k=1}^{n} \alpha_k \left\| x_i^k - \overline{x}^k \right\| \leq \Gamma \left(\sum_{j=1}^{m} \left\| w_j^0 \right\| \right) \left[\frac{1}{2} \sum_{k=1}^{n} {\alpha_k}^2 + \frac{1}{2(1-\gamma^2)} \right] + 2m\Gamma G \cdot \frac{1}{1-\gamma} \sum_{k=1}^{n} {\alpha_k}^2 + 4G \sum_{k=1}^{n} {\alpha_k}^2$$

Gathering terms gives the statement in the corollary with

$$C_1 = \frac{\Gamma\left(\sum_{j=1}^m \left\|w_j^0\right\|\right)}{2(1-\gamma^2)}$$
$$C_2 = \frac{\Gamma\left(\sum_{j=1}^m \left\|w_j^0\right\|\right)}{2} + \frac{2m\Gamma G}{1-\gamma} + 4G$$

□

3.2.2 Convergence Rate Analysis

With consensus results from Corollary 2, we are now ready to evaluate the quality of a local iterate x_j^k with respect to the global function, $f(x) = \frac{1}{m}\sum_{i=1}^m f_i(x) = \frac{1}{m}\sum_{i=1}^m g_i(x) + h_i(x)$. The result is summarized in the following theorem:

Theorem 1. *(Convergence rate of the basic distributed proximal-gradient method)*

Algorithm (3.1)*, with step sizes* $\alpha_k \leq \frac{1}{L}$*, maintains local estimates* x_j^k *with*

$$f_j^n - f^* \leq \frac{D_1 + D_2 \sum_{k=1}^n {\alpha_k}^2}{\sum_{k=1}^n \alpha_k}$$

where $f_j^n = \min_{1\leq k\leq n} f(x_j^k)$ *is the best global function value obtained so far,* f^* *is the optimal global function value, and* D_1, D_2 *are scalars.*

Proof. Before looking at the global function $f(x) = \frac{1}{m}\sum_{i=1}^m f_i(x)$, we first consider the private function value of the local estimate, $f_i(x_i^k) = g_i(x_i^k) + h_i(x_i^k)$. This part is the standard analysis inherited from the centralized proximal gradient method. Indeed, by Proposition 6 (c),

$$\begin{aligned} f_i(u) &\geq f_i(x_i^k) + \frac{1}{\alpha_k}\left\langle w_i^{k-1} - x_i^k, u - x_i^k\right\rangle - \frac{1}{2\alpha_k}\left\|x_i^k - w_i^{k-1}\right\|^2 \\ &= f_i(x_i^k) + \frac{1}{2\alpha_k}\left[\left\|x_i^k - u\right\|^2 - \left\|w_i^{k-1} - u\right\|^2\right] \end{aligned}$$

Since the squared-norm function is convex and $w_i^{k-1} = \sum_{j=1}^m a_{ij}^{k-1} x_i^{k-1}$ is a convex combination of $\{x_j^{k-1}\}_{j=1}^m$, we have

$$\sum_{i=1}^m \left\| w_i^{k-1} - u \right\| \leq \sum_{j=1}^m \left\| x_j^{k-1} - u \right\|.$$

Therefore, averaging the previous expression over all i and taking $u = x^*$ to be the global optimum, we have

$$\frac{\alpha_k}{m} \sum_{i=1}^m \left[f_i(x_i^k) - f_i(x^*) \right] \leq \frac{1}{2m} \sum_{i=1}^m \left\| x_i^{k-1} - x^* \right\|^2 - \frac{1}{2m} \sum_{i=1}^m \left\| x_i^k - x^* \right\|^2 \tag{3.9}$$

Note that the right-hand side of (3.9) is handy for recursive sums. Indeed, summing (3.9) over $k = 1, .., n$ yields

$$\sum_{k=1}^n \frac{\alpha_k}{m} \sum_{i=1}^m \left[f_i(x_i^k) - f_i(x^*) \right] \leq \frac{1}{2m} \sum_{i=1}^m \left\| x_i^0 - x^* \right\|^2 - \frac{1}{2m} \sum_{i=1}^m \left\| x_i^n - x^* \right\|^2 \leq \frac{1}{2m} \sum_{i=1}^m \left\| x_i^0 - x^* \right\|^2 \tag{3.10}$$

Returning to the global objective function, $f(x) = \frac{1}{m} \sum_{i=1}^m f_i(x)$, we wish to evaluate the performance of x_j^k with respect to it. However, since x_j^k is not optimized according to $f_i(x)$ for $i \neq j$, the best we can do is to bound $f_i(x_j^k)$ with $f_i(x_i^k)$ using convexity:

$$f_i(x_i^k) \geq f_i(x_j^k) + \left\langle \nabla f_i(x_j^k), x_i^k - x_j^k \right\rangle \geq f_i(x_j^k) - G \left\| x_i^k - x_j^k \right\| \tag{3.11}$$

Substituting (3.11) into (3.10) bounds the global function value at x_j^k:

$$\begin{aligned} \sum_{k=1}^n \alpha_k \left[f(x_j^k) - f^* \right] &= \sum_{k=1}^n \frac{\alpha_k}{m} \sum_{i=1}^m \left[f_i(x_j^k) - f_i(x^*) \right] \\ &\leq \sum_{k=1}^n \frac{\alpha_k}{m} \sum_{i=1}^m \left[f_i(x_i^k) + G \left\| x_i^k - x_j^k \right\| - f_i(x^*) \right] \\ &\leq \frac{1}{2m} \sum_{i=1}^m \left\| x_i^0 - x^* \right\|^2 + \frac{G}{m} \sum_{k=1}^n \alpha_k \sum_{i=1}^m \left\| x_i^k - x_j^k \right\| \end{aligned} \tag{3.12}$$

The left-hand side of (3.12) is bounded below by

$$\left(\sum_{k=1}^{n}\alpha_k\right)\left[f_j^n - f^*\right] \leq \sum_{k=1}^{n}\alpha_k\left[f(x_j^k) - f^*\right] \tag{3.13}$$

by the definition $f_j^n = \min_{1\leq k\leq n} f(x_j^k)$.

On the other hand, concerning the final term in (3.12), note that

$$\left\|x_i^k - x_j^k\right\| \leq \left\|x_i^k - \overline{x}^k\right\| + \left\|x_j^k - \overline{x}^k\right\|,$$

which leads to

$$\frac{1}{m}\sum_{i=1}^{m}\left\|x_i^k - x_j^k\right\| \leq \left\|x_j^k - \overline{x}^k\right\| + \frac{1}{m}\sum_{i=1}^{m}\left\|x_i^k - \overline{x}^k\right\|.$$

Also, recall that by Lemma 2 we have, for all i,

$$\sum_{k=1}^{n}\alpha_k\left\|x_i^k - \overline{x}^k\right\| \leq C_1 + C_2\sum_{k=1}^{n}{\alpha_k}^2$$

Therefore, utting these together,

$$\begin{aligned}\frac{G}{m}\sum_{k=1}^{n}\alpha_k\sum_{i=1}^{m}\left\|x_i^k - x_j^k\right\| &\leq G\sum_{k=1}^{n}\left[\alpha_k\left\|x_j^k - \overline{x}^k\right\| + \frac{1}{m}\sum_{i=1}^{m}\alpha_k\left\|x_i^k - \overline{x}^k\right\|\right] \\ &\leq 2G\left(C_1 + C_2\sum_{k=1}^{n}{\alpha_k}^2\right)\end{aligned} \tag{3.14}$$

Finally, substituting (3.13) and (3.14) back to (3.12) gives the desired result:

$$\left(\sum_{k=1}^{n}\alpha_k\right)\left[f_j^n - f^*\right] \leq \frac{1}{2m}\sum_{i=1}^{m}\left\|x_i^0 - x^*\right\|^2 + 2G\left(C_1 + C_2\sum_{k=1}^{n}{\alpha_k}^2\right)$$

or equivalently,

$$f_j^n - f^* \leq \frac{D_1 + D_2\sum_{k=1}^{n}{\alpha_k}^2}{\sum_{k=1}^{n}\alpha_k}$$

where the constants are given by

$$D_1 = \frac{1}{2m}\sum_{i=1}^{m}\left\|x_i^0 - x^*\right\|^2 + 2GC_1$$

$$D_2 = 2GC_2$$

□

We have thus explicitly characterized the convergence rate of the basic proximal gradient method in terms of step sizes. In the following sections, we discuss the effect of different step size rules on the convergence rate.

3.2.3 Error with Constant Step Size

Observe that if we use a constant step size, the convergence bound reduces to the following:

Corollary 3. *(Convergence rate of the basic distributed proximal gradient method with a constant step size) Consider Algorithm* (3.1) *with* $\alpha_k = \alpha \leq \frac{1}{L}$. *Then we have*

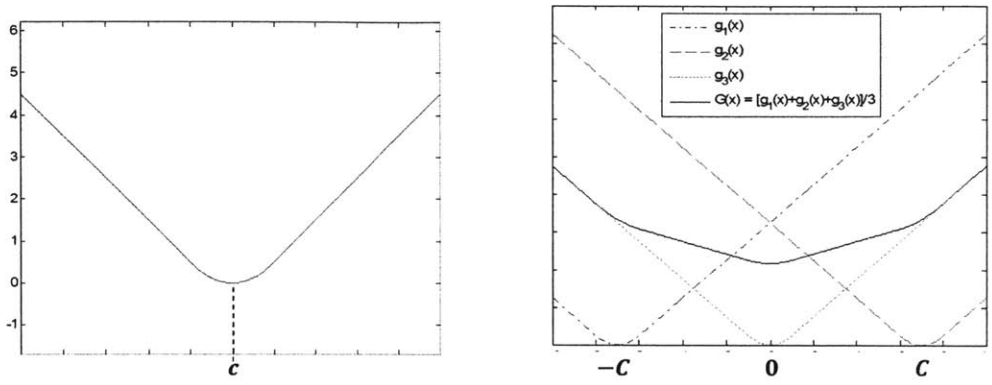
$$f_j^n - f^* \leq \frac{D_1/\alpha}{n} + \alpha D_2$$

where f_j^n, f^*, D_1, D_2 *are defined as in Theroem 1.*

The first term diminishes with the rate of $O(\frac{1}{n})$, similar to its centralized counterpart; however, instead of converging exactly to the optimum, the method is only able to converge to an error neighborhood. The size of the error neighborhood is proportional to α. Under the conventional notion of convergence rates described in Section 2.5, this amounts to a convergence rate of $O(1/\sqrt{n})$, which is the same as that of both the centralized and the distributed subgradient methods [4, 13].

We next present an example showing that due to the distributed nature of the problem, an error neighborhood will be inevitable when a constant step size is used, and therefore, exact convergence cannot be guaranteed.For simplicity, we consider

the gradient method, i.e. $h_i(x) = 0$ for all i, but similar examples could also be constructed for the proximal-gradient method.



(a) $g(x; c)$ (b) A distributed problem with 3 agents

Figure 3-1: Error neighborhood inevitable with a constant step size: an example

Consider the function $g : \mathbb{R} \to \mathbb{R}$, parametrized by $c \in \mathbb{R}$, defined as

$$g(x; c) = \begin{cases} \frac{1}{2}(x-c)^2, & |x-c| \leq 1 \\ x - c - \frac{1}{2}, & x - c > 1 \\ -x + c - \frac{1}{2}, & x - c < -1 \end{cases} \tag{3.15}$$

Figure 3-1(a) illustrates the function. It is convex, continuously differentiable, and its gradient is Lipschitz-continuous with a Lipschitz constant $L = 1$. Moreover, the gradient is bounded by $G = 1$.

Suppose we have a network of three agents whose respective functions are $g_1 = g(x; -C), g_2 = g(x; C)$, and $g_3 = g(x; 0)$, where $C > 0$ is a very large constant. Figure 3-1(b) show that the global objective $g(x) = \frac{1}{3}(g_1(x) + g_2(x) + g_3(x))$ is also convex and has a unique minimum $x^* = 0$.

Next, we construct the communication network that would result in the iterates x_1^k and x_2^k being symmetric with respect to the y-axis, and $x_3^k = 0$ for all k. Suppose there is a communication link between each pair of agents, and that the transition matrix $A^k = A$ is symmetric and static. We choose to $x_i^0 = 0$ to be the common initial point of all three agents. Also, let $a_{31} = a_{32}$, i.e. agent 3 gives equal weights

to the other two agents. Since $g_1(x)$ and $g_2(x)$ are symmetric, the average of x_1^k and x_2^k is always 0, so $\overline{x}^k = 0$ for all k, and $x_3^k = w_3^k = \sum_{j=1}^m a_{3j}^k x_j^k$ will also remain at the global optimum 0. Moreover, let $a_{ii} \geq a_{ij}$ for $i = 1, 2, j = 1, 2$. Then for agents 1 and 2, the result of consensus stage can be modeled as

$$w_i^k = \delta x_i^k, i = 1, 2$$

for some positive scalar $\delta \in [0, 1]$ that depends on the entries of A. In other words, δ measures the effectiveness of the communication step in pulling the iterates toward the average $\overline{x}^k = 0$. Indeed, $\delta = 0$ amounts to $w_i^k = 0 = \overline{x}^{k-1}$, where the consensus stage in fact performs averaging. On the other hand, if δ is close to 1, then the consensus stage performs poorly in averaging the iterates, and each w_i^k is close to x_i^k.

Now let's consider the optimization stage. For agents 1 and 2, if they are not close to their local minimum, so that the gradient has a magnitude of $G = 1$, then the optimization stage brings the iterates closer to their respective local minimum by $\alpha G = \alpha$. More formally, if $x_1^{k-1} > -C + 1$ and $x_2^{k-1} < C - 1$, then

$$x_1^k = w_1^k - \alpha = \delta x_1^{k-1} - \alpha$$
$$x_2^k = w_2^k + \alpha = \delta x_2^{k-1} + \alpha$$

Expressing this recursively, we obtain

$$x_1^k = \delta^k x_1^0 - \delta^{k-1}\alpha - \cdots - \alpha = -\alpha\frac{1-\delta^k}{1-\delta}$$
$$x_2^k = \delta^k x_2^0 + \delta^{k-1}\alpha + \cdots + \alpha = \alpha\frac{1-\delta^k}{1-\delta}$$

This means that when $k \to \infty$, the iterates of agent 1 oscillate between $w_1 = \delta x_1$ and $x_1 = -\frac{\alpha}{1-\delta} = w_1 - \alpha$, and similarly for agent 2. For any $\alpha > 0$, this error $\lim_{k\to\infty} \left\| x_i^k - \overline{x}^k \right\| = \frac{\alpha}{1-\delta}$ will always be present. Indeed, this error represents a "tug-of-war" between the local optimum of $f_i(x)$ and the average iterate $\overline{x}^k$, and is inevitable in a distributed problem.

3.2.4 Diminishing Step Size Choices

In contrast to having an error neighborhood when using a constant step size, we can guarantee exact convergence by using a diminishing step size rule. For example, if the step sizes satisfy the infinite-travel property, $\sum_{k=1}^{\infty} \alpha_k = \infty$, but is square-summable, $\sum_{k=1}^{\infty} {\alpha_k}^2 < \infty$, then it is clear from Theorem 1 that $f_j^n - f^* \to 0$. However, there is a trade-off: diminishing step sizes result in a slower convergence rate. The relationship between the two is the focus for the following section.

We now investigate the convergence rate of Algorithm (3.1) for a parametric class of diminishing step size choices, $\alpha_k = \frac{1}{k^a}$, with $a > 0$.

The convergence bound given in Theorem 1 can be decomposed into two terms: the first of order $O\left(\frac{1}{\sum_{k=1}^{n} \alpha_k}\right)$, and the second of order $O\left(\frac{\sum_{k=1}^{n} {\alpha_k}^2}{\sum_{k=1}^{n} \alpha_k}\right)$. To bound these terms, we use the continuous approximation $s(x) = \frac{1}{x^a}, a > 0$, which is a function decreasing in x and satisfies $s(x_1) < \alpha_k = s(k) < s(x_2)$ for $x_1 > k > x_2$. Then we have the following bounds:

$$\sum_{k=1}^{n} \alpha_k \geq \int_1^{n+1} s(x)dx \geq \int_1^n s(x)dx = \int_1^n x^{-a}dx$$

$$= \begin{cases} \ln n, & a = 1 \\ \frac{n^{-a+1}-1}{-a+1}, & a \neq 1 \end{cases}$$

$$\sum_{k=1}^{n} {\alpha_k}^2 \leq (s(1))^2 = \int_1^n (s(x))^2 dx = \int_1^n x^{-2a}dx$$

$$= \begin{cases} 1 + \ln n, & a = \frac{1}{2} \\ 1 + \frac{n^{-2a+1}-1}{-2a+1} = \frac{n^{-2a+1}-2a}{-2a+1}, & a \neq \frac{1}{2} \end{cases}$$

Therefore, the following cases should be considered:

Case 1. $0 < a < \frac{1}{2}$

$$\frac{1}{\sum_{k=1}^{n} \alpha_k} \leq \frac{1-a}{n^{1-a}-1} = O\left(\frac{1}{n^{1-a}}\right)$$

$$\frac{\sum_{k=1}^{n} {\alpha_k}^2}{\sum_{k=1}^{n} \alpha_k} \leq \frac{1-a}{1-2a} \cdot \frac{n^{1-2a}-2a}{n^{1-a}-1} = O\left(\frac{1}{n^a}\right)$$

Since $1-a > a$, the first term decreases faster than the second term, so the second term dominates.

Case 2. $a = \frac{1}{2}$

$$\frac{1}{\sum_{k=1}^{n} \alpha_k} = \frac{1-a}{n^{1-a}-1} = O\left(\frac{1}{n}\right)$$

$$\frac{\sum_{k=1}^{n} {\alpha_k}^2}{\sum_{k=1}^{n} \alpha_k} = \frac{1+\ln n}{2\left(n^{1/2}-1\right)} = O\left(\frac{\ln n}{\sqrt{n}}\right)$$

Therefore, the second term dominates, and the overall convergence rate is $O\left(\frac{\ln n}{\sqrt{n}}\right)$.

Case 3. $\frac{1}{2} < a < 1$

$$\frac{1}{\sum_{k=1}^{n} \alpha_k} \leq \frac{1-a}{n^{1-a}-1} = O\left(\frac{1}{n^{1-a}}\right)$$

$$\frac{\sum_{k=1}^{n} {\alpha_k}^2}{\sum_{k=1}^{n} \alpha_k} \leq \frac{1-a}{2a-1} \cdot \frac{2a-1/n^{2a-1}}{n^{1-a}-1}$$

Here, the first term dominates.

Case 4. $a > 1$

$$\frac{1}{\sum_{k=1}^{n} \alpha_k} = \frac{a-1}{1-\frac{1}{n^{a-1}}} \to a-1$$

$$\frac{\sum_{k=1}^{n} {\alpha_k}^2}{\sum_{k=1}^{n} \alpha_k} = \frac{a-1}{2a-1} \cdot \frac{2a-\frac{1}{n^{2a-1}}}{1-\frac{1}{n^{a-1}}} \to \frac{(a-1)2a}{(2a-1)}$$

Both terms approach constant values. Step sizes under this choice diminishes too quickly to guarantee exact convergence.

Note that in the first and third cases, the convergence rate is $O\left(\frac{1}{n^b}\right)$, where $b < \frac{1}{2}$. Observe that all such convergence rates are in fact slower than $O\left(\frac{\ln n}{\sqrt{n}}\right)$ in the second case. Indeed, if we compare $O\left(n^{1/2-b}\right)$ and $O\left(\ln n\right)$, the former grows faster since it is a polynomial.

In conclusion, the optimal convergence rate is achieved by choosing $a = \frac{1}{2}$, or equivalently, $\alpha_k = \frac{1}{\sqrt{k}}$. [7, p.1579] also gives the same conclusion, though an analysis was not given.

We also remark that this choice step sizes is not square-summable, because $\sum_{k=1}^{\infty} \left(\frac{1}{\sqrt{k}}\right)^2 = \sum_{k=1}^{\infty} \frac{1}{k} = \infty$. Square-summable step sizes are sufficient for absolute convergence, but not necessary, and certainly not optimal. In contrast, the infinite-travel property is necessary: if $\sum_{k=1}^{\infty} \alpha_k < \infty$, then $\frac{1}{\sum_{k=1}^{n} \alpha_k}$ approaches a constant, creating an error neighborhood instead of guarenteeing exact convergence.

3.3 Challenges for the Accelerated Method

The method presented in the previous section is an extension of the (centralized) basic proximal-gradient method, which is outperformed by Nesterov's accelerated method [3]. This leads us to wonder whether Nesterov's techniques could also be applied to distributed methods so as to accelerate the convergence rate.

[21] studied this for a static communication network whose weight matrix is fixed

and positive-definite. They found that for a constant step size, the function values converge to an error neighborhood of the optimal value with the rate of $O(1/n^2)$. They also proposed a method that uses diminishing step sizes $\alpha_k = 1/k$ and achieves exact convergence at the rate of $O(\log n/n)$, which exhibits a loss, due to consensus effects, from the rate of $O(1/n)$ for the centralized accelerated method with the same diminishing step sizes.

Unfortunately, our time-varying network does not guarantee such a performance. Since the acceleration analysis is more sensitive to error (see [9]), the inevitable consensus terms $\left\|x_i^k - \overline{x}^k\right\|$ results in an accumulation of error that leads to a bound that grows with the iteration number n. From simulation results, we see that in some cases, the iterates indeed diverge and grow unbounded.

To see why this is the case, recall that the acceleration technique makes the prediction y_i^k about where the the function is decreasing, and then takes a proximal-gradient step from there. In the distributed method (3.1), they exchange their iterates y_i^k before taking the proximal-gradient step; other variations for the are possible— for example, the agents may communicate their x_i^k before making a prediction. In any case, information exchange in the consensus stage is the key component for distributed methods. However, time-varying communication weights prevent the prediction from representing a move in the decreasing function of the overall function. In particular, since w_i^{k-1} and w_i^k are different convex combinations of the other agents' iterates, x_i^{k-1} and x_i^k do not have the same relationship as their counterparts in the centralized method, and the prediction $y_i^k = x_i^k + \beta_k(x_i^k - x_i^{k-1})$ may take the iterate even farther from the decreasing direction than without a prediction.

We do note that, in our simulations, when the time-varying weight matrix is restricted to being a positive-definite matrix at all times (which is ensured by additional assumptions on diagonals of the weight matrix, i.e. the weight every agent assigns to itself), the method seems to converge with rate $O(1/n^2)$ to an error neighborhood. It is an open question as to what condition is required of the weight matrix so as to guarantee this performance.

Sensitivity of the (centralized) accelerated method to error leads us to consider al-

ternative ways to control the error so as to provide a convergent bound for distributed accelerated methods. Our solution is to utilize multiple communication steps in the consensus stage, which allows us to bound the error due to consensus. In the next chapter, we shall see that this leads to exact convergence with rate $O(1/t)$ (where t is the number of communication steps), an even better performance than existing accelerated methods with single-step consensus.

Chapter 4

Distributed First-Order Methods with Multi-Step Consensus

In this chapter, we present distributed first-order methods that achieves exact convergence at a rate superior than $O(1/t)$, where t is number of communication steps taken[1].

Section 4.1 is a preliminary discussion that explains the concept of multiple communication steps within the consensus stage, as well as present relevant results pertaining to the convergence of inexact proximal-gradient methods and the summation of polynomial-geometric sequences. In Section 4.2, we present a distributed gradient method with multi-step consensus, and in Section 4.3, a proximal-gradient method. We differentiate the analysis of the two due to the extra consensus stage required in the latter. Finally, in Section 4.4, we show that if the number of communication steps are chosen optimally, these methods can in fact perform better than $O(1/t)$.

[1]Since the number of communication steps are increasing in our multi-step consensus scheme, we assume that it dominates the time required for the optimization stage, and therefore our results are characterized in terms of the number of communication steps. In reality, if the time required for communication steps is significantly less than that of the optimization stage, then the convergence rate of the method could be characterized in terms of the number of optimization stages, and the result would be $O(1/n^2)$ (where n is the iteration number, with one optimization stages per iteration), same as the centralized accelerated method.

4.1 Preliminaries

We saw, in the previous chapter, that the effect of consensus may result in divergence of the accelerated method. Is it possible to somehow control the consensus so as to take advantage of the technique for acceleration? The answer is yes! In this chapter, we present a method that converges exactly to the optimal solution with a rate that is superior than $O(1/t)$, where t is the total number of communication steps.

The main idea is to take an increasing number of communication steps in the consensus stage, so that the effect of the consensus stage becomes more and more like averaging. Due to this extra effort required in the consensus stage, the time required to complete iteration n is increasing with n, so the overall rate does not match up with the centralized version, which converges at rate $O\left(\frac{1}{n^2}\right) = O\left(\frac{1}{t^2}\right)$ since each iteration only takes unit time. Nevertheless, the method achieves a rate of $O(1/t)$, which to our knowledge is faster than all other existing distributed methods for solving (2.1).

With multi-step consensus, the distributed methods can be formulated as *inexact first-order methods*, i.e. centralized first-order methods with error. In the remainder of this section, we state two results that are crucial to our formulation and analysis.

The first characterizes the convergence rate of the inexact proximal-point method in terms of errors.

Proposition 7. *[8, Proposition 2] Let $g : \mathbb{R}^d \to \mathbb{R}$ be a convex function that has a Lipschitz continuous gradient with Lipschitz constant L, and let $h : \mathbb{R}^d \to (-\infty, \infty]$ be a lower semi-continuous proper convex function. Suppose the function $f = g + h$ attains its minimum at a certain $x^* \in \mathbb{R}^d$.*

Given two sequences $\{e^k\}_{k=1}^{\infty}$ and $\{\varepsilon_k\}_{k=1}^{\infty}$, where $e^k \in \mathbb{R}^d$ and $\varepsilon \in \mathbb{R}$ for every k, consider the accelerated inexact proximal gradient method, which iterates the following recursion:

$$\begin{cases} x^k & \in prox_{h,\varepsilon_k}^{\alpha}\{y^{k-1} - \alpha\left(\nabla g(y^{k-1}) + e^k\right)\} \\ y^k & = x^k + \frac{k-1}{k+2}\left(x^k - x^{k-1}\right) \end{cases} \tag{4.1}$$

where the step size is $\alpha = \frac{1}{L}$, *and*

$$prox_{h,\varepsilon}^{\alpha}\{y\} = \left\{x \in \mathbb{R}^d \mid h(x) + \frac{1}{2\alpha}\|x-y\|^2 \leq \min_{z\in\mathbb{R}^d}\left(h(z) + \frac{1}{2\alpha}\|z-y\|^2\right) + \varepsilon\right\} \tag{4.2}$$

indicates the set of all ε*-optimal solutions for the proximal operator.*

Then, for all $n \geq 1$, *we have*

$$f(x^n) - f(x^*) \leq \frac{2L}{(n+1)^2}\left(\|x_o - x^*\| + 2\tilde{A}_n + \sqrt{2\tilde{B}_k}\right)^2$$

where

$$\tilde{A}_n = \sum_{k=1}^{n} k\left(\frac{\|e^k\|}{L} + \sqrt{\frac{2\varepsilon_k}{L}}\right), \quad \tilde{B}_n = \sum_{k=1}^{n}\frac{k^2\varepsilon_k}{L}.$$

Proposition 7 indicates that as long as the sequences $\{k\|e^k\|\}_{k=1}^{\infty}$ and $\{k\sqrt{\varepsilon_k}\}_{k=1}^{\infty}$ are both summable, then the accelerated inexact gradient method achieves the optimal convergence rate of $O(\frac{1}{n^2})$. It is straightforward to verify that according to the analysis in [8], the result also holds for a constant step size $\alpha \leq \frac{1}{L}$. Also, note that if we simply set $h(x) = 0$ and $\varepsilon_k = 0$, then the result holds for inexact gradient methods, which will be considered in Section 4.2.

We shall see that errors in our inexact formulation, introduced by the distributed nature of our problem and controlled by multi-step consensus, can be bounded by polynomial-geometric sequences, i.e. sequences of the form $\{p(k)\gamma^k\}_{k=1}^{\infty}$ for some $\gamma \in (0,1)$ and some polynomial $p(k)$ of k. The next proposition show that such sequences are summable.

Proposition 8. *(Summability of polynomial-geometric sequences)*

Let γ *be a positive scalar such that* $\gamma < 1$, *and let*

$$P(k,N) = \{c_N k^N + c_{N-1}k^{N-1} + ... + c_1 k + c_0 \mid c_j \in \mathbb{R}, j = 0, ..., N\}$$

denote the set of all N*-th order polynomials of* k, *where* N *is a nonnegative integer.*

Then for all $p(k) \in P(k, N)$,

$$\sum_{k=0}^{\infty} p(k)\gamma^k < \infty.$$

Proof. We proceed by induction on N, the degree of the polynomial.

For $N = 0$, every polynomial $p(k) \in P(k, N)$ is a constant, i.e., $p(k) = c_0$ for some scalar c_0. Therefore, $\sum_{k=0}^{\infty} p(k)\gamma^k = \frac{c_0}{1-\gamma}$, which is the well-known geometric series.

Now suppose the induction hypothesis holds for some nonnegative interger N. We show that it holds for $N+1$. Note that it is sufficient to show that $\sum_{k=0}^{\infty} k^{N+1}\gamma^k < \infty$, because $P(k, N+1) = \{c_{N+1}k^{N+1} + p(k) \mid c_{N+1} \in \mathbb{R}, p(k) \in P(k, N)\}$. Using simple algebraic identities, we can write

$$\begin{aligned}
\sum_{k=0}^{\infty} k^{N+1}\gamma^k &= \sum_{k=1}^{\infty}\sum_{l=1}^{k} \left[l^{N+1} - (l-1)^{N+1}\right]\gamma^k \\
&= \sum_{l=1}^{\infty}\sum_{k=l}^{\infty} \left[l^{N+1} - (l-1)^{N+1}\right]\gamma^k \\
&= \sum_{l=1}^{\infty} \left[l^{N+1} - (l-1)^{N+1}\right]\sum_{k=l}^{\infty}\gamma^k \\
&= \sum_{l=1}^{\infty} \left[l^{N+1} - (l-1)^{N+1}\right]\frac{\gamma^l}{1-\gamma}.
\end{aligned}$$

Note that $l^{N+1} - (l-1)^{N+1}$ is a N-th order polynomial of l, i.e., $l^{N+1} - (l-1)^{N+1} \in P(l, N)$. Therefore, by the induction hypothesis on N, $\sum_{l=1}^{\infty}\left(l^{N+1} - (l-1)^{N+1}\right)\gamma^l < \infty$, and thus, $\sum_{k=0}^{\infty} k^{N+1}\gamma^k < \infty$.

□

The result of this proposition for $p(k, N) = k^N$ will be particularly useful for our analysis in the upcoming sections. Therefore, we make the following definition:

$$S_N^{\gamma} := \sum_{k=0}^{\infty} k^N\gamma^k < \infty. \tag{4.3}$$

4.2 Gradient Method

4.2.1 Introduction

Consider the optimization problem

$$\min_{x\in\mathbb{R}^d} g(x) = \frac{1}{m}\sum_{i=1}^{m} g_i(x)$$

where for each $i = 1, ..., m$, $g_i : \mathbb{R}^d \to \mathbb{R}$ is a convex and continuously differentiable function whose gradient is Lipschitz-continuous with Lipschitz constant L.

The *distributed accelerated gradient method with multi-step consensus* solves this by iteratively performing the following updates for $k \geq 1$ from intial points $\{w_i^0\}_{i=1}^m$:

$$\begin{cases} x_i^k = w_i^{k-1} - \alpha_k \nabla g_i(w_i^{k-1}) & \text{(4.4a)} \\ y_i^k = x_i^k + \beta_k \left(x_i^k - x_i^{k-1}\right) & \text{(4.4b)} \\ w_i^k = \sum_{j=1}^{m} \lambda_{ij}^k y_j^k & \text{(4.4c)} \end{cases}$$

Similar to (2.2), this method also has two stages:

- In the *optimization stage* (4.4a)–(4.4b), $\alpha_k \leq \frac{1}{L}$ is the constant step size, and $\beta_k = \frac{k-1}{k+2}$ is chosen so as to achieve the optimal convergence rate. We shall focus on the constant step size rule, i.e. $\alpha_k = \alpha$ for all $k \geq 1$.

- In the *consensus stage* (4.4c), the effect of consensus is controlled by performing multiple communication steps in each consensus stage. Specifically, $\lambda_{ij}^k = [\Phi(t_k, t_{k-1} + 1)]_{ij}$, where $t_0 = 0$ and $t_k = t_{k-1} + k + 1$ for $k \geq 1$. In other words, k communication steps are taken in the consensus stage of iteration k.

  We note two implications of the consensus stage that will be useful for our analysis in relating w_i^k to the average iterate $\overline{w}^k = \frac{1}{m}\sum_{i=1}^m w_i^k$. First, since $[\lambda_{ij}^k]$

is doubly stochastic, we have

$$\overline{w}^k = \frac{1}{m}\sum_{i=1}^{m} w_i^k = \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_{ij}^k y_j^k = \frac{1}{m}\sum_{j=1}^{m} y_j^k = \overline{y}^k,$$

similar to the single-step consensus case.

Secondly, we have

$$\begin{aligned}\left\|w_i^k - \overline{w}^k\right\| &= \left\|\sum_{j=1}^{m}\left(\lambda_{ij}^k - \frac{1}{m}\right) y_j^k\right\| \leq \sum_{j=1}^{m}\left|\lambda_{ij}^k - \frac{1}{m}\right| \left\|y_j^k\right\| \\ &\leq \Gamma\gamma^k \sum_{j=1}^{m}\left\|y_j^k\right\|, \end{aligned} \tag{4.5}$$

where in the final inequality, we used Lemma 1 to obtain that for the multi-step consensus stage,

$$\left|\lambda_{ij}^k - \frac{1}{m}\right| = \left|[\Phi(t_k, t_{k-1}+1)]_{ij} - \frac{1}{m}\right| \leq \Gamma\gamma^k.$$

The effect of having multiple communication steps in the consensus stage is that the problem can be reformulated as the centralized gradient method with an error is controllable by the consensus stage. To understand this relationship, we first examine the case where the consensus stage provides complete information that allows the agents to reach perfect consensus, i.e. $\lambda_{ij}^k = \frac{1}{m}$ for every i, j and $w_i^k = \overline{w}^k$ for every i. Then, taking the average of (4.4) and recalling that $\nabla g(x) = \frac{1}{m}\sum_{i=1}^{m} \nabla g_i(x)$, we have

$$\begin{cases} \overline{x}^k = \overline{y}^{k-1} - \alpha\nabla g(\overline{y}^{k-1}) & \text{(4.6a)} \\ \overline{y}^k = \overline{x}^k + \beta_k\left(\overline{x}^k - \overline{x}^{k-1}\right) & \text{(4.6b)} \end{cases}$$

which is the centralized accelerated gradient method, known to converge with rate $O(1/n^2)$ [4].

In general, however, w_i^k is not $\overline{w}^k$, but an approximation of it, and the quality of this approximation is bounded by (4.5). The problem then becomes an *inexact* proximal-point method under the framework of [8]. Indeed, taking the average of

(4.4), we have the following formulation:

Proposition 9. *(Distributed gradient method as an inexact centralized gradient method)*

Algorithm (4.4)*, with a constant step size* $\alpha_k = \alpha$*, can be written as*

$$\begin{cases} \overline{x}^k &= \overline{y}^{k-1} - \alpha \left[\nabla g(\overline{y}^{k-1}) + e^k\right] \\ \overline{y}^k &= \overline{x}^k + \beta_k \left(\overline{x}^k - \overline{x}^{k-1}\right) \end{cases} \tag{4.7}$$

where

$$\left\|e^k\right\| \leq L\Gamma\gamma^k \sum_{j=1}^{m} \left\|y_j^k\right\|.$$

Proof. Note that

$$e^k = \frac{1}{m}\sum_{i=1}^{m} \left[\nabla g_i(w_i^{k-1}) - \nabla g_i(\overline{y}^{k-1})\right].$$

Since the gradients of g_i are Lipschitz-continuous with Lipschitz constant L, we have

$$\left\|e^k\right\| \leq \frac{1}{m}\sum_{i=1}^{m} \left\|\nabla g_i(w_i^{k-1}) - \nabla g_i(\overline{y}^{k-1})\right\| \leq \frac{L}{m}\sum_{i=1}^{m} \left\|w_i^{k-1} - \overline{w}^{k-1}\right\| \tag{4.8}$$

where we recall that $\overline{y}^{k-1} = \overline{w}^{k-1}$. The right-hand side, in turn, can be bounded by (4.5), giving the desired expression.

□

In the next subsection, we shall see that the term $\sum_{j=1}^{m} \left\|y_j^k\right\|$ can be bounded by a polynomial of k. This in turn allows for, first, the use of Lemma 8 in showing that $\{k\left\|e^k\right\|\}$ is summable, and then, the application of Lemma 7 for the convergence rate in the final subsection

4.2.2 Bound on Iterates

In this subsection, we show that $\sum_{j=1}^{m} \left\|y_j^k\right\|$ can be bounded by a second-order polynomial of k. We need the following recursive expressions of the iterates:

Proposition 10. *(Recursive expressions of iterates)*

Consider Algorithm (4.4) *with a constant step size* $\alpha_k = \alpha$. *Then we have, for every* $k \geq 2$,

(a) $\sum_{i=1}^m \left\|y_i^k\right\| \leq \sum_{j=1}^m \left\|y_j^{k-1}\right\| + m\alpha G + \beta_k \sum_{i=1}^m \left\|x_i^k - x_i^{k-1}\right\|$

(b) $\sum_{i=1}^m \left\|x_i^k - x_i^{k-1}\right\| \leq 2m\Gamma \sum_{l=1}^{k-2} \gamma^l \sum_{j=1}^m \left\|y_j^l\right\| + C_x + (k-2)5m\alpha G$ *for some scalar* C_x.

Proof. (a) Substituting (4.4c) into (4.4a), we have

$$x_i^k = \sum_{j=1}^m \lambda_{ij}^{k-1} y_j^{k-1} - \alpha \nabla g_i(w_i^{k-1}) \tag{4.9}$$

Applying this to (4.4b), we have

$$y_i^k = \sum_{j=1}^m \lambda_{ij}^{k-1} y_j^{k-1} - \alpha \nabla g_i(w_i^{k-1}) + \beta_k \left(x_i^k - x_i^{k-1}\right)$$

which, upon taking the norm and summing over all i, yields (a):

$$\begin{aligned}\sum_{i=1}^m \left\|y_i^k\right\| &\leq \sum_{i=1}^m \sum_{j=1}^m \lambda_{ij}^{k-1} \left\|y_j^{k-1}\right\| + \alpha \sum_{i=1}^m \left\|\nabla g_i(w_i^{k-1})\right\| + \beta_k \sum_{i=1}^m \left\|x_i^k - x_i^{k-1}\right\| \\ &\leq \sum_{j=1}^m \left\|y_j^{k-1}\right\| + m\alpha G + \beta_k \sum_{i=1}^m \left\|x_i^k - x_i^{k-1}\right\|\end{aligned}$$

where in the last line we used the fact that $[\lambda_{ij}^k] = \Phi(t_k, t_{k-1} + 1)$ is doubly stochastic, and that the gradients are bounded by Assumption 1. (These two properties will also be used extensively in deriving the following expressions, and we shall avoid repeating this sentence when the application is straightforward.)

(b) Next, we focus on the last term of part (a), $\sum_{i=1}^m \left\|x_i^k - x_i^{k-1}\right\|$. Substituting (4.4b) into (4.9) gives

$$x_i^k = \sum_{j=1}^m \lambda_{ij}^{k-1} \left[x_j^{k-1} + \beta_{k-1}\left(x_j^{k-1} - x_j^{k-2}\right)\right] - \alpha \nabla g_i(w_i^{k-1}).$$

Therefore, subtracting x_i^{k-1} and taking the norm, we obtain

$$\left\|x_i^k - x_i^{k-1}\right\| \leq \sum_{j=1}^{m} \lambda_{ij}^{k-1} \left[\left\|x_j^{k-1} - x_i^{k-1}\right\| + \beta_{k-1}\left\|x_j^{k-1} - x_j^{k-2}\right\|\right] + \alpha G,$$

which, upon summation over all i, yields

$$\sum_{i=1}^{m} \left\|x_i^k - x_i^{k-1}\right\| \leq \sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_{ij}^{k-1}\left\|x_j^{k-1} - x_i^{k-1}\right\| + \beta_{k-1}\sum_{j=1}^{m}\left\|x_j^{k-1} - x_j^{k-2}\right\| + m\alpha G. \tag{4.10}$$

The first term on the right-hand side of (4.10) is a measure of how "scattered" the iterates x_i^k are, and is bounded by their distance to the average iterate $\overline{x}^k$:

$$\begin{aligned}\sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_{ij}^k \left\|x_i^k - x_j^k\right\| &\leq \sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_{ij}^k \left(\left\|x_i^k - \overline{x}^k\right\| + \left\|x_j^k - \overline{x}^k\right\|\right) \\ &= \sum_{i=1}^{m}\left(\sum_{j=1}^{m}\lambda_{ij}^k\right)\left\|x_i^k - \overline{x}^k\right\| + \sum_{j=1}^{m}\left(\sum_{i=1}^{m}\lambda_{ij}^k\right)\left\|x_j^k - \overline{x}^k\right\| \\ &= 2\sum_{i=1}^{m}\left\|x_i^k - \overline{x}^k\right\|. \end{aligned} \tag{4.11}$$

This can, in turn, be bounded using (4.4a) and (4.5),

$$\begin{aligned}\left\|x_i^k - \overline{x}^k\right\| &= \left\| w_i^{k-1} - \overline{w}^{k-1} - \alpha\left[\nabla g_i(w_i^k) - \frac{1}{m}\sum_{j=1}^{m}\nabla g_j(w_j^{k-1})\right]\right\| \\ &\leq \Gamma\gamma^{k-1}\sum_{j=1}^{m}\left\|y_j^{k-1}\right\| + 2\alpha G. \end{aligned} \tag{4.12}$$

Therefore,

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\lambda_{ij}^k\left\|x_i^k - x_j^k\right\| \leq 2m\Gamma\gamma^{k-1}\sum_{j=1}^{m}\left\|y_j^{k-1}\right\| + 4m\alpha G.$$

Substituting this (with a decrement of 1 in the indices) into (4.10), we have

$$\sum_{i=1}^{m} \left\|x_i^k - x_i^{k-1}\right\| \leq 2m\Gamma\gamma^{k-2} \sum_{j=1}^{m} \left\|y_j^{k-2}\right\| + \beta_{k-1} \sum_{i=1}^{m} \left\|x_i^{k-1} - x_i^{k-2}\right\| + 5m\alpha G.$$

Recursively expanding the term $\sum_{i=1}^{m} \left\|x_i^k - x_i^{k-1}\right\|$ yields

$$\sum_{i=1}^{m} \left\|x_i^k - x_i^{k-1}\right\| \leq 2m\Gamma \sum_{l=1}^{k-2} \gamma^l \sum_{j=1}^{m} \left\|y_j^l\right\| + \beta_2 \sum_{i=1}^{m} \left\|x_i^2 - x_i^1\right\| + (k-2)5m\alpha G.$$

Concerning the term $\beta_2 \sum_{i=1}^{m} \|x_i^2 - x_i^1\|$, note that

$$\sum_{i=1}^{m} \left\|x_i^1\right\| = \sum_{i=1}^{m} \left\|w_i^0 - \alpha \nabla g_i(w_i^0)\right\| \leq \sum_{i=1}^{m} \left\|w_i^0\right\| + m\alpha G < \infty.$$

Also, since $\beta_1 = 0$ and $y_i^1 = x_i^1$, (4.9) gives

$$\sum_{i=1}^{m} \left\|x_i^2\right\| = \sum_{i=1}^{m} \left\|\sum_{j=1}^{m} \lambda_{ij}^1 x_j^1 - \alpha \nabla g_i(w_i^1)\right\| \leq \sum_{j=1}^{m} \left\|x_j^1\right\| + m\alpha G < \infty.$$

Therefore, given initial points $\{w_i^0\}_{i=1}^m$, there exists a scalar C_x such that

$$\beta_2 \sum_{i=1}^{m} \left\|x_i^2 - x_i^1\right\| \leq C_x.$$

□

We now present the main lemma that leads to a polynomial bound on $\sum_{i=1}^{m} \left\|y_i^k\right\|$:

Lemma 3. *(Polynomial bound on $\sum_{i=1}^{m} \left\|y_i^k\right\|$)*

Let sequences $\{x_i^k\}_{k=1}^{\infty}$ and $\{y_i^k\}_{k=1}^{\infty}$ be generated as in algorithm (4.4). Then there exist nonnegative scalars $\overline{C}_0, \overline{C}_1, \overline{C}_2$ such that for $k \geq 1$,

$$\sum_{i=1}^{m} \left\|y_i^k\right\| \leq \overline{C}_0 + \overline{C}_1 k + \overline{C}_2 k^2$$

Proof. We proceed by induction on the iteration number k. First, we show that the result holds for $k = 1$ by showing that $\sum_{i=1}^{m} \left\|y_i^1\right\| < \infty$ and simply choosing

$\overline{C}_0 \geq \sum_{i=1}^m \|y_i^1\|$.

Since $\beta_1 = 0$, we have $y_i^1 = x_i^1 = w_i^0 - \alpha \nabla g_i(w_i^0)$ for all i. Therefore,

$$\sum_{i=1}^m \|y_i^1\| \leq \sum_{i=1}^m \left(\|w_i^0\| + \alpha G\right) < \infty.$$

for arbitrarily given initial points $\{w_i^0\}_{i=1}^m$.

Now suppose the result holds for all nonnegative integers no greater than $k-1$. We wish to show that it also holds for k.

By Proposition 10(b) and the induction hypothesis for $1, ..., k-2$,

$$\begin{aligned}\sum_{i=1}^m \|x_i^k - x_i^{k-1}\| &\leq 2m\Gamma \sum_{l=1}^{k-2} \gamma^l \left(\overline{C}_0 + \overline{C}_1 l + \overline{C}_2 l^2\right) + C_x + (k-2)5m\alpha G \\ &\leq 2m\Gamma \left(\overline{C}_0 S_0^\gamma + \overline{C}_1 S_1^\gamma + \overline{C}_2 S_2^\gamma\right) + C_x + (k-2)5m\alpha G,\end{aligned}$$

where the last line is due to Proposition 8, with scalars $S_0^\gamma, S_1^\gamma, S_2^\gamma$ defined in (4.3) as

$$S_N^\gamma := \sum_{k=0}^\infty k^N \gamma^k, N = 0, 1, 2.$$

Substituting this into Proposition 10(a), using the induction hypothesis for $k-1$, we obtain

$$\begin{aligned}\sum_{i=1}^m \|y_i^k\| &\leq \overline{C}_0 + \overline{C}_1(k-1) + \overline{C}_2(k-1)^2 + 2m\Gamma \left(\overline{C}_0 S_0^\gamma + \overline{C}_1 S_1^\gamma + \overline{C}_2 S_2^\gamma\right) + C_x + (5k-9)m\alpha G \\ &= \overline{C}_2 k^2 + (\overline{C}_1 - 2\overline{C}_2 + 5m\alpha G)k \\ &\quad + \overline{C}_0 + \overline{C}_2 - \overline{C}_1 + 2m\Gamma \left(\overline{C}_0 S_0^\gamma + \overline{C}_1 S_1^\gamma + \overline{C}_2 S_2^\gamma\right) + C_x - 9m\alpha G\end{aligned}$$

Comparing coefficients, we see that the right-hand side is bounded above by $\overline{C}_0 + \overline{C}_1 k + \overline{C}_2 k^2$ if

$$\overline{C}_1 - 2\overline{C}_2 + 5m\alpha G \leq \overline{C}_1,$$

for the coefficient of k, and

$$\overline{C}_0 + \overline{C}_2 - \overline{C}_1 + 2m\Gamma\left(\overline{C}_0 S_0^\gamma + \overline{C}_1 S_1^\gamma + \overline{C}_2 S_2^\gamma\right) + C_x - 9m\alpha G \leq \overline{C}_0,$$

for the constant coefficient. Therefore, the following choices of scalars ensure that the induction hypothesis also holds for k:

$$\begin{aligned}
\overline{C}_2 &= \frac{5}{2} m\alpha G, \\
\overline{C}_1 &= \frac{2m\Gamma S_0^\gamma \overline{C}_0 + (1 + 2m\Gamma S_2^\gamma)\,\overline{C}_2 + C_x - 9m\alpha G}{1 - 2m\Gamma S_1^\gamma}, \\
\overline{C}_0 &= \sum_{i=1}^{m} \|y_i^1\|.
\end{aligned}$$

□

4.2.3 Convergence Rate

We now apply Lemma 5 on the error sequences in (4.19) to show that $\{k\|e^k\|\}_{k=1}^\infty$ is a polynomial-geometric sequence, and therefore summable.

Lemma 4. *In the formulation* (4.7)*, where*

$$\|e^k\| \leq L\Gamma\gamma^k \sum_{j=1}^{m} \|y_j^k\|,$$

we have

$$\sum_{k=1}^{\infty} k\|e^k\| < \infty.$$

Proof. It is clear from Lemma 3 that

$$k\|e^k\| \leq L\Gamma\gamma^k k\left(\overline{C}_0 + \overline{C}_1 k + \overline{C}_2 k^2\right),$$

which is a polynomial-geometric sequence, thus summable by Lemma 8. □

Empowered by the lemma above, we can apply the convergence result of the inexact gradient method in Proposition 7 to show that the method achieves exact

convergence with

$$f(x^n) - f(x^*) \leq \frac{D}{(n+1)^2}$$

for some scalar D, where n is the number of full iterations of (4.4). However, the computation time required to complete each iteration is increasing, due to the increasing number of communication steps taken. The following theorem expresses the convergence rate in terms of the actual running time, which is proportional to the number of optimization and communication steps taken.

Theorem 2. *(Convergence rate of the distributed gradient method with multi-step consensus)*

Let Algorithm (4.4) *be such that k communication steps are performed within* (4.4c) *at iteration k, i.e. $\lambda_{ij}^k = [\Phi(t_k, t_{k-1}+1)]_{ij}$, where $t_k = t_{k-1} + k + 1$ and $t_0 = 0$. Suppose also that optimization and communication steps each takes unit time. Then, for all $t \geq 1$, where t is the total number of communication steps taken, we have*

$$f(\overline{x}^t) - f(x^*) = O(1/t).$$

Proof. Since it takes k communication steps to complete iteration k, the total number of communication steps required to execute iterations $1, ..., n$ is

$$\sum_{k=1}^{n} k = \frac{n(n+1)}{2}.$$

In other words, after t communication steps, the number of iterations completed is n, where n is the greatest integer such that

$$\frac{n(n+1)}{2} = \frac{n^2+n}{2} \leq t,$$

or equivalently,

$$n = \left\lfloor \frac{-1+\sqrt{1+8t}}{2} \right\rfloor$$

As a result,

$$(n+1)^2 \geq \left(\frac{-1+\sqrt{1+8t}}{2}\right)^2 = \frac{2+8t-2\sqrt{1+8t}}{4},$$

and thus,

$$f(\overline{x}^t) - f(x^*) \leq \frac{D}{(n+1)^2} \leq \frac{2D}{4t+1-\sqrt{1+8t}} = O(1/t).$$

□

We have thus shown that our distributed gradient method with multi-step consensus achieves a convergence rate of $O(1/t)$, which is superior than currently known distributed gradient methods.

Finally, we remark that although this convergence rate is given in terms of the average iterate $\overline{x}^k$, the local iterates x_i^k also comes close to $\overline{x}^k$ because of the increasing number of consensus steps. Due to the distributed nature of the problem, there is no guarantee that x_i^k will converge exactly to $\overline{x}^t$, because even at the global optimum x^*, where $\nabla g(x^*) = \frac{1}{m}\sum_{i=1}^m \nabla g_i(x^*) = 0$, each local component of the gradient, $\nabla g_i(x^*)$, may still be nonzero, causing $x_i^k = w_i^k - \alpha \nabla g_i(w_i^k)$ to be taken away from $w_i^k \approx x^*$. Therefore, w_i^k may be a better estimate of x^* than x_i^k in terms of practical implementation.

4.3 Proximal-Gradient Method

4.3.1 Introduction

In the previous section, we developed a distributed gradient method using multi-step consensus that converges exactly. We now wish to extend this to the proximal-gradient method for functions that have a non-differentiable component.

A straightforward extension is as follows:

$$\begin{cases} x_i^k &= \text{prox}_{h_i}^{\alpha_k}\left\{w_i^{k-1} - \alpha_k \nabla g_i(w_i^{k-1})\right\} \\ y_i^k &= x_i^k + \beta_k\left(x_i^k - x_i^{k-1}\right) \\ w_i^k &= \sum_{j=1}^m \lambda_{ij}^k y_j^k \end{cases} \tag{4.13}$$

The difficulty in this approach is that its formulation as an inexact centralized proximal-gradient method has an error sequence that does not leads to exact convergence as in the previous chapter. To see why, we consider the special case where the nonsmooth function $h_i(x) = h(x)$ for every i, and all weights λ_{ij}^k are equal to $\frac{1}{m}$, implying that $w_i^k = \overline{w}^k$. Taking the average of (4.13) and using a constant step size $\alpha_k = \alpha$, we have

$$\begin{cases} \overline{x}^k & = \frac{1}{m}\sum_{i=1}^m \operatorname{prox}_h^\alpha \left\{\overline{w}^{k-1} - \alpha \nabla g_i(\overline{w}^{k-1})\right\} \\ \overline{y}^k & = \overline{x}^k + \beta_k \left(\overline{x}^k - \overline{x}^{k-1}\right) \\ \overline{w}^k & = \frac{1}{m}\sum_{i=1}^m \sum_{j=1}^m \lambda_{ij}^k y_j^k = \overline{y}^k \end{cases}$$

Recall from (4.1) that the centralized proximal gradient method for $g(x) = \frac{1}{m}\sum_{i=1}^m g_i(x)$ and $h(x) = \frac{1}{m}\sum_{i=1}^m h_i(x)$ is

$$\begin{cases} \overline{x}^k & \in \operatorname{prox}_{h,\varepsilon_k}^\alpha \{\overline{y}^{k-1} - \alpha \left(\nabla g(\overline{y}^{k-1}) + e^k\right)\}, \\ \overline{y}^k & = \overline{x}^k + \frac{k-1}{k+2}\left(\overline{x}^k - \overline{x}^{k-1}\right). \end{cases} \tag{4.14}$$

Let $\tilde{x}^k = \operatorname{prox}_h^\alpha\{\overline{y}^{k-1} - \alpha\left(\nabla g(\overline{y}^{k-1}) + e^k\right)\}$. Then by (4.2), we can define ε_k using the following:

$$\begin{aligned} & h(\overline{x}^k) + \frac{1}{2\alpha}\left\|\overline{x}^k - \left[\overline{y}^{k-1} - \alpha\left(\nabla g(\overline{y}^{k-1}) + e^k\right)\right]\right\|^2 \\ & - h(\tilde{x}^k) - \frac{1}{2\alpha}\left\|\tilde{x}^k - \left[\overline{y}^{k-1} - \alpha\left(\nabla g(\overline{y}^{k-1}) + e^k\right)\right]\right\|^2 \\ & \leq -\left\langle \tilde{x}^k - \overline{x}^k, z + \frac{1}{\alpha}(\overline{x}^k - \left[\overline{y}^{k-1} - \alpha\left(\nabla g(\overline{y}^{k-1}) + e^k\right)\right])\right\rangle \\ & \leq \left\|\tilde{x}^k - \overline{x}^k\right\| \cdot \left(\|z\| + \frac{1}{\alpha}\left\|\overline{x}^k - \left[\overline{y}^{k-1} - \alpha\left(\nabla g(\overline{y}^{k-1}) + e^k\right)\right]\right\|\right) =: \varepsilon_k, \end{aligned}$$

where the first inequality is due to convexity of the proximal function $h(x) + \frac{1}{2\alpha}\|x - y\|^2$,

and $z \in \partial h(\overline{x}^k)$. Thus, we have $\|z\| \leq G$. As for the term $\left\|\tilde{x}^k - \overline{x}^k\right\|$, we have

$$\begin{aligned}
\left\|\tilde{x}^k - \overline{x}^k\right\| &= \left\|\text{prox}_h^\alpha\{\overline{y}^{k-1} - \alpha\left(\nabla g(\overline{y}^{k-1}) + e^k\right) - \frac{1}{m}\sum_{i=1}^m \text{prox}_h^\alpha\left\{\overline{y}^{k-1} - \alpha\nabla g_i(\overline{y}^{k-1})\right\}\right\| \\
&\leq \frac{1}{m}\sum_{i=1}^m \left\|\text{prox}_h^\alpha\{\overline{y}^{k-1} - \alpha\left(\nabla g(\overline{y}^{k-1}) + e^k\right) - \text{prox}_h^\alpha\left\{\overline{y}^{k-1} - \alpha\nabla g_i(\overline{y}^{k-1})\right\}\right\| \\
&\leq \frac{\alpha}{m}\sum_{i=1}^m \left\|-\nabla g(\overline{y}^{k-1}) - e^k + \nabla g_i(\overline{y}^{k-1})\right\| \\
&\leq \alpha\left(2G + \left\|e^k\right\|\right),
\end{aligned}$$

where the inequalities are due to the property of the norm function in the second line, the nonexpansiveness of proximal operators (Proposition 5) in the third line, and the gradient bound in the final line. Therefore, the error sequence $\{\varepsilon_k\}$ is bounded below by

$$\varepsilon_k \geq \left\|\tilde{x}^k - \overline{x}^k\right\| \cdot \|z\| \geq \alpha\left(2G + \left\|e^k\right\|\right) G \geq 2\alpha G^2.$$

In other words, under this formulation, $\{\varepsilon_k\}$ is bounded below by a constant, and thus $\{k\sqrt{\varepsilon_k}\}$ is not summable. Therefore, the inexact method (4.14) does not converge exactly.

This difficulty can be avoided by performing multi-step consensus between the gradient and proximal steps, and limiting our attention to the case where all private functions have the same non-differentiable component, that is, $h_i(x) = h(x)$ for all i, which is often the case for applications of our interest (such as when h is a regularization term that does not depend on local information.)

Our proposed *distributed proximal-gradient method with multi-step consensus* is as follows:

$$\begin{cases}
q_i^k = y_i^{k-1} - \alpha_k \nabla g_i(y_i^{k-1}) & \text{(4.15a)} \\
\hat{q}_i^k = \sum_{j=1}^m \hat{\lambda}_{ij}^k q_j^k & \text{(4.15b)} \\
x_i^k = \text{prox}_h^{\alpha_k}\left\{\hat{q}_i^k\right\} & \text{(4.15c)} \\
y_i^k = x_i^k + \beta_k\left(x_i^k - x_i^{k-1}\right) & \text{(4.15d)}
\end{cases}$$

In this method, optimization is divided into two parts, (4.15a) and (4.15c)–(4.15d), with multi-step consensus inserted between them. More specifically:

- In (4.15a), q_i^k denotes the result of the gradient step from the previous iterate y_i^{k-1}.

- (4.15b) is the consensus stage, in which we take k communication steps for iteration k, i.e.
$$\left|\hat{\lambda}_{ij}^k - \frac{1}{m}\right| \leq \Gamma\gamma^k.$$
The result is $\hat{q}_i^k$, which is an estimate of $\overline{q}^k = \frac{1}{m}\sum_{i=1}^m q_i^k$ and a convex combination of other agents' gradient step results. Note that by Lemma 1, we have
$$\begin{aligned}\left\|\hat{q}_i^k - \overline{q}^k\right\| &= \left\|\sum_{j=1}^m \hat{\lambda}_{ij}^k q_j^k - \frac{1}{m}q_j^k\right\| \leq \sum_{j=1}^m \left|\hat{\lambda}_{ij}^k - \frac{1}{m}\right| \left\|q_j^k\right\| \\ &\leq \Gamma\gamma^k \sum_{j=1}^m \left\|q_j^k\right\|. \end{aligned} \tag{4.16}$$
This expression will help us bound the error in the inexact formulation, to be introduced shortly.

- In (4.15c), x_i^k denotes the result of the proximal step from $\hat{q}_i^k$. Note that we could also write
$$x_i^k = \hat{q}_i^k - \alpha z_i^k, \text{ where } z_i^k \in \partial h(x_i^k). \tag{4.17}$$
Since h has bounded subgradients, this also implies
$$\left\|x_i^k - \hat{q}_i^k\right\| \leq \alpha G. \tag{4.18}$$

- (4.15d) is the Nesterov-type acceleration step, with $\beta_k = \frac{k-1}{k+2}$.

We now show that this method can be formulated as an inexact centralized proximal gradient method in the framework of [8]:

Proposition 11. *(Distributed proximal-gradient method as an inexact centralized proximal-gradient method)*

Algorithm (4.15), *with a constant step size* $\alpha_k = \alpha$, *can be written as*

$$\begin{cases} \overline{x}^k & \in prox^{\alpha}_{h,\varepsilon_k}\{\overline{y}^{k-1} - \alpha\left[\nabla g(\overline{y}^{k-1}) + e^k\right]\} \\ \overline{y}^k & = \overline{x}^k + \beta_k\left(\overline{x}^k - \overline{x}^{k-1}\right) \end{cases} \tag{4.19}$$

where $\overline{x}^k = \frac{1}{m}\sum_{i=1}^m x_i^k, \overline{y}^k = \frac{1}{m}\sum_{i=1}^m y_i^k$, *and* $prox^{\alpha}_{h,\varepsilon}\{\cdot\}$ *is as defined in* (4.2). *Moreover, we have*

$$\left\|e^k\right\| \leq \frac{L}{m}\sum_{i=1}^m \left\|y_i^{k-1} - \overline{y}^{k-1}\right\| \tag{4.20}$$

$$\varepsilon_k \leq \frac{2G}{m}\sum_{i=1}^m \left\|\hat{q}_i^k - \overline{q}^k\right\| + \frac{1}{2\alpha}\left(\frac{1}{m}\sum_{i=1}^m \left\|\hat{q}_i^k - \overline{q}^k\right\|\right)^2. \tag{4.21}$$

Proof. By taking the average of (4.15a), we can see that

$$\overline{q}^k = \overline{y}^{k-1} - \alpha(\nabla g(\overline{y}^{k-1}) + e^k)$$

where, similar to (4.8) in the gradient method of the previous section,

$$e^k = \frac{1}{m}\sum_{i=1}^m \left[\nabla g_i(y_i^{k-1}) - \nabla g_i(\overline{y}^{k-1})\right],$$
$$\left\|e^k\right\| \leq \frac{L}{m}\sum_{i=1}^m \left\|y_i^{k-1} - \overline{y}^{k-1}\right\|.$$

Let

$$\tilde{x}^k = \mathrm{prox}_h^{\alpha}\{\overline{q}^k\} = \operatorname*{argmin}_x \left\{h(x) + \frac{1}{2\alpha}\left\|x - \overline{q}^k\right\|^2\right\}$$

denote the result of the exact centralized proximal step. Then $\overline{x}^k = \frac{1}{m}\sum_{i=1}^m x_i^k = \frac{1}{m}\sum_{i=1}^m \mathrm{prox}_h^{\alpha}\{\hat{q}_i^k\}$, the result of the proximal step in the distributed method, can be seen as an approximation of $\tilde{x}^k$.

We next relate $\tilde{x}^k$ and $\overline{x}^k$ by formulating the latter as an inexact proximal step with error ε_k. A simple algebraic expansion gives

$$\begin{aligned}
&h(\overline{x}^k) + \frac{1}{2\alpha}\left\|\overline{x}^k - \overline{q}^k\right\|^2 \\
\leq &h(\tilde{x}^k) + G\left\|\overline{x}^k - \tilde{x}^k\right\| + \frac{1}{2\alpha}\left[\left\|\tilde{x}^k - \overline{q}^k\right\|^2 + 2\left\langle \tilde{x}^k - \overline{q}^k, \overline{x}^k - \tilde{x}^k\right\rangle + \left\|\overline{x}^k - \tilde{x}^k\right\|^2\right] \\
= &\min_{z\in\mathbb{R}^d}\left\{h(z) + \frac{1}{2\alpha}\left\|z - \overline{q}^k\right\|^2\right\} + \left\|\overline{x}^k - \tilde{x}^k\right\|\left(G + \frac{1}{\alpha}\left\|\tilde{x}^k - \overline{q}^k\right\|\right) + \frac{1}{2\alpha}\left\|\overline{x}^k - \tilde{x}^k\right\|^2
\end{aligned}$$

where in the inequality we used the convexity of $h(x)$ and the bound on the subgradient $\partial h(\overline{x}^k)$ to obtain $h(\overline{x}^k) \leq h(\tilde{x}^k) + G\left\|\overline{x}^k - \tilde{x}^k\right\|$; and in the equality, we used the fact that by definition, $\tilde{x}^k$ is the optimizer of $h(x) + \frac{1}{2\alpha}\left\|x - \overline{q}^k\right\|^2$.

With this expression, we can write

$$\overline{x}^k = \mathrm{prox}^{\alpha}_{h,\varepsilon_k}\left\{\overline{q}^k\right\}$$

where

$$\varepsilon_k = \left\|\overline{x}^k - \tilde{x}^k\right\|\left(G + \frac{1}{\alpha}\left\|\tilde{x}^k - \overline{q}^k\right\|\right) + \frac{1}{2\alpha}\left\|\overline{x}^k - \tilde{x}^k\right\|^2.$$

By Proposition 4, $\tilde{x}^k = \mathrm{prox}^{\alpha}_h\{\overline{q}^k\}$ also implies $\frac{1}{\alpha}\left(\overline{q}^k - \tilde{x}^k\right) \in \partial h(\tilde{x}^k)$, and therefore its norm is bounded by G. As a result,

$$\varepsilon_k \leq 2G\left\|\overline{x}^k - \tilde{x}^k\right\| + \frac{1}{2\alpha}\left\|\overline{x}^k - \tilde{x}^k\right\|^2.$$

Combined with the nonexpansiveness of the proximal operator (Proposition 5),

$$\left\|\overline{x}^k - \tilde{x}^k\right\| \leq \frac{1}{m}\sum_{i=1}^m \left\|\mathrm{prox}^{\alpha}_h\{\hat{q}_i^k\} - \mathrm{prox}^{\alpha}_h\{\overline{q}^k\}\right\| \leq \frac{1}{m}\sum_{i=1}^m \left\|\hat{q}_i^k - \overline{q}^k\right\|,$$

we arrive at the desired expression.

□

Under this formulation, the two error sequences $\left\|e^k\right\|$ and ε_k have upper bounds in terms of $\frac{1}{m}\sum_{i=1}^m \left\|y_i^{k-1} - \overline{y}^{k-1}\right\|$ and $\frac{1}{m}\sum_{i=1}^m \left\|\hat{q}_i^k - \overline{q}^k\right\|$, respectively, which are in turn controlled by the two multi-step consensus stages. According to [8, Proposition 2], if

$\{k\|e^k\|\}$ and $\{k\sqrt{\varepsilon_k}\}$ are both summable, then the inexact proximal-gradient method exhibits the optimal exact convergence rate of $O(1/n^2)$. In the following subsections, we shall see that this is indeed the case.

4.3.2 Bounds on Iterates

Note that the upper bounds of $\|e^k\|$ and ε_k involve $\frac{1}{m}\sum_{i=1}^m \|y_i^{k-1} - \overline{y}^{k-1}\|$ and $\frac{1}{m}\sum_{i=1}^m \|\hat{q}_i^k - \overline{q}^k\|$, respectively, and the latter is in turn bounded by $\Gamma\gamma^k \sum_{j=1}^m \|q_j^k\|$ according to (4.16). We shall see that they are both polynomial-geometric sequences.

As in the multi-step gradient method, we first give some helpful expressions of the iterates:

Proposition 12. *(Recursive expressions of iterates)*

Let the sequences $\{x_i^k\}_{k=1}^\infty, \{y_i^k\}_{k=1}^\infty, \{q_i^k\}_{k=1}^\infty, \{\hat{q}_i^k\}_{k=1}^\infty, i = 1, ..., m$, *be generated as in Algorithm 4.15, with a constant step size* $\alpha_k = \alpha$. *Then we have, for every* $k \geq 2$,

(a) $\sum_{i=1}^m \|q_i^{k+1}\| \leq \sum_{i=1}^m \|q_i^k\| + 2\alpha mG + \sum_{i=1}^m \|x_i^k - x_i^{k-1}\|$

(b) $\sum_{i=1}^m \|x_i^k - x_i^{k-1}\| \leq 2m\Gamma \sum_{l=1}^{k-1} \gamma^l \sum_{j=1}^m \|q_j^l\| + (k-1)2\alpha mG$

(c) $\|y_i^k - \overline{y}^k\| \leq 4\Gamma\gamma^k \sum_{i=1}^m \|q_i^k\| + 2\Gamma\gamma^{k-1} \sum_{i=1}^m \|q_i^{k-1}\|$

Proof. (a) Taking norm of (4.15a) and summing over i, we have

$$\sum_{i=1}^m \|q_i^k\| = \sum_{i=1}^m \|y_i^{k-1} - \alpha\nabla g_i(y_i^{k-1})\| \leq \sum_{i=1}^m \|y_i^{k-1}\| + \alpha mG \tag{4.22}$$

According to (4.15d), we have $y_i^{k-1} = x_i^{k-1} + \beta_{k-1}(x_i^{k-1} - x_i^{k-2})$, and by (4.18), we have $\|x_i^{k-1}\| - \|\hat{q}_i^{k-1}\| \leq \|x_i^{k-1} - \hat{q}_i^{k-1}\| \leq \alpha G$. Therefore,

$$\|y_i^{k-1}\| \leq \|\hat{q}_i^{k-1}\| + \alpha G + \beta_{k-1}\|x_i^{k-1} - x_i^{k-2}\| \tag{4.23}$$

Finally, we use (4.15b), which states that $\hat{q}_i^{k-1} = \sum_{j=1}^m \hat{\lambda}_{ij}^{k-1} q_j^{k-1}$ is a convex

combination of $\{q_j^{k-1}\}_{j=1}^m$, so

$$\sum_{i=1}^{m} \left\|\hat{q}_i^{k-1}\right\| \leq \sum_{i=1}^{m} \left\|q_i^{k-1}\right\| \tag{4.24}$$

Substituting (4.23)-(4.24) back in (4.22), we have

$$\sum_{i=1}^{m} \left\|q_i^k\right\| \leq \sum_{i=1}^{m} \left\|q_i^{k-1}\right\| + 2\alpha mG + \beta_{k-1} \sum_{i=1}^{m} \left\|x_i^{k-1} - x_i^{k-2}\right\|$$

Finally, we omit $\beta_{k-1} \leq 1$, and increment the indices by 1 so that the expression is applicable to $k \geq 2$.

(b) Starting with (4.17) and applying (4.15b), (4.15a), (4.15d) in order, we have

$$\begin{aligned}
x_i^k &= \hat{q}_i^k - \alpha z_i^k \\
&= \sum_{j=1}^{m} \hat{\lambda}_{ij}^k q_j^k - \alpha z_i^k \\
&= \sum_{j=1}^{m} \hat{\lambda}_{ij}^k \left[y_j^{k-1} - \alpha \nabla g_j(y_j^{k-1})\right] - \alpha z_i^k \\
&= \sum_{j=1}^{m} \hat{\lambda}_{ij}^k \left[x_j^{k-1} + \beta_{k-1}(x_j^{k-1} - x_j^{k-2})\right] - \sum_{j=1}^{m} \hat{\lambda}_{ij}^k \alpha \nabla g_j(y_j^{k-1}) - \alpha z_i^k.
\end{aligned}$$

Subtracting x_i^{k-1} from the previous expression and taking the sum of the norm, we have

$$\sum_{i=1}^{m} \left\|x_i^k - x_i^{k-1}\right\| \leq \sum_{i=1}^{m} \sum_{j=1}^{m} \hat{\lambda}_{ij}^k \left\|x_j^{k-1} - x_i^{k-1}\right\| + \beta_{k-1} \sum_{j=1}^{m} \left\|x_j^{k-1} - x_j^{k-2}\right\| + 2\alpha mG \tag{4.25}$$

where we used the convexity of the norm operator along with the fact that $\sum_{i=1}^m \hat{\lambda}_{ij}^k = 1$.

Now consider the term $\sum_{i=1}^m \sum_{j=1}^m \hat{\lambda}_{ij}^{k-1} \left\|x_j^{k-1} - x_i^{k-1}\right\|$ in the expression above.

By the nonexpansiveness of the proximal operator,

$$\left\|x_j^{k-1} - x_i^{k-1}\right\| \leq \left\|\hat{q}_j^{k-1} - \hat{q}_i^{k-1}\right\|$$

Applying the same argument as in (4.11), we have

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \hat{\lambda}_{ij}^{k-1}\left\|\hat{q}_j^{k-1} - \hat{q}_i^{k-1}\right\| \leq 2\sum_{i=1}^{m}\left\|\hat{q}_i^{k-1} - \overline{q}^{k-1}\right\|$$

and finally, we can bound the right-hand side with (4.16). As a result,

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \hat{\lambda}_{ij}^{k-1}\left\|x_j^{k-1} - x_i^{k-1}\right\| \leq 2m\Gamma\gamma^{k-1}\sum_{j=1}^{m}\left\|q_j^{k-1}\right\|. \tag{4.26}$$

Substituting this back to (4.25),

$$\begin{aligned}\sum_{i=1}^{m}\left\|x_i^k - x_i^{k-1}\right\| &\leq 2m\Gamma\gamma^{k-1}\sum_{j=1}^{m}\left\|q_j^{k-1}\right\| + \beta_{k-1}\sum_{i=1}^{m}\left\|x_i^{k-1} - x_i^{k-2}\right\| + 2\alpha mG \\ &\leq \sum_{l=1}^{k-1}\left(2m\Gamma\gamma^l\sum_{j=1}^{m}\left\|q_j^l\right\| + 2\alpha mG\right)\end{aligned}$$

where the final line is due to recursion, and omitting $\beta_l \leq 1$ for $l > 1$ while using $\beta_1 = 0$ to eliminate the tailing term $\sum_{i=1}^{m}\left\|x_i^1 - x_i^0\right\|$. This is the desired expression.

(c) By (4.15d),

$$y_i^k - \overline{y}^k = (1+\beta_k)(x_i^k - \overline{x}^k) - \beta_k(x_i^{k-1} - \overline{x}^{k-1})$$

Note also that

$$\sum_{i=1}^{m}\left\|x_i^k - \overline{x}^k\right\| \leq \frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{m}\left\|x_i^k - x_j^k\right\| \leq 2\Gamma\gamma^k\sum_{j=1}^{m}\left\|q_j^k\right\|,$$

by a similar reasoning as that for (4.26). As a result,

$$\begin{aligned}\left\|y_i^k - \overline{y}^k\right\| &\leq (1+\beta_k)\left\|x_i^k - \overline{x}^k\right\| + \beta_k\left\|x_i^{k-1} - \overline{x}^{k-1}\right\| \\ &\leq (1+\beta_k)2\Gamma\gamma^k \sum_{j=1}^{m}\left\|q_j^k\right\| + \beta_k 2\Gamma\gamma^{k-1}\sum_{j=1}^{m}\left\|q_j^{k-1}\right\|.\end{aligned}$$

Omitting $\beta_k < 1$ gives statement (c).

□

Next, we use the expressions established in Proposition 12 to show that $\sum_{j=1}^{m}\left\|q_i^k\right\|$ is bounded by second-order polynomials of k.

Lemma 5. *(Polynomial Bounds on $\sum_{j=1}^{m}\left\|q_i^k\right\|$)*

Let the sequences $\{x_i^k\}_{k=1}^{\infty}, \{y_i^k\}_{k=1}^{\infty}, \{q_i^k\}_{k=1}^{\infty}, \{\hat{q}_i^k\}_{k=1}^{\infty}, i = 1, ..., m$, be generated as in Algorithm 4.15, with a constant step size $\alpha_k = \alpha$ and initial points $\{y_i^0\}_{i=1}^{m}$. Then there exists scalars C_q, C_q', C_q'' such that for $k \geq 2$,

$$\sum_{i=1}^{m}\left\|q_i^k\right\| \leq C_q + C_q' k + C_q'' k^2.$$

Proof. We proceed by induction on k. First, we show that the result holds for $k = 2$ by choosing $C_q = \sum_{i=1}^{m}\left\|q_i^2\right\|$. It suffices to show that, given the initial points y_j^0, $\sum_{j=1}^{m}\left\|q_j^2\right\|$ is bounded.

Indeed, by (4.22),

$$\sum_{i=1}^{m}\left\|q_i^1\right\| \leq \sum_{i=1}^{m}\left\|y_i^0\right\| + 2\alpha m G < \infty$$

and

$$\begin{aligned}\sum_{j=1}^{m}\left\|q_i^2\right\| &\leq \sum_{j=1}^{m}\left\|y_i^1\right\| + 2\alpha mG \\ &= \sum_{j=1}^{m}\left\|x_i^1\right\| + 2\alpha mG \\ &\leq \sum_{j=1}^{m}\left\|\hat{q}_i^1\right\| + 4\alpha mG \leq \sum_{j=1}^{m}\left\|q_i^1\right\| + 4\alpha mG < \infty\end{aligned}$$

where the second line is due to the fact that $\beta_1 = 0$ so $y_i^1 = x_i^1$, and the third line is because of (4.17) and (4.24).

Now suppose the result holds for some positive integer $k \geq 2$. We show that it also holds for $k+1$.

Substituting the induction hypothesis for k into Proposition 12(b), we have

$$\sum_{i=1}^{m}\left\|x_i^k - x_i^{k-1}\right\| \leq \sum_{l=1}^{k-1}\left(2m\Gamma\gamma^l\left(C_q + C_q'k + C_q''k^2\right) + 2\alpha mG\right).$$

By Proposition 8 and (4.3), there exists constants $S_0^\gamma, S_1^\gamma, S_2^\gamma$ such that

$$\sum_{l=0}^{\infty}\gamma^l\left(C_q + C_q'k + C_q''k^2\right) \leq C_qS_0^\gamma + C_q'S_1^\gamma + C_q''S_2^\gamma.$$

Therefore,

$$\sum_{i=1}^{m}\left\|x_i^k - x_i^{k-1}\right\| \leq 2m\Gamma\left(C_qS_0^\gamma + C_q'S_1^\gamma + C_q''S_2^\gamma\right) + 2\alpha mG(k-1).$$

Proposition 12(a) and the induction hypothesis then gives us

$$\sum_{i=1}^{m}\left\|q_i^{k+1}\right\| \leq C_q + C_q'k + C_q''k^2 + 2\alpha mG + 2m\Gamma\left(C_qS_0^\gamma + C_q'S_1^\gamma + C_q''S_2^\gamma\right) + 2\alpha mG(k-1)$$

Comparing coefficients, we see that the right-hand side can be bounded by

$$C_q + C_q'(k+1) + C_q''(k+1)^2 = C_q + C_q'k + C_q''k^2 + (2C_q''k + C_q' + C_q'')$$

if

$$2\alpha mG \leq C_q'',$$

for the coefficient of k, and

$$2m\Gamma\left(C_q S_0^\gamma + C_q' S_1^\gamma + C_q'' S_2^\gamma\right) \leq C_q' + C_q'',$$

for the constant coefficient. Therefore, the induction hypothesis holds for $k+1$ if we take

$$\begin{aligned} C_q &= \sum_{i=1}^{m} \|q_i^2\|, \\ C_q' &= \frac{2m\Gamma C_q S_0^\gamma + (2m\Gamma S_2^\gamma - 1)C_q''}{2m\Gamma S_1^\gamma - 1}, \\ C_q'' &= 2\alpha mG. \end{aligned}$$

□

4.3.3 Convergence Rate

We now apply Lemma 5 on the error sequences in (4.19) to show that $\{k\|e^k\|\}$ and $\{k\sqrt{\varepsilon_k}\}$ are polynomial-geometric sequences, thus summable:

Lemma 6. *(Summability of $\{k\|e^k\|\}$ and $\{k\sqrt{\varepsilon_k}\}$)*

In the formulation (4.19)*, where*

$$\begin{aligned} \|e^k\| &\leq \frac{L}{m} \sum_{i=1}^{m} \|y_i^{k-1} - \overline{y}^{k-1}\|, \\ \varepsilon_k &\leq \frac{2G}{m} \sum_{i=1}^{m} \|\hat{q}_i^k - \overline{q}^k\| + \frac{1}{2\alpha} \left(\frac{1}{m} \sum_{i=1}^{m} \|\hat{q}_i^k - \overline{q}^k\| \right)^2, \end{aligned}$$

we have

(a) $\sum_{k=1}^{\infty} k\|e^k\| < \infty$

(b) $\sum_{k=1}^{\infty} k\sqrt{\varepsilon_k} < \infty$

Proof. In both cases, it suffices to show that the sequence is a polynomial-geometric sequence; then, it is summable by Proposition 8.

(a) By Proposition 12(c),

$$\frac{1}{m}\sum_{i=1}^{m}\left\|y_i^k-\overline{y}^k\right\| \le 4\Gamma\gamma^k\sum_{i=1}^{m}\left\|q_i^k\right\| + 2\Gamma\gamma^{k-1}\sum_{i=1}^{m}\left\|q_i^{k-1}\right\|$$

and by Lemma 5(a),

$$\sum_{i=1}^{m}\left\|q_i^k\right\| \le C_q + C_q'k + C_q''k^2.$$

Therefore,

$$k\left\|e^k\right\| \le 4L\Gamma\gamma^k k\left(C_q + C_q'k + C_q''k^2\right) + 2L\Gamma\gamma^{k-1}k\left(C_q + C_q'(k-1) + C_q''(k-1)^2\right),$$

which is a polynomial-geometric sequence.

(b) Recall (4.16),

$$\left\|\hat{q}_i^k-\overline{q}^k\right\| \le \Gamma\gamma^k\sum_{j=1}^{m}\left\|q_j^k\right\|,$$

and Lemma 5(a),

$$\sum_{j=1}^{m}\left\|q_j^k\right\| \le C_q + C_q'k + C_q''k^2.$$

Therefore,

$$\varepsilon_k \le 2G\Gamma\gamma^k\left(C_q + C_q'k + C_q''k^2\right) + \frac{1}{2\alpha}\left[\Gamma\gamma^k\left(C_q + C_q'k + C_q''k^2\right)\right]^2.$$

Using the fact that $\sqrt{a+b} \le \sqrt{a} + \sqrt{b}$ for all nonnegative real numbers a, b, we have

$$\begin{aligned}\sqrt{\varepsilon_k} &\le \sqrt{2G\Gamma\gamma^k\left(C_q + C_q'k + C_q''k^2\right)} + \frac{1}{\sqrt{2\alpha}}\left[\Gamma\gamma^k\left(C_q + C_q'k + C_q''k^2\right)\right] \\ &\le \sqrt{2G\Gamma}\sqrt{\gamma}^k\left(\sqrt{C_q} + \sqrt{C_q'}k + \sqrt{C_q''}k\right) + \frac{\Gamma}{\sqrt{2\alpha}}\gamma^k\left(C_q + C_q'k + C_q''k^2\right)\end{aligned}$$

where in the last line we used the fact that $\sqrt{k} \le k$ for all $k \ge 1$. This is a

polynomial-geometric sequence. Therefore, $k\sqrt{\varepsilon_k}$ is also a polynomial-geometric sequence.

□

Using the lemma above, we can establish the convergence rate of our distributed proximal-gradient method, with the same proof as that of the distributed gradient method in Section 4.2.3:

Theorem 3. *(Convergence rate of the distributed proximal-gradient method with multi-step consensus)*

Let Algorithm (4.15) *be such that k communication steps are performed within the consensus stage* (4.15b) *at iteration k, i.e.*

$$\left|\hat{\lambda}_{ij}^{k} - \frac{1}{m}\right| \leq \Gamma\gamma^{k}.$$

Then, for all $t \geq 1$, where t is the total number of communication steps taken, we have

$$f(\overline{x}^{t}) - f(x^{*}) = O(1/t).$$

4.4 Beyond $O(1/t)$

In the previous sections, we saw that taking k communication steps in the k-th iteration results in the summability of error sequences $\{k\|e^{k}\|\}$ and $\{k\sqrt{\varepsilon_k}\}$. A natural question arises: can we do better? In particular, will the error sequences still converge if we took less than k communication steps in the k-th iteration? We address the question in this section.

Let s_k be the number of communication steps taken in the consensus stage at iteration k. In our methods presented earlier, $s_k = k$. We wish to find a smaller choice of s_k that would reduce the steps required for each iteration, while preserving the guarantee for exact convergence.

With s_k, we see that (4.5) for the gradient method can be written as

$$\left\|w_i^k - \overline{w}^k\right\| \leq \Gamma\gamma^{s_k} \sum_{j=1}^{m} \left\|y_j^k\right\|,$$

and (4.16) for the proximal-gradient method can be written as

$$\left\|\hat{q}_i^k - \overline{q}^k\right\| \leq \Gamma\gamma^{s_k} \sum_{j=1}^{m} \left\|q_j^k\right\|.$$

Therefore, Proposition 10(b) and Proposition 12(b) becomes

$$\sum_{i=1}^{m} \left\|x_i^k - x_i^{k-1}\right\| \leq 2m\Gamma \sum_{l=1}^{k-2} \gamma^{s_l} \sum_{j=1}^{m} \left\|y_j^l\right\| + C_x + (k-2)5m\alpha G \text{ (gradient)}$$

$$\sum_{i=1}^{m} \left\|x_i^k - x_i^{k-1}\right\| \leq \sum_{l=1}^{k-1} 2m\Gamma\gamma^{s_l} \sum_{j=1}^{m} \left\|q_j^l\right\| + (k-1)2\alpha mG \text{ (proximal-gradient)}$$

As a result, if we have the equivalent of Proposition 8 for s_k, i.e. if

$$\sum_{k=0}^{\infty} k^N \gamma^{s_k} < \infty$$

for any given $\gamma \in (0,1)$ and nonnegative integer N, then Lemmas 3 and 5 would hold, and so would Theorems 2 and 3.

Since $\sum_{k=0}^{\infty} k^a < \infty$ for $a < -1$, a sufficient condition for the above is

$$\gamma^{s_k} < k^{-N-1},$$

or equivalently,

$$s_k > \frac{-N-1}{\log\gamma} \log k.$$

Note that while this is at the order of $O(\log k)$, which is smaller than our previous choice of $s_k = k = O(k)$, the hidden constant depends on N. Fortunately, in our

case, we only require the condition to hold up to $N = 3$. Therefore, by choosing

$$s_k = \left\lceil \frac{4}{-\log \gamma} \log(k+1) \right\rceil, \tag{4.27}$$

the distributed first-order methods are guaranteed to converge with rate $O(1/n^2)$, where n is the iteration number.

The time it takes to complete iterations $1, ..., n$, which we denote by $T(n)$, is then

$$T(n) = \sum_{k=1}^{n} s_k = O(n \log n - n)$$

since $\int \log x dx = x(\log x - 1)$. Unfortunately, $T(n)$ has the form of what is known as the Lambert W function, for which there is no explicit inverse expression. Therefore, we can only express the convergence rate as

$$f(\overline{x}^t) - f(x^*) = O(1/(T^{-1}(t))^2)$$

which we know is better than $O(1/t)$.

In closing, we remark that the improved choice of s_k in (4.27) requires the knowledge of γ, which may not be readily available if detailed information or performance guarantees of the communication network is unknown. In such cases, the safe choice of $s_k = k$ is still recommended.

Chapter 5

Numerical Experiments

This chapter presents the performance of our methods applied to a machine learning task on a benchmark dataset for text categorization. Convergence results verify bounds and properties given in our theoretical analysis, demonstrating the potential of our distributed first-order methods in real-world applications.

5.1 Setup

In this chapter, we present experimental results on the 20 Newsgroups dataset.

The 20 Newsgroups dataset [26, 27] consists of about 20,000 news articles, each labelled with one news topic out of 20; we use the preprocessed version found on [28]. Since the articles are evenly distributed among the 20 topics, we arbitrarily pick topic 1 as the label to learn in this experiment.

The task is to perform L_1-regularized logistic regression so as to learn the classification model for the chosen topic. Specifically, we wish to minimize

$$f(x) = \frac{1}{N}\sum_{j=1}^{N} \log\left(1 + \exp(-b_j \langle a_j, x\rangle)\right) + \lambda\|x\|_1$$

where N is the total number of news articles, a_j is the 8615-dimensional feature vector of article j, and b_j is its the label for the chosen topic, which is equal to 1 if this article belongs to the topic, and -1 otherwise. x contains parameters of the classification

model that we wish to learn, and $f(x)$ is its corresponding regularized loss function. It has both a smooth and a nonsmooth component, so it is appropriate to use our proximal gradient method to search for the optimizer. The reader is referred to [29] for details on logistic regression and gradient descent.

The data is partitioned into a training set and a test set of 60% and 40%, respectively. We shall only concern ourselves with the training set, since our major focus is to compare the rate with which various optimization algorithms find the optimal solution to the above function in the training phase, rather than evaluate the performance of a new machine learning algorithm. For the same reason, we do not λ, but simply chose $\lambda = 0.005$ since the convergence properties of the proximal-gradient methods are clearly demonstrated with this level of regularization.

Our experiment focuses on distributed proximal gradient methods for logistic regression. Instead of having a single processing unit, we consider the case where data is distributed across a network of m data centers. Each data center has the following private objective function:

$$f_i(x_i) = \frac{1}{|N_i|} \sum_{j \in N_i} \log\left(1 + \exp(-b_j \langle a_j, x_i \rangle)\right) + \lambda \|x_i\|_1$$

where $x_i, i = 1, ..., m$ is a local estimate of the global classification model, and N_i is its data set.

We divide our data into $m = 10$ data centers, each of which contain 1129 data points. In order to make the private functions as different from each other as possible, the data was partitioned roughly according to the labels, so that each data center contains data of similar labels. The power of distributed methods is manifest in that even though not all data centers have information pertaining to the label of interest, each of them are still able to learn the model for it.

Randomness in the communication network is simulated as follows: we generate 5 sets of weights from each of the graphs illustrated in Figure 5-1. For every communication step, the program randomly selects one of the 10 communication networks. The resulting communication pattern satisfies the requirements of Assumption 2.

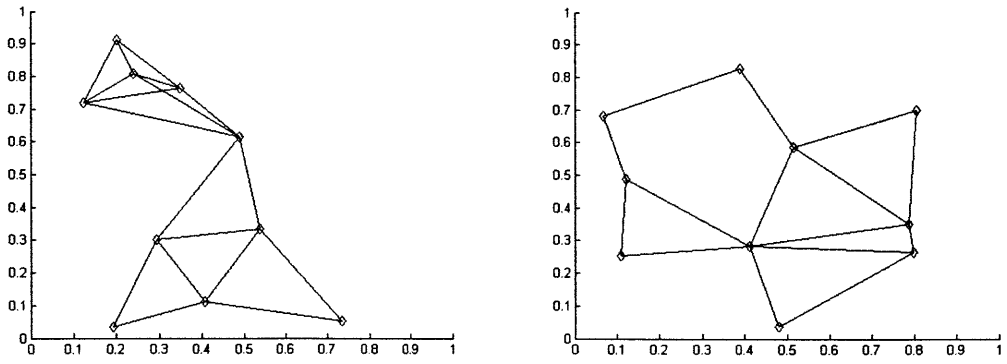


Figure 5-1: Underlying Communication Networks

5.2 Experiments and Results

5.2.1 Step Size Choices for Single-Step Consensus

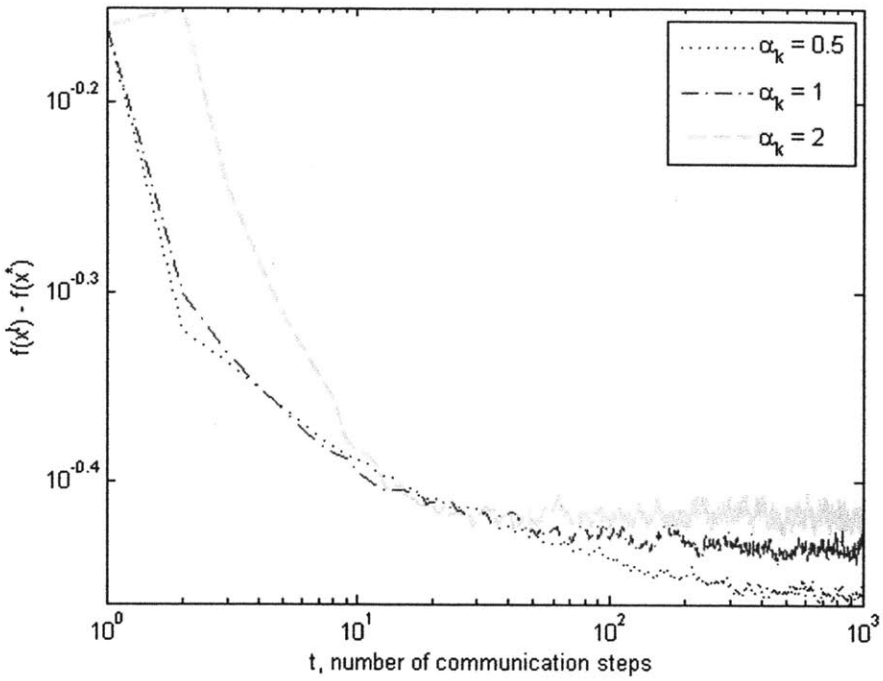
This experiment explores the effect of step size choices in the single-step consensus scheme described in Chapter 3. The performance is tested both for different constant step sizes, and for diminishing step sizes of the class $\alpha_k = 1/k^a$, where $a > 0$.

Figure 5-2(a) illustrates the effect of the constant step size on the error neighborhood. As expected, the function value converges to within an error neighborhood of the optimal value at rate $O(1/n)$, and the size of this neighborhood increases with α.

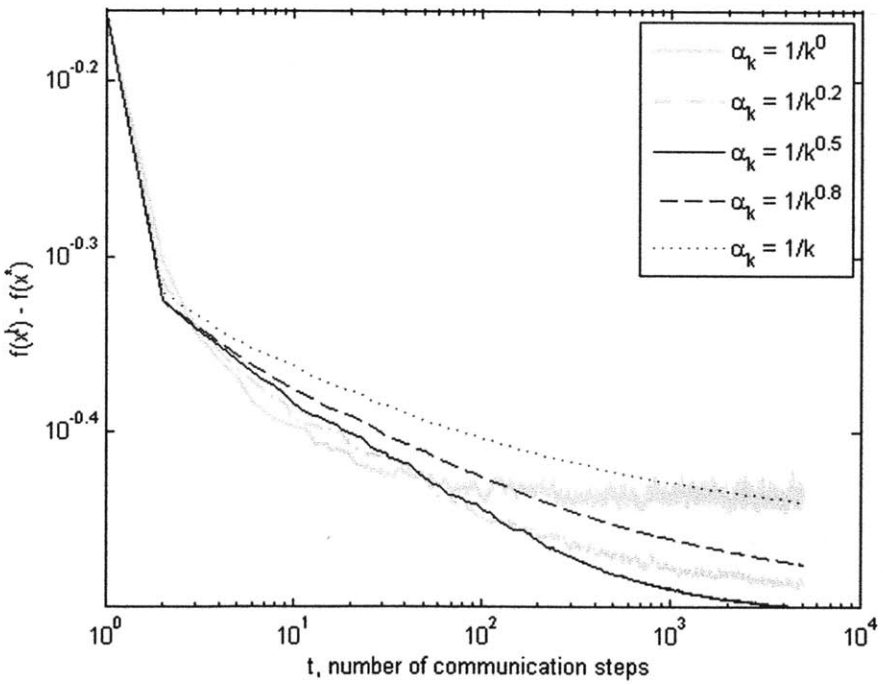
On the other hand, the optimal choice of parameter a for the class of diminishing step sizes $\alpha_k = 1/k^a$ is clear from Figure 5-2(b), where the rate of exact convergence is the fastest when $a = 0.5$.

5.2.2 Convergence Rate Comparison for Single- and Multi-Step Consensus

In this experiment, we verify and compare the convergence rates of methods developed in Chapters 2 and 3, namely, the basic and accelerated proximal-gradient methods with single-step consensus (3.1), and the accelerated proximal-gradient method with multi-step consensus (4.15). To illustrate the need for an additional consensus stage



(a) Constant step size rules



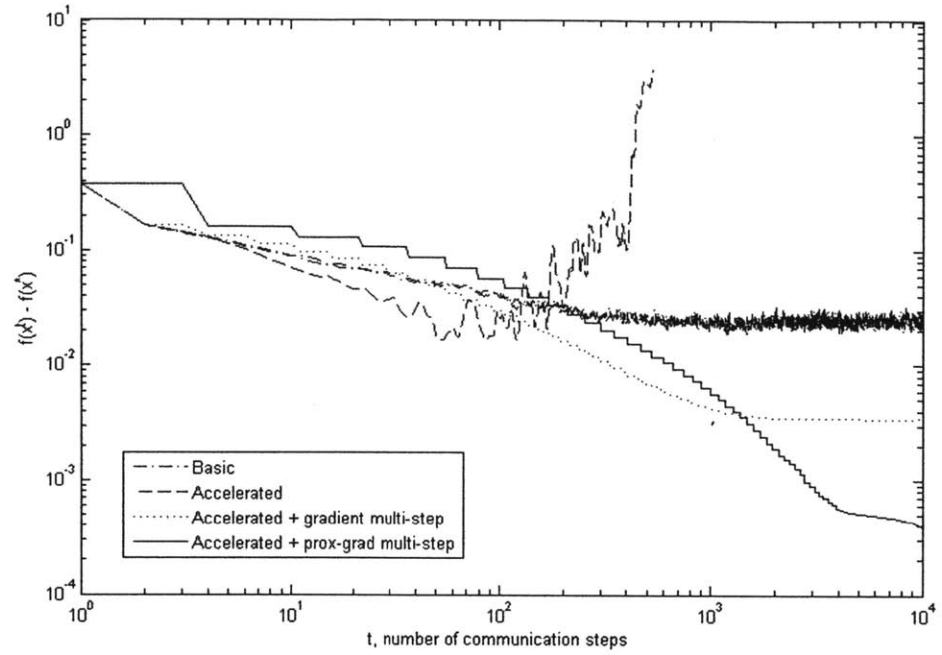
(b) Diminishing step size rules

Figure 5-2: Performance comparison under two step size rules

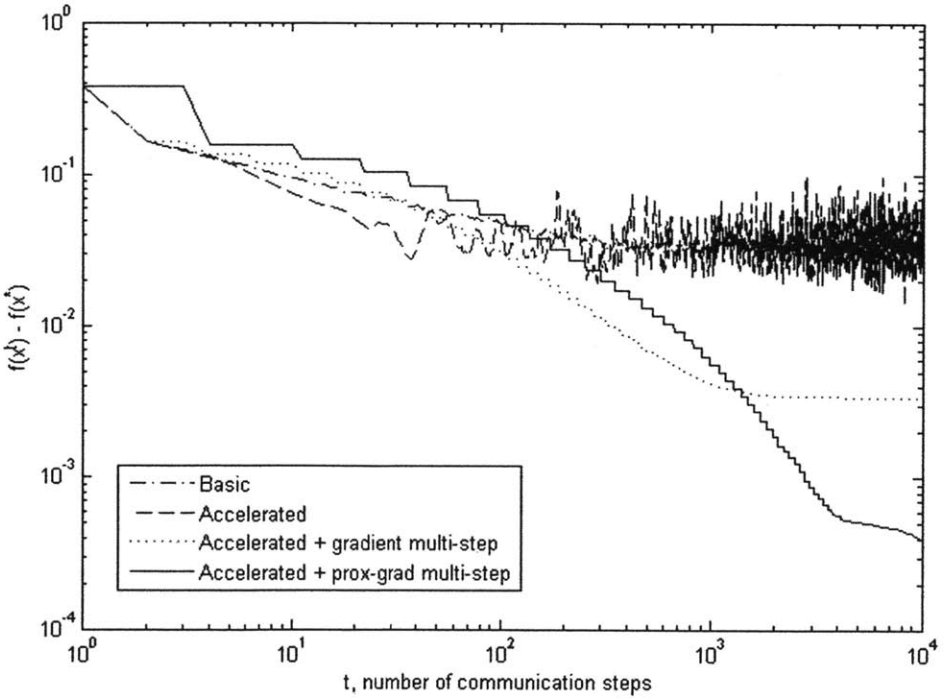
in (4.15), we also include the accelerated proximal-gradient method with only one multi-step consensus stage after the gradient method (4.13), which we refer to as "gradient multi-step" for short.

We explore two conditions for communication weights, the results for which are presented in Figure 5-3. In each communication step, the method randomly chooses a weight matrix from the pool of 10 weight matrices as mentioned previously. For (a), we simply generate weight matrices by setting $\eta = 0.2$, i.e. every link has a weight no smaller than 0.2. For (b), we take convex combinations of matrices in (b) and the identity matrix so as to give sufficient weights on the diagonal, thus ensuring the matrices to be positive-definite.

In both cases, as expected, the basic proximal-gradient single-step method converges to an error neighborhood at rate $O(1/t)$, as does the gradient multi-step method. The accelerated method with single-step consensus, in (a), fails to converge even to an error neighborhood; instead, the iterates grow unbounded, as explained in Section 3.3; in (b), where the weight matrix is always positive-definite, the accelerated single-step method converges to an error neighborhood. Finally, our accelerated multi-step method attains exact convergence with rate $O(1/t)$ in both cases, outperforming the convergence property of all methods above with a comparable rate.



(a) Arbitrary weight matrix with $\eta = 0.2$



(b) Positive-definite weight matrix

Figure 5-3: Performance comparison under two conditions for communication weight matrices

Chapter 6

Conclusions

This thesis develops a framework for the analysis of distributed proximal-gradient methods in multi-agent networks. We showed that it is possible for an agent to optimize the global objective function using only local information both from its private function and from the communication network. The basic method with a constant step size does not convergence exactly, and the convergence rate is $O(1/\sqrt{n})$; on the other hand, with a diminishing step size rule of the form $\alpha_k = 1/k^a$, the method achieves exact convergence at the rate of $O(\log n/\sqrt{n})$ with the optimal choice of a being $1/2$.

We also presented a new method that combined Nesterov-type acceleration techniques and multi-step communication. While the time required to execute one iteration is longer than the single-step scheme due to the extra effort required of communication, it improves the quality of consensus, thus guaranteeing exact convergence. Moreover, with the help of acceleration techniques, this method achieves a convergence rate of $O(1/t)$, with t being the number of communication steps executed. Simulation results also verified our theoretical findings, and show that the method with multi-step consensus can be superior in robustness where single-step consensus methods fail to converge.

There are several potential directions for future work. First of all, it is of interest to characterize conditions for network communication under which methods with single-step consensus have stable performance, as opposed to the oscillating or

divergent behavior seen in simulation results. In addition, to understand the nature of distributed problems, it would be helpful to determine the lower bound on the convergence rate of this class of distributed methods, similar to the optimal achievable rates obtained in centralized methods. Finally, other variations in the distributed proximal-gradient method may be considered, including asynchronous updates, objective functions that are time-varying or stochastic, or even online optimization settings where information is revealed over time.

Bibliography

[1] A. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *Comptes Rendus de l'Académie des Sciences Paris*, vol. 25(Série I), 1847.

[2] D. P. Bertsekas, *Nonlinear Programming.* Athena Scientific.

[3] Y. Nesterov, "A method for solving a convex programming problem with convergence rate $o(1/k^2)$," *Soviet Math. Dokl.*, vol. 27, no. 2, pp. 372–376, 1976.

[4] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course.* Kluwer Academic Publishers.

[5] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[6] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 183–202, March 2009.

[7] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, pp. 1574–1609, 2009.

[8] M. Schmidt, N. L. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," *CoRR*, vol. abs/1109.2415, 2011.

[9] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *CORE Discussion Paper (2011/02)*, 2011.

[10] J. N. Tsitsiklis, *Problems in Decentralized Decision Making and Computation.* PhD thesis, Department of EECS, MIT, 1984.

[11] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Transactions on Automatic Control*, no. 9, pp. 803–812, 1986.

[12] D. P. Bertsekas, "Incremental gradient, subgradient, and proximal methods for convex optimization: A survey," tech. rep., Laboratory for Information and Decision Systems, MIT.

[13] A. Nedic and A. Ozdaglar, "On the rate of convergence of distributed asynchronous subgradient methods for multi-agent optimization," in *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 4711–4716, 2007.

[14] S. S. Ram, A. Nedic, and V. V. Veeravalli, "Distributed subgradient projection algorithm for convex optimization," in *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, ICASSP '09, 2009.

[15] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, 2010.

[16] I. Lobel and A. Ozdaglar, "Distributed subgradient methods for convex optimization over random networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 6, pp. 1291–1306, 2011.

[17] B. Touri, A. Nedic, and S. Ram, "Asynchronous stochastic convex optimization over random networks: Error bounds," in *Information Theory and Applications Workshop (ITA), 2010*, pp. 1 –10, 31 2010-feb. 5 2010.

[18] Y. Sun, A. Speranzon, and P. G. Mehta, "Convergence rate for distributed optimization methods: Novel bounds and distributed step size computation," 2012.

[19] R. L. G. Cavalcante, N. R. J. A. Rogers, , and I. Yamada, "Distributed asymptotic minimization of sequences of convex functions by a broadcast adaptive subgradient method," *IEEE Journal of Selected Topics in Signal Processing*, 2011.

[20] A. Agarwal and J. C. Duchi, "Distributed delayed stochastic optimization," *arXiv: 1104. 5525v1*.

[21] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Fast distributed gradient methods," *arXiv: 1112. 2972v1*, 2011.

[22] J. Duchi, A. Agarwal, and M. Wainwright, "Dual averaging for distributed optimization: Convergence and network scaling," *IEEE Transactions on Automatic Control*, 2012.

[23] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, 2011.

[24] G. Mateos, J. Bazerque, and G. Giannakis, "Distributed sparse linear regression," *Signal Processing, IEEE Transactions on*, vol. 58, pp. 5262 –5276, oct. 2010.

[25] D. P. Palomar and Y. C. Eldar, *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

[26] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the Twelfth International Conference on Machine Learning*, pp. 331–339, 1995.

[27] J. Rennie, "20 newsgroups." `http://people.csail.mit.edu/jrennie/20Newsgroups/`.

[28] R. F. Corra, "Text categorization datasets." `http://sites.google.com/site/renatocorrea02/textcategorizationdatasets`.

[29] J. Rennie, "Logistic regression." people.csail.mit.edu/jrennie/writing/lr.pdf.