DISTRIBUTED OPTIMIZATION WITH APPLICATIONS TO SENSOR NETWORKS
AND MACHINE LEARNING

BY

KUNAL SRIVASTAVA

DISSENTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Systems and Entrepreneurial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:
Assistant Professor Angelia Nedić, Chair
Associate Professor Dušan Stipanović, Co-Director of Research
Professor P. R. Kumar
Associate Professor R. S. Sreenivas
Abstract

This dissertation deals with developing optimization algorithms which can be distributed over a network of computational nodes. Specifically we develop distributed algorithms for the special class when the optimization problem of interest has a separable structure. In this case the objective function can be written as a sum of local convex objective functions. Each computational node has knowledge of its own local objective function and its local constraint set and needs to cooperatively solve the optimization problem under this information constraint. Furthermore we consider the case when the communication topology of the nodes is dynamic in nature. Recently, there has been a lot of interest in the so called “consensus” algorithms which has been shown to be remarkable robust to dynamic communication topology. Our algorithms leverage the robustness properties of consensus algorithm to compute the optimal solution in a distributed manner. We propose algorithms which have guaranteed convergence behavior in the presence of various forms of perturbations like communication noise, stochastic subgradient errors and stochastic communication topologies. This enables our algorithms to be useful in a wide class of application areas in sensor networks and machine learning. Specifically the consideration of stochastic subgradient errors enable our algorithms to be useful in an online setting, when the algorithm operates on streaming data. We adapt our algorithms for the binary classification problem in the support vector machine setting and show the behavior over a sample data set.

We further develop distributed algorithms for the min-max problem in a network. This formulation doesn’t readily fit the separable structure of the objective function discussed earlier. We develop an exact penalty based approach and an approach based on primal-dual iterative schemes. We show the applicability of the algorithms on a power allocation problem in cellular networks.
To my family, for their love and support.
Acknowledgements

First and foremost I express my deep sense of gratitude to my advisors Professor Angelia Nedić and Dušan Stipanović. They provided valuable guidance, motivation and support during the course of this research and have been great academic role models to look up to and emulate. I am also indebted to Professor Mark Spong for advising and supporting me during the early phase of my PhD. I would also like to thank Professor P.R. Kumar and Professor R.S. Sreenivas for serving on my doctoral committee.

I owe a great deal to all the wonderful teachers I had at University of Illinois. They have helped me develop tools which have held me in good stead. A special thanks to Professor P.R. Kumar, Professor Daniel Liberzon, Professor Tamer Başar, Professor Sean Meyn and Professor Richard Sowers.

I had the good fortune of meeting a lot of wonderful people at CSL who made CSL an ideal research environment. In particular I would like to thank Hemant Kowshik, Jayakrishnan Unnikrishnan, Arvind Ganesh, Jayanand Asok-Kumar, Anand Muralidhar, Ramakrishna Gummadi, Vijay Raman, Nikos Freris, Silvia Mastellone and Peter Hokayem. In addition I would like to thank Anil Kumar, Mamta Singh, Jagannathan Rajagopalan, Vivek Natarajan, Gayathri Mohan, Sophie Pwet, Anjan Raghunathan, Rajan Lakshmi Narasimha, Shankar Sivaramakrishnan, Karthik Jambunathan, Sreeram Kannan and Chaitanya Sathe for their friendship and helping me in various tangible and intangible ways.

My brothers Anurag and Neelabh have always been there despite the distance and finally my parents have been a pillar of strength providing me with their love, unwavering support and motivation and I owe all of this to them.
Table of Contents

Chapter 1 Introduction ................................................................. 1
1.1 Problem Outline ........................................................................ 2
1.2 Statement of Contribution and Organization .......................... 4

Chapter 2 Mathematical Preliminaries ............................................. 8
2.1 Notation and Terminology .......................................................... 8
2.2 Bregman Distance .................................................................... 10
2.3 Basic Results ............................................................................ 12

Chapter 3 Consensus Based Algorithm for Convex Feasibility Problems .... 13
3.1 Successive Orthogonal Projection .............................................. 14
   3.1.1 The Multiprojection Algorithm ............................................. 15
3.2 Problem Formulation ................................................................. 16
   3.2.1 Model Assumptions .............................................................. 18
3.3 Preliminary Results .................................................................. 19
3.4 Convergence Result ................................................................. 22
3.5 Subspace Constraints ............................................................... 23
3.6 Conclusion .............................................................................. 26

Chapter 4 Synchronous Algorithm for Distributed Constrained Stochastic Optimization ................................................................. 27
4.1 Incremental Subgradient Algorithm .......................................... 28
4.2 The Stochastic Optimization Problem ...................................... 29
4.3 Distributed Optimization Problems Arising In Networks .......... 29
   4.3.1 Consensus and Robust Estimation ...................................... 30
   4.3.2 Constrained Estimation ....................................................... 30
   4.3.3 Network Utility Maximization in a Wireless Network .......... 32
   4.3.4 Distributed Model Predictive Control ................................. 32
4.4 Main Algorithm ....................................................................... 34
   4.4.1 Assumptions ..................................................................... 35
4.5 Preliminary Results .................................................................. 36
4.6 Convergence Result ................................................................. 41
4.7 Conclusion .............................................................................. 43
Chapter 1

Introduction

There has been a sustained effort in the research community over the years to develop algorithms for distributed decision making and control. The need for distributed algorithms typically arise in two situations. In many cases all of the data needed to solve a problem is not located at a central node. This situation often arises in network applications consisting of multiple sensing and actuating entities like in sensor networks or in networked control systems. On the other hand in many computationally challenging problems using just one processor may be inefficient. In this case it is suitable to apply the technique of “divide and conquer” and make use of the emerging technology of multicore processors to develop efficient algorithms.

The main driver for these problems are the more application specific problems arising in wireless and sensor networks, transmission control protocols for the internet, distributed machine learning, multi-vehicle coordination and more recently social networks.

The two main mathematical abstractions which have been employed to address these problems are the problem of reaching consensus on the decision variables [1–5] in a network of computational agents and the problem of cooperative solution to distributed optimization problems [6–11]. The algorithms for reaching consensus have proven useful in a wide variety of contexts from formation control [3], distributed parameter estimation [12], [13], load balancing [14], to synchronization of Kuramoto oscillators [15]. The problem of distributed optimization, where the objective is to minimize a sum of convex functions appears widely in the context of wireless and sensor networks [16–18]. A more recent application area for distributed optimization is the problem of distributed machine learning. In many machine learning applications it is highly desirable to come up with distributed schemes to solve an optimization problem as the ubiquity of large and distributed data sets makes it impractical to solve the problem in a centralized fashion [19, 20]. In many cases it is not possible to store the massive amount of data at the node, which makes algorithms which rely
on multiple iterations over the data sets infeasible. This feature of the problem makes stochastic gradient descent algorithms attractive for online learning problems, since these algorithms typically require a single pass over the data. A related problem to distributed optimization is the problem of fair allocation of resources. This has been thoroughly studied in the area of microeconomics [21]. Recent interest in the resource allocation problem has arisen in the context of utility maximization in communication networks [22–24]. One of the most important characteristics of the network utility maximization problem is the fact that the objective function to be minimized has a separable form. Under this structure various primal or dual decomposition methods can be applied to make the problem amenable to a distributed solution. In the present work we study distributed optimization algorithms and provide several ways in which they can be applied to problems arising in sensor networks and large scale machine learning.

We now give the broad outline of the structure of the problems that are central to this work.

### 1.1 Problem Outline

In this work we are dealing with distributed schemes for solving the following optimization problem, where the objective function is composed of a sum of local objective functions:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} f_i(x) \\
\text{subject to} & \quad x \in X = \bigcap_{i=1}^{m} X_i, \quad X_i \subseteq \mathbb{R}^n.
\end{align*}
\tag{1.1.1}
\]

Here \( f_i(x) \) are convex not necessarily differentiable function of the decision variable \( x \), and \( X_i \) are closed and convex constraint sets. In some instances the decomposition of the constraint set \( X \) is not given \emph{a priori}. In such cases we start with the problem \( \{ \min_x \sum f_i(x) | x \in X \} \) and simplify the problem by expressing the constraint set \( X \) as an intersection of simpler constraints \( X_i \). An example is the representation of a polyhedral constraint set as an intersection of half spaces. The distributed nature of the problem arises from the fact that there are \( m \) computational agents cooperatively trying to solve the problem. The objective functions \( f_i(x) \), and the constraint sets \( X_i \) are local and private information to agent \( i \). Thus, the lack of a central hub having global information about the
objective functions and the constraints makes the problem challenging to solve.

Some of the special forms for the local objective functions $f_i(x)$ which are often used in various applications are as follows:

1. $f_i(x) = \|x - P_{X_i}[x]\|^2$. In this case if the intersection set $X$ is nonempty, then the problem (1.1.1) is equivalent to finding a point in the intersection of the convex sets $X_i$. This is also known as the convex feasibility problem.

2. $f_i(x) = \mathbb{E}_z [L_i(x, z)] + \Omega_i(x)$, where $L(x, z)$ is a convex function of variable $x$ and $\Omega_i(x)$ is a convex not necessarily differentiable function.

3. $f_i(x) = \frac{1}{n} \sum_{j=1}^{n} [L_i(x, z)] + \Omega_i(x)$. This form is obtained by approximating the expectation in the formulation above by an empirical mean.

The problem (1.1.1) is clearly a convex optimization problem and, hence, there exist various efficient algorithms for finding its solution. However, existing algorithms often use the assumption that there is a central processing node which has all the information regarding the objective function and the constraints. The lack of a central hub necessitates that the agents $i = 1, \ldots, m$, cooperatively solve the problem. We discuss later how many of the problems arising in sensor networks and distributed machine learning fit our framework. A distributed algorithm enables agents to communicate with other agents to arrive at the solution. The inter-agent communication can be represented by a graph with time varying topology with the agents as their nodes. Thus, a distributed algorithm has to be robust to the time varying topology of the communication graph. This issue becomes more relevant in the scenario when the agents are sensors communicating over a wireless network. Another major issue in a wireless network is the presence of noise in the communication links. Hence, the algorithm has to be robust to noisy communication. Another interesting source of noise comes into picture when the local objective functions are of the form $f_i(x) = \mathbb{E}_z [L_i(x, z)] + \Omega_i(x)$. In this case any gradient descent algorithm has to account for the fact that the gradient in use is typically an unbiased estimate of the true gradient. Thus, to incorporate the more general stochastic optimization problem in our framework we need to deal with gradient errors in addition to noisy communication. Other desired characteristics of an algorithm are fast convergence rate, low complexity, ease of implementation and low communication and memory
overhead. Gradient descent algorithms have the desired characteristics of being low complexity and easy to implement. However, they suffer from slow convergence rate. Another important issue worthy of consideration is the problem of asynchronous computation. It is well known that the problem of clock synchronization among different processors is a hard problem [25]. Thus it becomes imperative that we develop algorithms which do not rely on clock synchronization for their convergence behavior.

1.2 Statement of Contribution and Organization

In this section we summarize the contributions of this thesis work and provide an outline for the organization of the theses. The contributions of this thesis are threefold. First, we develop distributed synchronous and asynchronous algorithms for some distributed optimization problems arising in networks. Second, we establish various assumptions which are necessary for the algorithms and provide detailed mathematical analysis proving the convergence behavior of the proposed algorithms. Third, we discuss various applications of the proposed algorithms. The outline of the thesis is as follows:

In **Chapter 2** we fix our notation and provide some necessary mathematical background for the material which follows.

In **Chapter 3** we provide a distributed algorithm for the convex feasibility problem. The algorithm considers the presence of noise in the communication links and uses a step size sequence like the one used in stochastic approximation algorithms to guarantee almost sure convergence of the algorithm to a feasible point in the constraint set $X$. Furthermore, we prove that, for the special case when the constraint sets are subspaces and there is no noise present in the communication links, we can explicitly characterize the point to which the algorithm converges.

In **Chapter 4**, we consider the general problem formulation of (1.1.1). Once again we provide a distributed algorithm which considers the presence of both communication noise and noisy subgradients. This, makes our algorithm suitable for stochastic non-differentiable problems. Here, we need to use two step size sequences to damp communication and subgradient noise respectively. In this case we show that under some additional assumptions on the constraint sets and step size sequences we can achieve almost sure convergence to the optimal solution. More specifically our al-
gorithm characterizes the fact that almost sure convergence is achieved when the step size sequence associated with the subgradient noise goes to zero at a rate faster than the step size sequence associated with the communication noise. We then show that various problems arising in networks like consensus, constrained estimation, distributed power control in a wireless network and distributed model predictive control fit our framework of distributed optimization.

In Chapter 5, we develop asynchronous algorithms for both the problems introduced in Chapters 3 and 4. The main approach for dealing with asynchronicity we take is to make the local node step sizes a function of their local Poisson clocks instead of a global clock. Another major deviation from the models in the Chapters 3 and 4 is the introduction of a stochastic communication topology. This formulation enables us to include well known communication protocols like the broadcast and gossip into our formulation. We establish conditions on the network and the step sizes under which we can establish almost sure convergence of our algorithm for both the convex feasibility problem and the general distributed optimization problem. Furthermore, under the case when the step sizes are chosen to be constants we establish that under some stricter assumptions we can establish asymptotic error bounds for our algorithms.

In Chapter 6 we extend our distributed algorithms to the more general class of algorithms which are based on the notion of Bregman distance functions. It is well known that Bregman distance based algorithms are more general than the subgradient descent algorithms. Furthermore, we consider a problem setup in which we want to minimize the maximum loss incurred by any agent. This formulation doesn’t readily fall into the framework where the objective function is given as a sum of local objective functions. The min-max problem setup we consider is useful in the distributed resource allocation setting. We develop two algorithms for this task. Our first algorithm is based on including a non-differentiable penalty function in the Bregman distance framework. The second algorithm uses the primal-dual framework, where the agents update both the primal and dual variables in a distributed way. We prove convergence of our algorithms under the consideration of stochastic subgradient errors. Finally we show the utility of our algorithms in computing a min-max fair allocation for a power control problem in a cellular network.

In Chapter 7 we present various formulations for distributed large scale learning. We show that many distributed learning problems can be formulated in ways such that the algorithms developed in
earlier chapters can be directly applied to these problems. More specifically we present formulations for distributed regression and classification. We present parallels between the distributed regression problem with quadratic cost and the convex feasibility problem of Chapter 3, and show that the special characterization of the convergence point when the constraint sets are subspaces has direct bearing on this problem. The results of Chapter 4 and 5 are suitable when the cost functions are general non-differentiable convex functions and not necessarily quadratic. We adapt our algorithms for a distributed solution of the binary classification problem in a support vector machine setting. We provide simulation result for both the batch and online learning case.

Finally in **Chapter 8** we conclude our discussion and provide some future directions.

During the course of this thesis research the following publications were undertaken.

**Book Chapter**


**Journal Publications**


**Conference Publications**


Chapter 2

Mathematical Preliminaries

In this chapter we provide some brief background for the material which follows and introduce the notation for our future discussion.

2.1 Notation and Terminology

All vectors are viewed as column vectors. The set of real numbers is denoted by $\mathbb{R}$ and the set of positive real numbers is denoted $\mathbb{R}_+$. The $j^{th}$ component of a vector $x$ is denoted by $x_j$. For the case when there are multiple $n$ dimensional vectors indexed by an index $i$ we denote the $i^{th}$ $n$-dimensional vector as $x_i$. The transpose of a vector $x$ is denoted as $x^T$. For an $m \times m$ matrix $A \in \mathbb{R}^{m \times m}$, we use $A_{ij}$ or $[A]_{ij}$ to denote its entry in the $i^{th}$ row and $j^{th}$ column. We use $\| \cdot \|$ to denote the Euclidean norm for most of our work except for Chapter 6 where it denotes a general norm on $\mathbb{R}^n$. We write $I_r$ for the $r \times r$-dimensional identity matrix and $1_m$ for the $m$-dimensional vector with each component equal to 1. When the dimension is clear from the context we will drop the subscript and use $1$. We use $D(a_i)$ to denote a diagonal matrix with diagonal entries given by $\{a_1, \ldots, a_\ell\}$. The size of the diagonal matrix is thus given by the number of values the index $i$ takes. The null space of a matrix $A$ is denoted by $\mathcal{N}(A)$. An $m \times m$ matrix $W$ is said to be a stochastic matrix if $W_{ij} \geq 0$ for all $i, j$, and $W1_m = 1_m$. A stochastic matrix $W$ is said to be doubly stochastic if it satisfies $1_m^TW = 1_m$, where the vectors are understood to be dimensionally compatible.

Given a directed graph $G = (V, E)$, the link $(i, j) \in E$ is to be interpreted as the incoming edge from $j$ to $i$. For a bidirectional graph $G$, we have $(i, j) \in E$ if and only if $(j, i) \in E$. We will sometimes denote the edge set of a graph $G$ as $\mathcal{E}(G)$. Given any graph $G = (V, E)$ and a function $F : V \times V \rightarrow \mathbb{R}$, we use $\sum_{E} F(i, j)$ to denote the sum where the function $F(i, j)$ is evaluated for
all \((i, j) \in E\). When the graph \(G\) has bidirectional links, the sum \(\sum_{E} F(i, j)\) is assumed to be evaluated by taking every edge only once. We use the terms “agent” and “node” interchangeably. We say that agent \(j\) is a neighbor of agent \(i\) if \((i, j) \in E\), and we denote the set of all neighbors of agent \(i\) by \(N_i\). A graph \(G = (V, E)\) is \(r\)-regular if \(|N_i| = r\) for each node \(i\). The Laplacian of a graph \(G\) is a matrix \(L\) such that \(L_{ij} = -1\) if \((i, j) \in E\), \(L_{ii} = |N_i|\) and \(L_{ij} = 0\) for all \((i, j) \notin E\). For a bidirectional graph, the matrix \(L\) is symmetric, positive semidefinite, and satisfies \(L1 = 0\) and \(1^{T}L = 0\). If the graph \(G\) is connected then \(\{c1 : c \in \mathbb{R}\}\) is the unique null space of the matrix \(L\).

Given \(m\) vectors in \(\mathbb{R}^n, \{x^1, \ldots, x^m\}\), the consensus subspace is the subspace of the \(mn\)-dimensional product space, and it is defined as:

\[
C = \{z \in \mathbb{R}^{mn} : z = 1_m \otimes z, z \in \mathbb{R}^n\},
\]

which is the subspace of \(m\)-copies of the same \(n\)-dimensional vector. Thus the vectors \(x^1, \ldots, x^m\) are said to be in consensus if the concatenated vector \(x = ((x^1)^T, \ldots, (x^m)^T)^T\) lies in the consensus subspace, i.e., \(x \in C\). The set of \(mn \times mn\) symmetric positive semidefinite matrices \(A\) with \(\mathcal{N}(A) = C\) is denoted by \(S\).

Given a finite set of scalars \(\{\alpha_i\}_{i \in \mathcal{I}}\), we let \(\bar{\alpha} = \max_i \{\alpha_i\}\) and \(\underline{\alpha} = \min_i \{\alpha_i\}\). Furthermore, we let \(\Delta_\alpha = \bar{\alpha} - \underline{\alpha}\). We use both the notation \(\chi_p\) and \(1_p\) to denote the Boolean indicator function which takes the value 1 when the statement \(p\) is true, and 0 when \(p\) is false. Given a convex not necessarily differentiable function \(f(x)\), we say that a vector \(d\) is a subgradient of \(f\) at \(x\) if the following holds for every \(z\) in the domain of \(f\):

\[
d^T(z - x) \leq f(z) - f(x).
\]

The set of all subgradients of a function \(f\) at a point \(x\) is the subdifferential set, denoted by \(\partial f(x)\).

Given a closed convex set \(X \subset \mathbb{R}^n\), the projection operator \(P_X[\cdot]\) maps a vector \(x \in \mathbb{R}^n\) to the closest point to \(x\) in the set \(X\) under the Euclidean norm.
2.2 Bregman Distance

In this section we give a brief overview of the Bregman distance function. These concepts will prove useful in the development of Bregman distance based algorithms in Chapter 6. In this section we will denote a general norm on \( \mathbb{R}^n \) by \( \| \cdot \| \). The dual norm is defined and denoted as \( \| \cdot \|^* = \sup_{\|y\| \leq 1} y^T x \).

A convex function \( \omega : X \to \mathbb{R} \) is called strongly convex with convexity parameter \( \sigma \) with respect to the norm \( \| \cdot \| \), if it satisfies

\[
(x - y)^T (\nabla \omega(x) - \nabla \omega(y)) \geq \sigma \| x - y \|^2 \quad \forall x, y \in X^o,
\]

where \( X^o \) denotes the relative interior of the set \( X \). Given a strongly convex function \( \omega \) we can define the Bregman distance generating function or the prox-function \( V : X \times X^o \to \mathbb{R}_+ \) induced by \( \omega \) as

\[
V(y, x) = \omega(y) - [\omega(x) + \nabla \omega(x)^T (y - x)] . \tag{2.2.1}
\]

The Figure 2.1 shows a graphical representation of the Bregman distance function. Note that the order of the variables as we define \( V(y, x) \) is different from that of [26]. The following property of the Bregman distance are easy to establish and can be found in [26,27]. Let the Bregman distance function \( V(\cdot, \cdot) \) be generated by a strongly convex function \( \omega \) with convexity parameter \( \sigma \). Then,
$V(x, y)$ is nonnegative and for every $x \in X^\circ$ the function $V(\cdot, x)$ is strongly convex with parameter $\sigma$.

The distance function $V(y, x)$ can be used to define a nonlinear projection operator, also known as the prox-operator [26] as follows:

$$P_x(y) = \text{argmin}_{z \in X} \{ y^T (z - x) + V(z, x) \}.$$  \hspace{1cm} (2.2.2)

Let us consider the following convex optimization problem: $\min_{x \in X} f(x)$, where $f(x)$ is a convex function and $X$ is a convex set with nonempty interior. Let $\omega(x)$ be a strongly convex function defined on $X$ and $V(y, x)$ be the Bregman distance generated by $\omega(x)$. Then, the Bregman distance based algorithm for this problem is given as

$$x_{k+1} = P_x(\gamma_k d_k),$$ \hspace{1cm} (2.2.3)

where $P_x(y)$ is the prox-operator, $d_k$ is a subgradient of the convex function $f(x)$ at $x_k$ and $\gamma_k$ is the step size. The convergence of this algorithm is studied in [26], when $d_k$ is replaced by an unbiased estimate of the gradient. In the special case when the function $\omega(x) = \frac{1}{2} \|x\|_2^2$, the Bregman function becomes $V(y, x) = \frac{1}{2} \|x - y\|^2$, and in this case the algorithm (2.2.3) reduces to the well known subgradient descent algorithm [28]

$$x_{k+1} = P_X [x_k - \gamma_k d_k],$$ \hspace{1cm} (2.2.4)

where $P_X$ is the orthogonal projection operator on the set $X$. Thus, Bregman distance based algorithm can be thought of as a generalization of the orthogonal projection algorithm. An interesting algorithm case is when the constraint set $X$ is the probability simplex. In this case, we can equip the space with $\|\cdot\|_1$ norm, with $\|\cdot\|_\infty = \|\cdot\|_\infty$, and choose $\omega$ as the entropy function $w(x) = \sum_{i=1}^{n} x_i \log x_i$. This results in the Bregman function $V(x, z) = \sum_{i=1}^{n} z_i \log \frac{z_i}{x_i}$, and the prox-mapping takes the following form:

$$[P_x(y)]_i = \frac{x_i e^{-y_i}}{\sum_{r=1}^{n} x_r e^{-y_r}}, \quad i = 1, \ldots, n.$$
Thus, the algorithm (2.2.3) can be written as

\[ x_{k+1}^i = \frac{[x_k]^i e^{-\gamma_k |d_k|^2}}{\sum_{r=1}^{n} [x_k]^r e^{-\gamma_k |d_k|^2}}, \quad i = 1, \ldots, n. \]

### 2.3 Basic Results

Now, we state a couple of useful standard results. The following lemma from [11] gives some relations regarding the projection operator on a convex set, where the norm \( \| \cdot \| \) is the standard Euclidean norm.

**Lemma 2.3.1.** [11] Let \( X \) be a nonempty closed convex set in \( \mathbb{R}^n \). Then, we have for all \( x, y \in X \),

1. \( (P_X x - x)^T (x - y) \leq -\|P_X x - x\|^2 \).
2. \( \|P_X x - y\|^2 \leq \|x - y\|^2 - \|P_X x - x\|^2 \).
3. \( \|P_X x - P_X y\| \leq \|x - y\| \).

The following known theorem, which is a generalization of the supermartingale convergence theorem will be instrumental in proving our results. The theorem is also known as the Robbins-Siegmund convergence result.

**Theorem 2.3.2.** ([29], page 50) Let \( \{X_t\}, \{Y_t\}, \{Z_t\} \) and \( \{g_t\} \) be sequences of random variables and let \( F_t, t = 0, 1, 2, \ldots, \) be a filtration such that \( F_t \subset F_{t+1} \) for \( t \geq 0 \). Suppose that:

1. The random variables \( Y_t, X_t, Z_t \) and \( g_t \) are nonnegative, and are adapted to the filtration \( F_t \).
2. For each \( t \), we have almost surely

\[
\mathbb{E}[Y_{t+1} | F_t] \leq (1 + g_t)Y_t - X_t + Z_t. \quad (2.3.1)
\]

3. There holds \( \sum_{t=0}^{\infty} Z_t < \infty \) and \( \sum_{t=0}^{\infty} g_t < \infty \) almost surely.

Then, almost surely, we have \( \sum_{t=0}^{\infty} X_t < \infty \) and the sequence \( Y_t \) converges to a nonnegative random variable \( Y \).
Chapter 3

Consensus Based Algorithm for Convex Feasibility Problems

In this chapter we are concerned with the following problem. Given closed and convex sets $X_i \subseteq \mathbb{R}^n$, $i = 1, \ldots, m$, we need to find a point $x \in \cap_{i=1}^m X_i$. This is a classic problem and arises in various areas of science and engineering including distributed machine learning [20], low-order control design [30] and image restoration [31] among others. A simple example of this problem arises in checking the feasibility of a given set of convex constraints in an optimization problem. In this case the constraint sets have the representation

$$X_i = \{x \in \mathbb{R}^n | g_i(x) \leq 0\}$$

where $g_i(x)$ is a convex function. This is a well studied problem and several algorithms have been proposed for this task. One of the most widely used algorithms is the method of successive orthogonal projection (SOP) of Gubin, Polyak and Raik [32], also known as the method of projections onto convex sets (POCS) in the literature on image recovery [33]. There are several other variants proposed for this algorithm. A comprehensive treatment can be found in the book [27]. A major drawback of the SOP algorithm is the fact that it is sequential in nature. Recently the authors in [11] proposed a distributed algorithm which is more amenable to be employed in a sensor network.

The main contributions in this chapter are two fold. First we propose an algorithm based on the algorithm proposed in [11] which can handle the presence of noise in the communication channel. We provide a complete analysis showing almost sure convergence of the iterates. Second we show that when the constraint sets are hyperplanes, we can explicitly characterize the convergent points of the algorithm in [11]. This result proves to be crucial in our later work on distributed machine learning in Chapter 7.
The rest of the chapter is organized as follows. In Section 3.1, we provide a brief description of the SOP method along with its parallel variant. Then in Section 3.2 we formally state our problem of interest and the various assumptions. In Section 3.3 we state some preliminary results and derive our main convergence result in Section 3.4. Finally, we consider the special case of subspace constraints in Section 3.5 and provide the conclusion in Section 3.6. The results in this Chapter excluding the result in Section 3.5 is based on the results provided in [34].

3.1 Successive Orthogonal Projection

Given the closed and convex constraint sets $X_i \in \mathbb{R}^n$ for $i = \{1, \ldots, m\}$ the successive orthogonal projection method proceeds by projecting iteratively on the constraint sets. A control sequence is the order in which the constraint sets are chosen. The control sequence can be described as a mapping $\nu : \mathbb{N} \rightarrow \{1, \ldots, m\}$. Then the iterations of SOP proceed as follows

$$x_{k+1} = x_k + \gamma (P_{X_{\nu(k)}}[x_k] - x_k),$$

where $\gamma$ is a relaxation parameter. For $\gamma < 1$, the method is said to be under-relaxed and for $\gamma > 1$, it is said to be over-relaxed [35]. Of special interest is the case when $\gamma = 1$. Convergence of the above algorithm has been studied for the general case when the sets $X_i$ are subsets of a Hilbert space. A detailed review is provided in [35]. A drawback of the SOP algorithm is the fact that it
is sequential in nature. Some parallel algorithms have been suggested as an extension to the SOP algorithm. An algorithm suggested by Cimmino simultaneously projects on all the convex sets. Algorithmically this proceeds as

\[ x_{k+1} = x_k + \gamma \left( \sum_{i=1}^{m} w_i P_{X_i}[x_k] - x_k \right), \]

(3.1.2)

where \( w_i > 0 \) are weight vectors such that \( \sum_{i=1}^{m} w_i = 1 \). However, a drawback of the Cimmino algorithm is the fact that it requires a centralized hub with which all the processors need to communicate.

Next we discuss a multiprojection algorithm [27] which has an interesting interpretation in the product space and is relevant to our discussion later.

### 3.1.1 The Multiprojection Algorithm

Let us define the product space \( V = \mathbb{R}^{mn} = \mathbb{R}^n \times \cdots \times \mathbb{R}^n \). Given vectors \( x = (x_1, \ldots, x_m) \) and \( y = (y_1, \ldots, y_m) \) in \( V \), the inner product induced on \( V \) is given as \( \langle x, y \rangle = \sum_{i=1}^{m} \langle x_i, y_i \rangle \). Next, let us define in \( V \) the product set \( X = X_1 \times \ldots \times X_m = \prod_{i=1}^{m} X_i \). Also, let us define the consensus subspace as \( \Delta = \{ x \in V | x = (x, \ldots, x), x \in \mathbb{R}^n \} \). Then the sets \( \{X_i\} \) have nonempty intersection if and only if the set \( X \cap \Delta \) is non-empty. The SOP algorithm when applied to the product space can be written as

\[ x_{k+1} = P_X [P_{\Delta}[x_k]]. \]

On, expanding component-wise the above expression we get

\[ x_{k+1}^i = P_{X_i} \left[ \frac{1}{m} \sum_{i=1}^{m} x_k^i \right], \]

(3.1.3)

where we have used the fact that in a Euclidean space with \( l_2 \) norm, given a vector \( x \in V \), the projection operation is given by \( P_{\Delta}[x] = \frac{1}{m} \left( \sum_{i=1}^{m} x_i, \ldots, \sum_{i=1}^{m} x_i \right) \). Clearly this algorithm parallels the \( P_{X_i}[\cdot] \) operation. However, the operation \( P_{\Delta}[\cdot] \), either requires a central hub or a fully connected network. Thus, even this algorithm doesn’t achieve true parallelization. In [11], the authors provided an algorithm which achieves parallelization by relaxing the global averaging.
\[ \frac{1}{m} \sum_{i=1}^{m} x^i_k \] to a local averaging of the form \( \sum_{j \in N_i(k)} w_j(k)x^j_k \), where \( N_i(k) \) is a local neighborhood of the node \( i \).

In the next section we formally introduce the problem setup under consideration and our proposed algorithm.

### 3.2 Problem Formulation

We consider a setup where we are given a set of \( m \) agents, which can be viewed as the node set \( V = \{1, \ldots, m\} \). We use terms node and agent interchangeably. At each time \( k \), the communication pattern among the agents is represented by a time varying directed graph \( G(k) \). At each instant, each node receives information from a subset of nodes and, also, broadcasts its information to a subset of nodes. The subset of nodes from which a node receives information at any instant are termed as the neighboring nodes of the node at that instant. Such information exchanges are characterized by the edge set \( E(k) \) of the graph. For the scope of this paper we deal with synchronous algorithms. This implies that agents’ local clocks are synchronized and time proceeds in discrete steps \( k = 0, 1, \ldots \). Let us associate with each agent \( i \), a constraint set \( X_i \). Each local constraint set is private and local information to node \( i \). The objective of the network is to cooperatively solve the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} \|x - P_{X_i}[x]\|^2 \\
\text{subject to} & \quad x \in \mathbb{R}^n.
\end{align*}
\] (3.2.1)

When the intersection \( X = \bigcap_{i=1}^{m} X_i \) is nonempty then, a solution to the above problem is given by any vector \( x \) which lies in the constraint set \( X \). Clearly, in this case the objective function value is zero, which is also the optimal value. Let us denote agent \( i \)'s estimate of the solution at time \( k \) by the variable \( x^i_k \). Since, agent \( i \) has information only regarding his constraint set, it is desirable to require that the local estimates satisfy the constraint \( x^i_k \in X_i \), for all \( i \) and all times \( k \). An algorithm is said to solve the problem if it generates agent estimates \( x^i_k \in X_i \) that converge to a common value \( x^* \), and \( x^* \) satisfies the constraint \( x^* \in X \).

There is an alternative view to look at the problem 3.2.1. This problem can be equivalently cast as
a problem of constrained consensus. In this formulation the network tries to achieve consensus on the local variables $x^i_k$, while satisfying the local feasibility constraint $x^i_k \in X_i$. Clearly these two viewpoint are equivalent. A distributed algorithm for this problem was proposed in [11]. In the algorithm agent $i$ local variables $x^i_k$ evolves as follows:

$$x^i_{k+1} = P_{X_i} \left[ x^i_k + \alpha_{k+1} \sum_{j=1}^m a_{ij}(k+1)x^j_k \right],$$

where $a_{ij}(k+1)$ denotes the weight assigned by node $i$ to the estimate coming from node $j$. The problem of achieving consensus is a widely researched problem in it’s own right [1–3]. The algorithms for reaching consensus have proven useful in a wide variety of contexts from formation control [3], distributed parameter estimation [12], [13], load balancing [36], to synchronization of Kuramoto oscillators [15]. Consensus over noisy links in the lack of constraint sets has been studied in [37], [38] and [13] among others. A crucial assumption needed in the analysis in [11] was the requirement that if agent $i$ receives data from agent $j$ then $a_{ij}(k) \geq \eta > 0$. We are interested in the case when the communication links are noisy and hence node $i$ has access to a noise corrupted value of its neighbor’s local estimate. In this case it is detrimental to impose the requirement that $a_{ij}(k) \geq \eta$ since we need to asymptotically damp the impact of noise. To this effect we propose the following distributed algorithm for the problem:

$$x^i_{k+1} = P_{X_i} \left[ x^i_k + \alpha_{k+1} \sum_{j \in N_i(k)} r_{ij,k+1}[x^i_k - (x^j_k + \xi_{ij,k+1})] \right].$$

Here $\alpha_{k+1} > 0$ is a step size, $r_{ij,k+1}$ is a weighting parameter, $\xi_{ij,k+1}$ is a random variable denoting the additive noise in communication, and $N_i(k)$ denotes the set of agents communicating with agent $i$ at instance $k$. Let us define $r_{ii}(k+1) = -\sum_{j \in N_i(k)} r_{ij,k+1}$, and $\xi_{i,k+1} = \sum_{j \in N_i(k)} r_{ij,k+1}\xi_{ij,k+1}$. Then, the algorithm can be rewritten as

$$x^i_{k+1} = P_{X_i} \left[ x^i_k + \alpha_{k+1} \sum_{j=1}^m r_{ij,k+1} x^j_k + \alpha_{k+1} \xi_{i,k+1} \right].$$

(3.2.2)

The matrix $R_k$, where $R_{ij,k} = r_{ij,k}$, is thus a weighted graph Laplacian and it satisfies $\sum_{j=1}^m r_{ij,k} = 0$ for all $i$. We impose the following assumptions on the graph $G(k)$ and the weight sequence.
3.2.1 Model Assumptions

In this section we state our various assumptions which are useful in deriving our convergence result.

Assumption 1.

1. (Bi-directional communication) We assume that the communication is bi-directional; i.e if at any instant $k$, $r_{ij,k} > 0$ then $r_{ji,k} > 0$.

2. (Symmetric weights) The neighboring agents use symmetric weights, i.e., $r_{ij,k} = r_{ji,k}$.

3. (Connectedness) We assume that the graph $G(k)$ is connected at every instance, though it is free to be time varying.

4. We assume that if $r_{ij,k} \neq 0$ at any instant, then it satisfies $\eta \leq r_{ij,k} \leq \eta'$, where $\eta$ and $\eta'$ are positive constants.

Let us denote the filtration $\{F_k\}$ as the history up to time $k$:

$$F_k = \{x^i_s, \xi_{i,s}, i = 1, \ldots, m, 0 \leq s \leq k\}. \quad (3.2.3)$$

We impose the following assumptions on the spatio-temporal noise process.

Assumption 2.

1. The process $\{\xi_{ij}(k)\}$ is a martingale difference sequence, i.e., $\mathbb{E}[\xi_{ij,k+1}|F_k] = 0$ for all $i, j$ and $k \in \mathbb{N}$.

2. At any fixed instance $k$, the noise on link $e_1 = (i_1, j_1)$ is independent of the noise on link $e_2 = (i_2, j_2)$ for $e_1 \neq e_2$.

3. The noise process is uniformly bounded in the mean square sense, i.e., there is a deterministic scalar $\mu_i > 0$ such that $\mathbb{E}[\|\xi_{i,k+1}\|^2|F_k] \leq \mu_i^2$ for all $k \in \mathbb{N}$.

From Assumption 2.3, it follows that $\mathbb{E}[\|\xi_{i,k+1}\| | F_k] \leq \sqrt{\mathbb{E}[\|\xi_{i,k+1}\|^2 | F_k]} \leq \mu_i$ for all $k \in \mathbb{N}$ and all $i$. Let us also define $\mu^2 = \sum_{j=1}^{m} \mu_j^2$. Furthermore we impose the following assumption on the step size sequence $\{\alpha_k\}$.

Assumption 3.

1. The step size in (3.2.2) is such that $\alpha_k > 0$, $\sum_{k=1}^{\infty} \alpha_k = \infty$, and $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$.  

18
3.3 Preliminary Results

In this section we provide various results which will be useful in proving our main results.

Now, we establish certain relations for algorithm (3.2.2). Let

\[ v_{k+1}^i = x_k^i + \alpha_{k+1} \sum_{j=1}^{m} r_{ij,k+1} x_k^j + \alpha_{k+1} \xi_{i,k+1}. \]  (3.3.1)

Then, we have

\[ x_{k+1}^i = v_{k+1}^i + e_{k+1}^i, \]

where the error term is given by

\[ e_{k+1}^i = P_{X} [v_{k+1}^i] - v_{k+1}^i = x_{k+1}^i - v_{k+1}^i. \]

**Lemma 3.3.1.** When the set \( X = \cap_{i=1}^{m} X_i \) is nonempty, we surely have for all \( i \in V, k \geq 0 \), and for all \( x \in X \),

\[ \| x_{k+1}^i - x \|^2 \leq \| v_{k+1}^i - x \|^2 - \| e_{k+1}^i \|^2. \]

**Proof.** By the definition of \( x_{k+1}^i \) we have

\[ \| x_{k+1}^i - x \|^2 = \| P_{X} [v_{k+1}^i] - x \|^2. \]

Now, applying Lemma 2.3.1, we get along every sample path

\[ \| P_{X} [v_{k+1}^i] - x \|^2 \leq \| v_{k+1}^i - x \|^2 - \| P_{X} [v_{k+1}^i] - v_{k+1}^i \|^2 \]

\[ = \| v_{k+1}^i - x \|^2 - \| e_{k+1}^i \|^2. \]  (3.3.2)

Let us define \( x_k = (x_1^T, \ldots, x_m^T)^T \) as the joint state vector and, similarly, \( e_{k+1} \) and \( \xi_{k+1} \) be the respective stacked up versions. Define the \( mn \times mn \) matrix \( R_{k+1} = R_{k+1} \otimes I_n \). The joint state space representation of algorithm (3.2.2) can be given as:

\[ x_{k+1} = [I_{mn} + \alpha_{k+1} R_{k+1}] x_k + \alpha_{k+1} \xi_{k+1} + e_{k+1}. \]

Before we state the next lemma for notational convenience let us define \( H(k+1) = 1 + \alpha_{k+1}^2 \sigma_m^2 (R_{k+1}) \).

Here \( \sigma_m^2 (R_{k+1}) \) denotes the square of the maximum singular value of the matrix \( R_{k+1} \). For any
vector \( z \in X \) let us denote \( z = z \otimes 1_m \), where \( 1_m \) is the \( m \)-dimensional vector of ones. Also, let us denote the conditional expectation operator \( \mathbb{E}_k[\cdot] = \mathbb{E}[\cdot | F_k] \). Since we assume that the communication graph is connected at every instant (cf. Assumption 1.3), there exists a spanning tree \( S(k) \) such that the edge \((i, j)\) belongs to the tree if and only if \( r_{ij,k+1} > \eta \).

**Lemma 3.3.2.** Let Assumptions 1 and 2 hold. Also, assume that the set \( X = \cap_{j=1}^m X_j \) is nonempty. Then, the following relation holds for any \( z \in X \):

\[
\mathbb{E}_k[\|x_{k+1} - z\|^2] \leq H(k+1)\|x_k - z\|^2 + \alpha_{k+1}^2 \mu^2 - 2\eta \alpha_{k+1} \sum_{S(k)} \|x^i_k - x^j_k\|^2,
\]

where \( \sum_{S(k)} \|x^i_k - x^j_k\|^2 \) denotes summing the terms \( \|x^i_k - x^j_k\|^2 \) over all edges arising in the spanning tree \( S(k) \).

**Proof.** Note that from Lemma 3.3.1 we have

\[
\mathbb{E}_k[\|x_{k+1} - z\|^2] \leq \mathbb{E}_k[\|v_k - z\|^2] - \mathbb{E}_k[\|e_{k+1}\|^2],
\]

where \( v_k = [I_m + \alpha_{k+1}R_{k+1}]x_k + \alpha_{k+1}\xi_{k+1} \). Hence,

\[
\mathbb{E}_k[\|v_k - z\|^2] = \|[I_m + \alpha_{k+1}R_{k+1}]x_k - z\|^2 + \alpha_{k+1}^2 \mathbb{E}_k[\|\xi_{k+1}\|^2] + 2\alpha_{k+1}(\|[I_m + \alpha_{k+1}R_{k+1}]x_k - z\|) \mathbb{E}_k[\xi_{k+1}].
\]

By our Assumption 2.1 \( \mathbb{E}_k[\xi_{k+1}] = 0 \), so that

\[
\mathbb{E}_k[\|v_k - z\|^2] = \|[I_m + \alpha_{k+1}R_{k+1}]x_k - z\|^2 + \alpha_{k+1}^2 \mathbb{E}_k[\|\xi_{k+1}\|^2].
\]

Since \( R_{k+1} \) is symmetric and has zero-row sums, we have

\[
\|[I_m + \alpha_{k+1}R_{k+1}]x_k - z\|^2 = [x_k - z]^T [I_m + \alpha_{k+1}R_{k+1}]^T [I_m + \alpha_{k+1}R_{k+1}] [x_k - z]
\]

\[
= [x_k - z]^T [I_m + \alpha_{k+1}^2 R^T (k+1)R_{k+1}] [x_k - z] + 2\alpha_{k+1} \|x_k - z\|^2 R_{k+1} [x_k - z].
\]
We also have

\[ [x_k - z]^T R_{k+1}^T R_{k+1} [x_k - z] \leq \sigma_m^2 (R_{k+1}) \|x_k - z\|^2, \]

where \( \sigma_m(R_{k+1}) \) is the maximum singular value of the matrix \( R_{k+1} \). Since \( R_{k+1} = R_{k+1} \otimes I_n \), we have \( \sigma_m(R_{k+1}) = \sigma_m(R_{k+1}) \). Hence,

\[ [x_k - z]^T [I_{mn} + \alpha_{k+1}^2 R_{k+1}^T R_{k+1}] [x_k - z] \leq H(k+1) \|x_k - z\|^2, \]

with \( H(k+1) = 1 + \alpha_{k+1}^2 \sigma_m^2 (R_{k+1}) \).

Now, consider the term \( [x_k - z]^T R_{k+1} [x_k - z] \) in (3.3.4). Since \( R_{k+1} \) has row sum zero and \( R_{k+1} \) is symmetric, we have \( R_{k+1} z = 0 \) and \( z^T R_{k+1} = 0 \). Thus,

\[ [x_k - z]^T R_{k+1} [x_k - z] = x_k^T R_{k+1} x_k. \]

By definition \( R_{k+1} = R_{k+1} \otimes I_m \), where \( r_{ij,k+1} = r_{ij,k+1} \) and \( R_{ii,k+1} = -\sum_{j=1, j \neq i}^m r_{ij,k+1} \). Thus, we see that

\[ x_k^T R_{k+1} x_k = -\sum_{i=1}^m (x_k^T)^T \sum_{j=1, j \neq i}^m r_{ij,k+1} (x_k^i - x_k^j). \]

Since, the matrix \( R_{k+1} \) is symmetric we have

\[-\sum_{i=1}^m x_k^T (k) \sum_{j=1, j \neq i}^m r_{ij,k+1} (x_k^i - x_k^j) = -\sum_{i<j}^m r_{ij,k+1} \left\| x_k^i - x_k^j \right\|^2 \leq -\eta \sum_{S(k)} \left\| x_k^i - x_k^j \right\|^2, \]

where \( \sum_{S(k)} \left\| x_k^i - x_k^j \right\|^2 \) denotes summing the terms \( \left\| x_k^i - x_k^j \right\|^2 \) over all edges arising in the spanning tree. Using the preceding relation and the bound \( \mathbb{E}_k[\|\xi_{k+1}\|^2] \leq \mu^2 \) we arrive at

\[ \mathbb{E}_k[\|v_k - z\|^2] \leq H(k+1) \|x_k - z\|^2 + \alpha_{k+1}^2 \mu^2 - 2\alpha_{k+1} \eta \sum_{S(k)} \left\| x_k^i - x_k^j \right\|^2 \]

By substituting back in Eq. (3.3.3) we get

\[ \mathbb{E}_k[\|x_{k+1} - z\|^2] \leq H(k+1) \|x_k - z\|^2 + \alpha_{k+1}^2 \mu^2 - 2\alpha_{k+1} \eta \sum_{S(k)} \left\| x_k^i - x_k^j \right\|^2 - \mathbb{E}_k[\|e(k+1)\|^2]. \]
Neglecting the error term we have the desired result. \[ \square \]

### 3.4 Convergence Result

In this section we state and prove our main convergence result for the algorithm (3.2.2).

**Theorem 3.4.1.** Assume that \( X = \bigcap_{j=1}^{m} X_j \) is nonempty, and let Assumptions 1 and 2 hold. Let the step size sequence \( \{\alpha_k\} \) in algorithm (3.2.2) satisfy Assumption 3. Then, there exists a random variable \( x^* \) taking values in the set \( X \) such that almost surely \( \lim_{k \to \infty} \|x^i_k - x^*\| = 0 \) for all agents \( i \).

**Proof.** First, we consider the term \( H(k+1) = 1 + \alpha_{k+1}^2 \sigma_m^2(R_{k+1}) \) in Lemma 3.3.2. The entries of the matrix \( R_{k+1} \) are uniformly bounded, implying \( \sigma_m^2(R_{k+1}) \leq C \) for some scalar \( C \) and all \( k \). By our assumption on the step size, we have \( \sum_{k=1}^{\infty} \alpha_k^2 < \infty \). Since all the terms appearing in Lemma 3.3.2 are nonnegative, we can apply the result of Robbins-Siegmund (Theorem 2.3.2) to deduce that, with probability one, \( \|x_k - z\|^2 \) converges for any \( z \in X \) and

\[
\sum_{k=1}^{\infty} \alpha_{k+1} \sum_{S(k)} \|x^i_k - x^j_k\|^2 < \infty. \tag{3.4.1}
\]

By \( \sum_{k=1}^{\infty} \alpha_k = \infty \), relation (3.4.1) implies that there is a subsequence such that

\[
\lim_{k \to \infty} \sum_{S(n_k)} \|x^{i}_{n_k} - x^{j}_{n_k}\|^2 = 0.
\]

Now, since the number of spanning trees on a finite graph is finite, there exists a spanning tree \( S \) which appears infinitely often in the sequence \( \{S(n_k)\} \). Let us pick a further subsequence such that \( S(n_k^1) = S \), then we have along this subsequence \( \lim_{k \to \infty} \sum_{S} \|x^{i}_{n_k^1} - x^{j}_{n_k^1}\|^2 = 0 \) for all \( i \) and \( j \). The spanning tree \( S \) is in the connected graph, so the preceding relation yields \( \lim_{k \to \infty} \|x^{i}_{n_k^1} - x^{j}_{n_k^1}\|^2 = 0 \) for all \( i, j \). Now since \( \|x_k - z\|^2 \) converges almost surely for any \( z \in X \), the subsequence \( \{x_{n_k^1}\} \) is bounded almost surely. Again, we can extract a convergent subsequence \( x_{n_k^2} \) such that \( \lim_{k \to \infty} \|x^{i}_{n_k^2} - x^*_i\| = 0 \) almost surely for some random vector \( x^*_i \) for all \( i \). Since

\[
\lim_{k \to \infty} \|x^{i}_{n_k^2} - x^{j}_{n_k^2}\| = 0
\]
almost surely for all \(i, j\), it follows that \(x_i^* = x_j^* = x^*\) almost surely for all \(i, j\). The sets \(X_i\) are closed, so that \(x^* \in X_i\) almost surely for all \(i\), which in turn implies that \(x^* \in X\) almost surely. Therefore, \(\lim_{k \to \infty} \|x_n^2 - x^*\|^2 = 0\) almost surely. But, we know that \(\lim_{k \to \infty} \|x_k - z\|^2\) converges almost surely for all \(z \in X\). Hence, by looking at the sample paths, we can conclude that the limit of any subsequence is also the sequential limit, implying that \(\lim_{k \to \infty} \|x_k - x^*\| = 0\) almost surely.

\[\Box\]

### 3.5 Subspace Constraints

In this section we provide a unique result which holds when the constraint sets \(X_i\) are subspaces.

It was shown by Von Neumann [39] that the infinite product of the operator \(P_{X_2}P_{X_1}\), when \(X_1\) and \(X_2\) are subspaces in a Hilbert space, converges to the operator \(P_{X_1 \cap X_2}\), i.e \(\lim_{k \to \infty} [P_{X_2}P_{X_1}]^k = P_{X_1 \cap X_2}\). This result was extended by Halperin [40] to a finite number of subspaces. We essentially show that a similar result holds for the algorithm 3.2.2. We show that when the iterations \(x_k^i\) are initialized as the projection of a common point \(x_0\) on the constraint subspaces \(X_i\), then the
algorithm has the property that asymptotically the iterations converge to the projection of \( x_0 \) on the intersection of the subspaces \( X \).

In this section we restrict ourself to the noiseless case. In the noiseless case our algorithm is same as the one provided in [11]

\[
x^{i}_{k+1} = P_{X_i} \left[ \sum_{j=1}^{m} w_{ij,k+1} x^j_k \right]. \tag{3.5.1}
\]

In this case, we assume that the underlying space \( X \subseteq \mathcal{H} \), is a finite dimensional Hilbert space. Hence, the projection operation \( P_{X_i} \) is now understood in terms of minimizing the Hilbert space norm \( \| \cdot \|_{\mathcal{H}} \) on \( \mathcal{H} \). The constraint sets \( X_i \) have the following kernel representation in terms of a linear operator \( A_i : \mathcal{H} \rightarrow \mathcal{R}(A_i) \):

\[
X_i = \{ x \in \mathcal{H} : A_ix = 0 \}.
\]

Equivalently \( X_i \) could be represented as

\[
X_i = \{ x \in \mathcal{H} : \langle z_{ij}, x \rangle_{\mathcal{H}} = 0, \ z_{ij} \in \mathcal{H}, \ j = 1, \ldots, k_i \}.
\]

In this case the operator \( A_i \) can be constructed as

\[
A_ix = \begin{bmatrix}
\langle z_{i1}, x \rangle_{\mathcal{H}} \\
\vdots \\
\langle z_{ik_i}, x \rangle_{\mathcal{H}}
\end{bmatrix}.
\]

Let us denote the kernel of the operator \( A_i \) as \( \mathcal{N}(A_i) \). Let us define the joint operator

\[
A = \begin{bmatrix}
A_1 \\
\vdots \\
A_m
\end{bmatrix}.
\]

Then it is clear that \( X = \mathcal{N}(A) = \cap_{j=1}^{m} X_j = \{ x \in \mathcal{H} : Ax = 0 \} \). Let us denote the projection operator on the kernel of \( A_i \) as \( H_i \). Let us denote \( A_i^* \) as the adjoint operator of \( A_i \), then, it can be
checked that

\[ H_i = I - A_i^\dagger A_i, \]

where \( A_i^\dagger = A_i^* (A_i^* A_i)^{-1} \) is the pseudoinverse operator. Then our algorithm can be written as

\[ x_{i,k+1}^j = H_i \left[ \sum_{j=1}^{m} w_{ij,k+1} x_k^j \right]. \]

We now have the following result.

**Theorem 3.5.1.** Let us assume that the constraint sets \( X_i \) are subspaces. Given any point \( x_0 \in \mathcal{H} \), let us initialize \( x_1^i = P_{X_i}[x_0] \). Then the iterations of the algorithm 3.5.1 converge to a common point \( x^* \in X \), such that \( x^* = P_X[x_0] \), where \( X = \bigcap_{i=1}^m X_i \).

**Proof.** We know from our analysis earlier that each of the iterates \( x_k^i \) converges to the same point \( x^* \) in \( X \). We want to show that the point \( x^* \) is the projection of \( x_0 \) on \( X \). In other words we want to show that \( x^* - x_0 \perp \mathcal{N}(A) = X \). Since, \( \mathcal{R}(A^*) \perp \mathcal{N}(A) \), this is equivalent to showing that \( x^* - x_0 \in \mathcal{R}(A^*) \). We will prove this by induction. First, let us notice that for all \( i \)

\[ x_1^i = H_i x_0 = [I - A_i^\dagger A_i] x_0, \]

we have \( x_1^i - x_0 = -A_i^\dagger A_i x_0 \). Thus, \( x_1^i - x_0 \in \mathcal{R}(A_i^*) \subseteq \mathcal{R}(A^*) \). Thus we can rewrite \( x_1^i = x_0 + v_1^i \), where \( v_1^i \in \mathcal{R}(A^*) \). Now, let us assume that for all \( i \) we can write \( x_k^i = x_0 + v_k^i \), where \( v_k^i \in \mathcal{R}(A^*) \).

Now, since \( x_{k+1}^i = H_i \left[ \sum_{j=1}^{m} w_{ij,k+1} x_{k+1}^j \right] \), it follows that

\[ x_{k+1}^i - \sum_{j=1}^{m} w_{ij,k+1} x_{k+1}^j \in \mathcal{R}(A_i^*) \subseteq \mathcal{R}(A^*). \]

Hence, \( x_{k+1}^i - \sum_{j=1}^{m} w_{ij,k+1} x_{k+1}^j = s_i(k+1) \), for an element \( s_i(k+1) \in \mathcal{R}(A^*) \). Now, plugging back from our induction step we get

\[ x_{k+1}^i - \sum_{j=1}^{m} w_{ij,k+1} x_{k+1}^j = s_i(k+1). \]
Using the fact that \( \sum_{j=1}^{m} w_{ij,k+1} = 1 \), we get that

\[
x_{k+1}^i - x_0 = \sum_{j=1}^{m} w_{ij,k+1} v_{k}^j + s_i(k + 1).
\]

Since \( v_{k}^j \in \mathcal{R}(A^*) \), and \( s_i(k + 1) \in \mathcal{R}(A^*) \), we deduce that \( x_{k+1}^i - x_0 \in \mathcal{R}(A^*) \). Now, denoting \( v_{k+1}^i = \sum_{j=1}^{m} w_{ij,k+1} v_{j}(k) + s_i(k + 1) \), we can rewrite \( x_{k+1}^i = x_0 + v_{k+1}^i \). From the principle of mathematical induction we can deduce that \( x_{k}^i - x_0 \in \mathcal{R}(A^*) \). Thus, since \( \mathcal{R}(A^*) \) is a closed subspace we have \( x^* - x_0 \in \mathcal{R}(A^*) \).

\[\Box\]

### 3.6 Conclusion

In this Chapter we provided a distributed algorithm for the convex feasibility problem. The algorithm can be applied over a time varying network with stochastic noise in the communication links. The algorithm resembles a stochastic approximation scheme by utilizing a diminishing square summable step size sequence to mitigate the effect of communication noise. For the noiseless case we characterized the convergent point of the algorithm when the constraint sets are subspaces. In Chapter 5 we extend the proposed algorithm to the asynchronous setting. We also establish asymptotic error bounds for the case when the step size \( \alpha_k \) is chosen to be a constant.
Chapter 4

Synchronous Algorithm for
Distributed Constrained Stochastic
Optimization

In this chapter we consider the general problem of solving the distributed optimization problem when the objective function is a sum of \( m \) local convex objective functions corresponding to \( m \) agents. The objective of the agents is to cooperatively solve the following constrained optimization problem:

\[
\text{minimize} \quad \sum_{i=1}^{m} f_i(x) \\
\text{subject to} \quad x \in X = \bigcap_{i=1}^{m} X_i, \quad (4.0.1)
\]

where each \( f_i : \mathbb{R}^n \to \mathbb{R} \) is a convex function, representing the local objective function of agent \( i \), and each set \( X_i \subseteq \mathbb{R}^n \) is a compact and convex set, representing the local constraint set of agent \( i \). Since the objective function is continuous and the set \( X \) is compact, we know by Weirstrass theorem that the optimal set is nonempty. Let us denote the optimal set by \( X^* \). We assume that the local constraint set \( X_i \) and the objective function \( f_i \) are known to agent \( i \) only. In this formulation the local objective functions are allowed to be of the form

\[
f_i(x) = \mathbb{E}_z[L_i(x, z)] + \Omega(x),
\]

where the expectation is over the unknown distribution of random variable \( z \) and \( \Omega(x) \) is a not necessarily differentiable function of \( x \). Such formulations naturally arise in problems related to machine learning [41, 42], which is dealt in more detail in Chapter 7. \( \Omega(x) \) is a regularization term included to improve the generalization ability [43]. Recently a lot of interest in signal processing has been generated towards the use of the \( l_1 \)-norm as the regularization term. In many cases it has been shown that such a regularization yields sparse solutions. Since the \( l_1 \)-penalty is nondifferentiable,
our algorithm which doesn’t require the objective function to be differentiable is suitable for this problem. It is well known that the stochastic optimization problems of the form above can be dealt with by using first-order stochastic gradient descent methods [44]. Such algorithms are also known as stochastic approximation algorithms. Let us denote by $\partial f_i(x)$, the subdifferential of $f_i(x)$. Then a centralized solution to the problem is given by the following projected subgradient algorithm.

$$x_{k+1} = P_{X} \left[ x_k - \alpha_k \sum_{i=1}^{m} d_i^k \right],$$

(4.0.2)

where $d_i^k \in \partial f_i(x_k)$, and $\alpha_k$ is a step size sequence. In [6], an incremental subgradient algorithm was proposed for this problem. This algorithm shares the sequential nature of the SOP algorithm. We now briefly discuss the algorithm.

### 4.1 Incremental Subgradient Algorithm

The incremental subgradient algorithm proceeds by incrementally updating the vector $x$ through a sequence of $m$ steps. At each step only the subgradient information corresponding to the objective function $f_i$ is used. Similar to the SOP algorithm let us define a control sequence as a mapping $\nu : \mathbb{N} \rightarrow \{1, \ldots, m\}$. Then the iterations of the incremental subgradient algorithm suitable adapted to the our problem can be written as

$$x_{k+1} = P_{X_{\nu(k)}} \left[ x_k - \alpha_k d_{\nu(k)}^k \right],$$

(4.1.1)

where $d_{\nu(k)}^k \in \partial f_{\nu(k)}(x_k)$. This algorithm recovers the SOP algorithm for the case when $f_i(x) = \frac{1}{2} \| x - P_{X_i}[x] \|^2$. In this case $d_i^k = x_k - P_{X_i}[x_k]$. Substituting back in Eq (4.1.1) for the case $\alpha_k = 1$ we get

$$x_{k+1} = P_{X_{\nu(k)}}[x_k],$$

(4.1.2)

which is the SOP algorithm Eq (3.1.2) for $\gamma_k = 1$. In its original form the incremental subgradient algorithm was designed for deterministic cyclical control sequences. This requirement was relaxed in [8], in which the authors considered the case when the control sequence could be a sample path of an evolving Markov chain.
4.2 The Stochastic Optimization Problem

In this section we briefly talk about how the stochastic optimization problem when the objective functions are of the form $f_i(x) = \mathbb{E}_z[L_i(x, z)]$, can be incorporated in our framework. Here $z$ is a random variable and the expectation is taken with respect to the unknown distribution of the random variable. Thus the function $f_i(x)$ is not known to agent $i$. The agent $i$ however has access to samples of $z$. Let $\nabla_x f(x)$ denote a subgradient of the objective function $f(x)$ then under some broad assumptions it can be shown that $\nabla_x f_i(x) = \mathbb{E}[\nabla_x L_i(x, z)]$. Hence, we can write $\nabla_x f_i(x) = \nabla_x L_i(x, z) + \mathbb{E}[\nabla_x L_i(x, z)] - \nabla_x L_i(x, z)$. Denoting $\epsilon_i := \mathbb{E}[\nabla_x L_i(x, z)] - \nabla_x L_i(x, z)$, we can see that it is a martingale difference sequence,

$$\mathbb{E}[\epsilon_i | F_k] = 0.$$ 

Thus, we can see that the stochastic optimization problem can be fit in the gradient descent framework by including an error term which satisfies the criterion of being a martingale difference sequence.

The rest of this Chapter is organized as follows. In Section 4.3 we discuss various example which motivate the specific problem structure we consider in (4.0.1). In Section 4.4 we state our main algorithm for the problem under consideration and provide our assumptions. In Section 4.5 we state and prove some preliminary results which are useful in deriving our convergence result in Section 4.6. Finally in Section 4.7 we provide a brief conclusion. This Chapter is based on the results provided in [34].

4.3 Distributed Optimization Problems Arising In Networks

In this section we discuss several problems which arise in sensor networks and emphasize that the common theme among these problems is the fact that they can be cast as distributed optimization problems.
4.3.1 Consensus and Robust Estimation

The problem of achieving consensus in a sensor network has seen tremendous amount of research in recent times. This is fueled by the fact that many problems arising in distributed control and estimation can be approached by using this machinery. The basic problem of consensus is the following. Let us assume that each sensor makes an observation at time 0 denoted as $z_i(0)$. The objective of the network is for each sensor to compute the average $\bar{z} = \frac{1}{m}z_0^i$ of the observations. It can be easily seen that this is equivalent to solving the following optimization problem

$$\min_x \frac{1}{m} \sum_{i=1}^m \|z_0^i - x\|^2.$$

The optimal solution to the above problem is given by $x^* = \frac{1}{m} \sum_{i=1}^m z_0^i$. A variant of the consensus problem was posed in [16], which the authors termed as the robust estimation problem. In their framework the sensors have the observations $\{z_t^i\}$ for $t = (0, \ldots, T)$. The objective is to again compute the average value $\tilde{z} = \frac{1}{mT} \sum_{t=0}^T \sum_{i=1}^m z_t^i$. However, unlike the quadratic loss function which is used in the consensus problem they proposed the Huber loss function, which is defined as

$$\rho(z, x) = \begin{cases} 
\|z - x\|^2 / 2 & \|z - x\| \leq \gamma \\
\gamma \|z - x\| - \gamma^2 / 2 & \|z - x\| > \gamma,
\end{cases}$$

where $\gamma$ is a fixed parameter. The objective of the sensor network is to solve the following optimization problem

$$\min_x \frac{1}{mT} \sum_{i=1}^m \sum_{t=0}^T \rho(z_t^i, x).$$

The choice of the Huber loss function makes the network less susceptible to outliers in the data collection process as less weight is given to data which deviates more than the chosen parameter $\gamma$.

4.3.2 Constrained Estimation

Now, let us consider the case of estimation of a parameter in a sensor network of heterogeneous sensors. In this formulation the assumption of Gaussian noise is relaxed to include noise with
bounded support. In the linear case the $i^{th}$ sensor observation is modeled as follows.

$$z_i(t) = \theta^T x_i(t) + w_i(k) \quad \text{for} \quad t = 0, \ldots, T$$

where the noise $w_i(t)$ has the following truncated Gaussian distribution.

$$p_{w_i}(w) = \begin{cases} 
\frac{1}{K \sqrt{2\pi \sigma_i^2}} \exp\left(-\frac{w^2}{2\sigma_i^2}\right) & \text{if } |w| \leq b_i \\
0 & \text{otherwise,} 
\end{cases}$$

where $K$ is the normalization constant. Here the truncated Gaussian noise models the sensor characteristics and are assumed to be different for different sensors. It is assumed that the initial a priori information about the parameter is modeled as a Gaussian $N(\mu_0, P_0)$ distribution. In this case it can be shown that the maximum posteriori estimate (MAP) is given by the solution of the following optimization problem

$$\min_{\theta} \sum_{i=1}^{m} \sum_{t=1}^{T} \frac{1}{2\sigma_i} |z_i(t) - \theta^T x_i(t)|^2 + \frac{1}{2}(\theta - \mu_0)^T P_0^{-1}(\theta - \mu_0)$$

subject to $|z_i(t) - \theta^T x_i(t)| \leq b_i \quad \forall \quad t = 1, \ldots, T \quad \forall i = 1, \ldots, m$

This is an optimization problem of the form

$$\min_{\theta} \sum_{i=1}^{m} f_i(\theta)$$

subject to $\theta \in \bigcap_{i=1}^{m} X_i$

where $X_i := \{\theta : |y_k^{i} - \theta^T x_k^{i}| \leq b_i \quad \forall \quad k = 1, \ldots, N\}$.

This formulation illustrates the need to come up with distributed solutions schemes for the parameter estimation problem when different sensors have a-priori information about the parameter lying in different convex constraint sets.
4.3.3 Network Utility Maximization in a Wireless Network

The problem of fair allocation of resources has been thoroughly studied in the area of microeconomics [21]. Recent interest in the resource allocation problem has arisen in the context of utility maximization in communication networks [24]. One of the most important characteristics of the network utility maximization problem is the fact that the objective function to be minimized has a separable form. Under this structure various primal or dual decomposition methods can be applied to make the problem amenable to a distributed solution [45]. Various problems arising in network utility maximization (NUM) can be formulated as one of maximizing

$$\max_x \sum_{j=1}^{m} U_j(x_1, \ldots, x_m)$$

$$x \in X.$$ 

An illustrative example is provided in [17], where it is shown that the problem of distributed power control in a wireless network can be formulated as

$$\min_x \sum_{i=1}^{m} \left[ \log \left( \sigma_i^2 h_{i,i} - 2 e^{-x_i} + \sum_{j \neq i} h_{i,i} h_{j,j} e^{x_j - x_i} \right) + V(e^{x_i}) \right]$$

subject to  $x \in X,$

where $V$ is a convex and increasing function. Clearly this problem fits in our framework. We show in Chapter 6 that a variant of the above sum utility maximization problem which maximizes the minimum utility allocated can also made to fit in our framework after a suitable transformation.

4.3.4 Distributed Model Predictive Control

Model predictive control also known as receding horizon control is a widely popular technique to solve infinite horizon optimal control problems. It is typically employed in situations when the presence of constraints renders the dynamic programming solution of the infinite horizon optimal control problem infeasible. It solves the infinite horizon problem in a batch framework, where in each batch a finite time horizon nonlinear programming problem is solved. For a detailed treatment
refer to [46]. Distributed model predictive control arises naturally in many situations when there are multiple plants and controllers connected over a network. A survey of different architectures employed for distributed MPC can be found in [47] and the problems associated with coordinating multiple MPC based controllers can be found in [48]. Recently this approach has been applied to the problem of coordination of multiple unmanned air vehicles [49]. The difficulty in solving the distributed MPC problem stems from the coupling, which arises between different agents through the cost function, dynamics or constraints. We present the distributed MPC formulation from [50], which fits our framework.

There are $m$ interconnected subsystems, where for each subsystem $i$ there is associated a local state $x_i \in \mathbb{R}^{n_i}$, and a control vector $u_i \in \mathbb{R}^{p_i}$. The coupling among the subsystems is captured by the graph $G = (V, E)$. The state equation for the $i_{th}$ subsystem is given as

$$
x_{i,k+1}^i = A_i x_{i,k}^i + \sum_{j \in N_i} B_{ij} u_j(k),$$

where $N_i$ is the set of neighbors of node $i$. The MPC problem is to solve the following optimization problem

$$\min_{\bar{u}} \sum_{i=1}^m \Phi_i = \sum_{i=1}^m \sum_{k=1}^T \left[ \frac{1}{2}([x_{i,k}^i]'Q_i x_{i,k}^i + u_i(k-1)'R_i u_i(k-1)] \right]$$

subject to

$$x_{i,k+1}^i = A_i x_{i,k}^i + \sum_{j \in N_i} B_{ij} u_j(k) \quad x_{i,k}^i \in \mathcal{X}_i, u_i(k) \in \mathcal{U}_i \quad \forall i,$$

where $\mathcal{U}_i$ is the local closed, convex and compact constraint set on the control action of system $i$. Let us denote

$$\tilde{x}_i = \begin{pmatrix} x_{i,1}^i \\ \vdots \\ x_{i,T}^i \end{pmatrix} \quad \text{and} \quad \tilde{u}_i = \begin{pmatrix} u_i(0) \\ \vdots \\ u_i(T-1) \end{pmatrix}.$$ 

Then, it can be verified that we can write

$$\tilde{x}_i = \tilde{A}_i \tilde{x}_i(0) + \sum_{j \in N_i} \tilde{B}_{ij} \tilde{u}_i,$$ 

(4.3.1)
where $\bar{A}_i$ and $\bar{B}_{ij}$ can be easily derived. Since the equation 4.3.1 is an affine function of $\bar{u}_i$, constraint $x_i^k \in X_i$ can now be translated into a convex constraint on $\bar{u}_i$, as $\bar{u}_i \in U'_i$. Thus, the variable $\bar{u}_i \in U_i \cap U'_i := U_i$. Note that $U_i$ is the constraint set in the product space. The optimization problem can similarly be written as

$$
\min_{\bar{u}} \sum_{i=1}^m \Phi_i
$$

$$
\Phi_i = \frac{1}{2} x_i^0 \bar{A}_i^T \bar{Q}_i \bar{A}_i x_i^0 + \sum_{j \in N_i} x_i^0 \bar{A}_i^T \bar{Q}_i \bar{B}_{ij} \bar{u}_j + \frac{1}{2} \sum_{j \in I(i)} \sum_{k \in N_i} \bar{u}_i \bar{B}_{ij} \bar{Q}_i \bar{B}_{ij} \bar{u}_j + \frac{1}{2} \bar{u}_i \bar{R}_i \bar{u}_i
$$

subject to $\bar{u}_i \in U_i$.

### 4.4 Main Algorithm

We propose the following subgradient algorithm:

$$
x_{i+1}^k = P_{X_i} \left[ x_k^i - \alpha_{k+1} \sum_{j \in N_i(k)} r_{ij,k+1} [x_k^j - (x_k^i + \xi_{ij,k+1})] - \gamma_{k+1} [d_{k+1}^i + \epsilon_{k+1}^i] \right].
$$

Here, the vector $d_k^i$ is a subgradient of the local objective function $f_i$ and $\epsilon_{k+1}^i$ is the error in the evaluation of the subgradient of $f_i(x)$ at $x = v_{k+1}^i$, where $v_{k+1}^i$ is given by (3.3.1). The step size $\gamma_{k+1} > 0$ is used to attenuate the subgradient error. Proceeding similarly to the consensus part let us rewrite this as follows.

$$
v_{k+1}^i = x_k^i + \alpha_{k+1} \sum_{j=1}^m r_{ij,k+1} x_k^j + \alpha_{k+1} \xi_{i,k+1}
$$

$$
x_{k+1}^i = P_{X_i} [v_{k+1}^i - \gamma_{k+1} [d_{k+1}^i + \epsilon_{k+1}^i]]. \quad (4.4.1)
$$

We denote the noisy subgradient by $\tilde{d}_k^i = d_k^i + \epsilon_k^i$. There are several interesting aspects of the algorithm (4.4.1). The algorithm relies on the consensus enforcing term in the calculation of $v_{k+1}^i$ to ensure that the agents estimates converge to a common optimal point. The idea of using projection on local constraint sets $P_{X_i} [\cdot]$ is taken from the SOP algorithms. Furthermore the algorithm requires two step size sequences $\alpha_k$ and $\gamma_k$ to damp the communication noise and subgradient
noise respectively. Our analysis reveals an interesting interplay between these step sizes for which almost convergence can be proven for the algorithm. The use of diminishing step size sequences to handle subgradient noise arises in the stochastic approximation literature. To sum up, the proposed algorithm uses aspects of consensus algorithms, SOP algorithms and the stochastic approximation technique to provide a distributed algorithm for the optimization problem (4.0.1).

We next state the various assumptions needed to derive our convergence result.

4.4.1 Assumptions

Let $\partial f_i(x)$ denote the set of subgradients of the objective function $f_i(x)$. We impose the following assumptions on the subgradients and the constraint sets.

**Assumption 4.**

1. The subgradient errors $\epsilon^i_k$ when conditioned on the point of evaluation of the subgradient $d^i_k$ are mean zero, i.e., $E[\epsilon^i_k | v^i_k] = 0$ for all $i$ and $k \geq 0$.

2. The subgradient errors further satisfy the bound $E[\|\epsilon^i_k\|^2 | v^i_k] \leq \nu^2$ for all $i$ and $k \geq 0$.

3. The local constraint sets $X_i$ are compact and convex.

4. The intersection set $X$ has a nonempty interior, i.e., there exists a point $\bar{z} \in X$ such that the ball $B_\delta := \{x \in X : \|x - \bar{z}\| \leq \delta\} \subset X$.

In addition to the above assumptions on the model, we use the following assumptions on the step sizes $\alpha_k$ and $\gamma_k$.

**Assumption 5.**

1. (non-summability) The step sizes satisfy $\sum_{k=1}^{\infty} \alpha_k = \infty$ and $\sum_{k=1}^{\infty} \gamma_k = \infty$.

2. (square summability) $\sum_{k=1}^{\infty} \alpha^2_k < \infty$ and $\sum_{k=1}^{\infty} \gamma^2_k < \infty$.

3. $\sum_{k=1}^{\infty} \alpha_k \gamma_k < \infty$ and $\sum_{k=1}^{\infty} \frac{\gamma_k^2}{\alpha_k} < \infty$.

4. $\sum_{k=1}^{\infty} \min\{\alpha_k, \gamma_k\} = \infty$. 

35
The Assumptions 5.1 and 5.2 are standard in the stochastic approximation literature. The square summability is needed to damp out the noise terms. In addition to these conditions, our analysis relies on Assumptions 5.3 and 5.4 on the cross terms involving the step sizes. To verify that the set of step sizes satisfying these conditions is non-empty, we can assume that the step sizes are of the form $\alpha_k = \frac{1}{k^\theta_1}$ and $\gamma_k = \frac{1}{k^\theta_2}$. Then conditions 5.1 and 5.2 imply that $1/2 < \theta_1, \theta_2 \leq 1$. It is clear that in this case $\sum_{k=1}^{\infty} \alpha_k \gamma_k < \infty$. The condition $\sum_{k=1}^{\infty} \frac{\gamma_k^2}{\alpha_k} < \infty$ implies that $\theta_2 > 1 + \frac{1+\theta_1}{2}$. Also, since when $\theta_2 > \theta_1$, we have $\min(\alpha_k, \gamma_k) = \gamma_k$; and in this case Assumption 5.4 holds by Assumption 5.1.

4.5 Preliminary Results

In this section we provide some results which will be useful in deriving our main result. The first result provides a way to bound an error term of the form $\|x_k^i - P_{X_i}[x_k^i]\|$. The bound is established by using some of the techniques in [32] for the alternating projection method.

**Lemma 4.5.1.** Let Assumptions 4.3 and 4.4 hold, and let $x_i \in X_i$ be variables belonging to local constraint sets $X_i$. Then, we have the following bound:

$$\|x_i - P_{X_i}[x_i]\| \leq \frac{B}{\delta} \sum_{j=1}^{m} \|x_i - x_j\| \quad \text{for all } i,$$

where $B$ is a uniform upper bound on the norms of the vectors in the sets $X_i$ and $\delta$ is the radius from Assumption 4.4.

**Proof.** Let $i$ be arbitrary. Define $\lambda_i = \sum_{j=1}^{m} \|x_i - P_{X_j}[x_i]\|$ and the variable $s_i$ as follows:

$$s_i = \frac{\lambda_i}{\lambda_i + \delta} \bar{z} + \frac{\delta}{\lambda_i + \delta} x_i,$$

where $\bar{z}$ is the interior point of the set $X$ from Assumption 4.4. Then, we can write

$$s_i = \frac{\lambda_i}{\lambda_i + \delta} \left[ \bar{z} + \frac{\delta}{\lambda_i} \left(x_i - P_{X_j}[x_i]\right) \right] + \frac{\delta_i}{\lambda_i + \delta} P_{X_j}[x_i].$$

From definition of $\lambda_i$, it is clear that $\|x_i - P_{X_j}[x_i]\| \leq \lambda_i$ for any $j$, implying by the interior point assumption that the vector $\bar{z} + \frac{\delta}{\lambda_i} \left(x_i - P_{X_j}[x_i]\right)$ lies in the set $X$ and hence, in set $X_j$ for any
Since the vector \( s_i \) is a convex combination of two vectors in the set \( X_j \), by the convexity assumption on the set \( X_j \), we have that \( s_i \in X_j \) for any \( j \). Therefore, we have \( s_i \in X \). Now, we can see that

\[
\|x_i - s_i\| \leq \frac{\lambda_i}{\lambda_i + \delta} \|x_i - \bar{z}\| \leq \frac{\|x_i - \bar{z}\|}{\delta} \sum_{j=1}^{m} \|x_i - P_{X_j}[x_i]\|
\]

By our assumption the sets \( X_i \) are compact, so \( \|x_i - \bar{z}\| \leq B \) for \( B > 0 \). Since \( x_j \in X_j \), by the properties of the projection operator it follows

\[
\|x_i - P_{X_j}[x_i]\| \leq \|x_i - x_j\|.
\]

Thus,

\[
\|x_i - P_{X}[x_i]\| \leq \|x_i - s_i\| \leq \frac{B}{\delta} \sum_{j=1}^{m} \|x_i - x_j\|.
\]

Let us define \( s_k = \frac{1}{m} \sum_{j=1}^{m} P_{X}[x_k^j] \), which belongs to the set \( X \) since \( X \) is closed and convex.

The following lemma is crucial in proving our convergence result in the next section.

**Lemma 4.5.2.** Let Assumptions 1, 2 and 4 hold. Then, for the algorithm proposed in Eq. (4.4.1) the following relation holds for any \( z^* \) in the optimal set \( X^* \),

\[
\mathbb{E}_k[\|x_k - z^*\|^2] \leq H(k+1) \|x_k - z^*\|^2 + \alpha_{k+1}^2 \mu^2 - \alpha_{k+1} \sum_{S(k)} \|x_k^i - x_k^j\|^2
\]

\[
+ m(\nu + C)^2 \gamma_{k+1}^2 + \frac{K^2 \gamma_{k+1}^2}{\eta \alpha_{k+1}} - 2 \gamma_{k+1} [f(s_k) - f(z^*)] + 2C \gamma_{k+1} \alpha_{k+1} \sum_{i=1}^{m} \sum_{j=1}^{m} r_{ij,k+1} x_{ij,k+1}^2,
\]

where \( C \) is a uniform bound on subgradient norms of \( f_i \) over the sets \( X_i \), \( K = C(m-1)(\frac{mB+\delta}{\delta}) \) and \( f(x) = \sum_{i=1}^{m} f_i(x) \).

**Proof.** By definition we have \( x_{k+1}^i = P_{X_i}[v_{k+1}^i - \gamma_{k+1} d_{k+1}^i] \). From the contraction property of the projection operator we see that for any \( z^* \in X^* \subseteq X \),

\[
\|x_{k+1}^i - z^*\|^2 \leq \|v_{k+1}^i - z^* - \gamma_{k+1} d_{k+1}^i\|^2
\]

\[
= \|v_{k+1}^i - z^*\|^2 + \gamma_{k+1}^2 \|d_{k+1}^i\|^2 - 2 \gamma_{k+1} (d_{k+1}^i)^T(v_{k+1}^i - z^*).
\] (4.5.1)
Taking conditional expectation, we obtain for any $z^* \in X^*$,

$$
\mathbb{E}_k[\|x_{k+1}^i - z^*\|^2] \leq \mathbb{E}_k[\|v_{k+1}^i - z^*\|^2] + 2\gamma_{k+1}^2 \mathbb{E}_k[\|\tilde{d}_{k+1}^i\|^2] - 2\gamma_{k+1} \mathbb{E}_k[(d_{k+1}^i)^T(v_{k+1}^i - z^*)] - 2\gamma_{k+1} \mathbb{E}_k[(\epsilon_{k+1}^i)^T(v_{k+1}^i - z^*)].
$$

Since, $\mathbb{E}_k[(\epsilon_{k+1}^i)^T(v_{k+1}^i - z^*)] = \mathbb{E}_k[(v_{k+1}^i - z^*)^T\mathbb{E}_k[\epsilon_{k+1}^i | v_{k+1}^i]]$ and $\mathbb{E}_k[\epsilon_{k+1}^i | v_{k+1}^i] = 0$ and $d_{k+1}^i$ is a subgradient of $f_i$ at $v_{k+1}^i$, we have

$$
\mathbb{E}_k[\|x_{k+1}^i - z^*\|^2] \leq \mathbb{E}_k[\|v_{k+1}^i - z^*\|^2] + \gamma_{k+1}^2 (C + \nu)^2 - 2\gamma_{k+1} \mathbb{E}_k[f_i(v_{k+1}^i) - f_i(z^*)],
$$

where we have used $\|d_{k}^i\| \leq C$ and $\mathbb{E}_k[\|\epsilon_{k+1}^i\|^2] \leq \nu^2$, and the Hölder’s inequality. Now, since $f_i$ is a convex function, by Jensen’s inequality $-\mathbb{E}_k[f_i(v_{k+1}^i)] \leq -f(\mathbb{E}_k[v_{k+1}^i])$. By the definition of $v_{k+1}^i$ in (4.4.1) and $\mathbb{E}_k[\xi_{ij}(k)] = 0$ of Assumption 2.1, we have $\mathbb{E}_k[v_{k+1}^i] = x_k^i + \alpha_{k+1} \sum_{j=1}^m r_{ij,k+1} x_{k+1}^j$.

Letting $w_{k+1}^i := x_k^i + \alpha_{k+1} \sum_{j=1}^m r_{ij,k+1} x_{k+1}^j$, we obtain

$$
\mathbb{E}_k[\|x_{k+1}^i - z^*\|^2] \leq \mathbb{E}_k[\|v_{k+1}^i - z^*\|^2] + \gamma_{k+1}^2 (C + \nu)^2 - 2\gamma_{k+1} \mathbb{E}_k[f_i(w_{k+1}^i) - f_i(z^*)].
$$

Summing over all $i$ and using vector notation yield

$$
\mathbb{E}_k[\|x_{k+1} - z^*\|^2] \leq \mathbb{E}_k[\|v_{k+1} - z^*\|^2] + m\gamma_{k+1}^2 (C + \nu)^2 - 2\gamma_{k+1} \sum_{i=1}^m [f_i(w_{k+1}^i) - f_i(z^*)].
$$

As seen in the proof of Lemma 3.3.2, we have

$$
\mathbb{E}_k[\|v_{k+1} - z^*\|^2] \leq H(k + 1) \|x_k - z^*\|^2 + \alpha_{k+1}^2 \mu^2 - 2\alpha_{k+1} \sum_{S(k)} \|x_k^i - x_{k+1}^i\|^2.
$$

Let $s_k \in X$ be as defined earlier. Adding and subtracting $f_i(s_k)$, and using $f(x) = \sum_{i=1}^m f_i(x)$ and the above two relations, we obtain

$$
\mathbb{E}_k[\|x_{k+1} - z^*\|^2] \leq H(k + 1) \|x_k - z^*\|^2 + m\gamma_{k+1}^2 (C + \nu)^2 - 2\gamma_{k+1} \sum_{i=1}^m [f_i(w_{k+1}^i) - f_i(s_k)] + \alpha_{k+1}^2 \mu^2 - 2\alpha_{k+1} \sum_{S(k)} \|x_k^i - x_{k+1}^i\|^2.
$$

38
By the convexity and subgradient boundedness of each $f_i$, $|f_i(w_{k+1}^i) - f_i(s_k)| \leq C \|w_{k+1}^i - s_k\|$, implying

$$
\mathbb{E}_k[\|x_{k+1} - z^*\|^2] \leq H(k + 1) \|x_k - z^*\|^2 + m\gamma_{k+1}^2(C + \nu)^2 - 2\gamma_{k+1}\|f(s_k) - f(z^*)\| \\
+ 2\gamma_{k+1}C \sum_{i=1}^m \|w_{k+1}^i - s_k\| + \alpha_{k+1}^2 \sum_{j=1}^m \|x^j_k - x^j_k\|^2. \quad (4.5.2)
$$

We now estimate the term with $\|w_{k+1}^i - s_k\|$. Since $w_{k+1}^i = x_k^i + \alpha_{k+1} \sum_{j=1}^m r_{ij,k+1}x_k^j$, it follows that

$$
2C\gamma_{k+1} \sum_{i=1}^m \|w_{k+1}^i - s_k\| \leq 2C\gamma_{k+1} \sum_{i=1}^m \|x_k^i - s_k\| + 2C\gamma_{k+1}\alpha_{k+1} \sum_{i=1}^m \sum_{j=1}^m r_{ij,k+1}x_k^j. \quad (4.5.3)
$$

Next, we focus on the term $\sum_{i=1}^m \|x_k^i - s_k\|$. Substituting for $s_k = \frac{1}{m} \sum_{j=1}^m P_X[x_k^j]$, adding and subtracting the term $P_X[x_k^i]$ inside the norm and using the convexity of norm, we have

$$
\sum_{i=1}^m \|x_k^i - s_k\| = \sum_{i=1}^m \|x_k^i - \frac{1}{m} \sum_{j=1}^m P_X[x_k^j]\| \leq \sum_{i=1}^m \|x_k^i - P_X[x_k^i]\| + \frac{1}{m} \sum_{i=1}^m \sum_{j=1}^m \|P_X[x_k^j] - P_X[x_k^i]\|.
$$

By Lemma 4.5.1 we have that

$$
\|x_k^i - P_X[x_k^i]\| \leq \frac{B}{\delta} \sum_{j=1}^m \|x_k^i - x_k^j\|,
$$

and by the non-expansiveness property of projection $\|P_X[x_k^i] - P_X[x_k^j]\| \leq \|x_k^i - x_k^j\|$. Hence,

$$
\sum_{i=1}^m \|x_k^i - s_k\| \leq \frac{mB + \delta}{m\delta} \sum_{i=1}^m \sum_{j=1}^m \|x_k^i - x_k^j\| = 2 \frac{mB + \delta}{m\delta} \sum_{i<j} \|x_k^i - x_k^j\|, \quad (4.5.4)
$$

where the notation $\sum_{i<j}$ denotes $\sum_{i=1}^m \sum_{j=i+1}^m$. For any two nodes $i$ and $j$, there is a path from node $i$ to node $j$ in the spanning tree $S(k)$. By suppressing the dependence on time, we represent such a path by $p_{ij} = (i_1, \ldots, i_\ell)$, where $i_1 = i$, $i_\ell = j$ and $\{i_\tau, i_{\tau+1}\}$ is edge in the spanning tree for $1 \leq \tau \leq \ell - 1$. Therefore, we have $\|x_k^i - x_k^j\| \leq \sum_{\tau=1}^{\ell-1} \|x_{i_{\tau+1}(k)}(k) - x_{i_\tau(k)}(k)\|$. Let all the (undirected) edges in the tree $S(k)$ be enumerated by $e_s = \{i_s, i_{s+1}\}$ for $1 \leq s \leq m - 1$. By
summing over $i$ and $j$ with $i < j$ we get
\[ \sum_{i<j} \| x^i_k - x^j_k \| \leq \sum_{e_s \in S(\mathcal{k})} \kappa_i \| x_{i*}(k) - x_{i+1}(k) \|, \]
where $\kappa_i$ denotes the number of times the edge $e_s$ appears in the collection of the paths $\{p_{ij}, i < j\}$. A simple upper bound on the number $\kappa_i$ is given by $\binom{m}{2}$. Thus,
\[ \sum_{i<j} \| x^i_k - x^j_k \| \leq \left( \binom{m}{2} \right) \sum_{S(k)} \| x^i_k - x^j_k \|, \]
where the summation on the right hand side is over all edges $\{i,j\}$ in $S(k)$. Using the preceding estimate in relation (4.5.4), we obtain
\[ \sum_{i=1}^{m} \| x^i_k - s_k \| \leq \frac{mB + \delta}{\delta} (m - 1) \sum_{S(k)} \| x^i_k - x^j_k \|. \]

Multiplying the preceding relation by $2\gamma_k K$ and letting $K = C(m - 1) \frac{mB + \delta}{\delta}$, we have
\[ 2\gamma_k K \sum_{i=1}^{m} \| x^i_k - s_k \| \leq 2\gamma_k K \sum_{S(k)} \| x^i_k - x^j_k \|. \tag{4.5.5} \]

By using the identity $2ab \leq a^2 + b^2$, we obtain
\[ 2\gamma_k K \sum_{S(k)} \| x^i_k - x^j_k \| = 2\gamma_k K \sum_{S(k)} \sqrt{\eta \alpha_k + 1} \| x^i_k - x^j_k \| \leq \gamma_k \frac{K^2}{\eta \alpha_k + 1} + \eta \alpha_k + 1 \sum_{S(k)} \| x^i_k - x^j_k \|^2. \tag{4.5.6} \]

Substituting the preceding inequality in relation (4.5.5), and then combining the resulting relation with Eq. (4.5.3), we obtain the following estimate:
\[ 2\gamma_k C \sum_{i=1}^{m} \| w^i_{k+1} - s_k \| \leq \gamma_k \frac{K^2}{\eta \alpha_k + 1} + \eta \alpha_k + 1 \sum_{S(k)} \| x^i_k - x^j_k \|^2 + 2C\gamma_k \sum_{i=1}^{m} \sum_{j=1}^{m} r_{ij,k+1} x^j_k. \tag{4.5.7} \]

Substituting the preceding relation in Eq. (4.5.2), we arrive at the desired result. \qed
4.6 Convergence Result

In this section we prove our main convergence result for the projected subgradient algorithm.

**Theorem 4.6.1.** Let Assumptions 1, 2 and 4 hold. Then if the step size sequences \(\{\alpha_k\}\) and \(\{\gamma_k\}\) in the algorithm (4.4.1) are chosen to satisfy the conditions in Assumption 5, then almost surely the iterate sequences \(\{x_k^i\}\) converge to a common (random) point in the optimal set \(X^*\).

**Proof.** Referring to Lemma 4.5.2, if the step size sequences are chosen to satisfy Assumption 5.2 we have
\[
\sum_{k=1}^{\infty} \left[ \alpha_{k+1}^2 \mu^2 + m(C + \nu)^2 \gamma_{k+1}^2 \right] < \infty. \tag{4.6.1}
\]
By Assumption 5.3 we have \(\sum_{k=1}^{\infty} \frac{k^2 \gamma_k^2}{\eta \alpha_{k+1}} < \infty\). The term \(\sum_{i=1}^{m} \left\| \sum_{j=1}^{m} r_{ij,k+1} x_k^i \right\|\) is bounded since the sets \(X_i\) are compact by Assumption 4.3 and \(r_{ij,k+1} \leq r'\) by Assumption 1.4. Also, we have \(\sum_{k=1}^{\infty} \alpha_k \gamma_k < \infty\) by Assumption 5.3, so we can conclude that
\[
2C \sum_{k=1}^{\infty} \gamma_{k+1} \alpha_{k+1} \sum_{i=1}^{m} \left\| \sum_{j=1}^{m} r_{ij,k+1} x_k^i \right\| < \infty. \tag{4.6.2}
\]
Now, since \(H(k+1) = (1 + \alpha_{k+1}^2 \sigma_m^2(R_{k+1}))\) and the elements of matrix \(R_k\) are uniformly bounded, we have \(\sigma_m^2(R_{k+1}) \leq C\). Hence \(\sum_{k=1}^{\infty} \alpha_{k+1}^2 \sigma_m^2(R_{k+1}) \leq C \sum_{k=1}^{\infty} \alpha_{k+1}^2 < \infty\). Also since \(z^*\) belongs to the optimal set \(X^*\) and \(s_k \in X\), we have \(f(s_k) - f(z^*) \geq 0\) for all \(k\). Thus, we can apply the result of Robbins-Siegmund (Lemma 2.3.2) to conclude that \(||x_{k+1} - z^*||^2\) converges almost surely for any \(z^* \in X^*\) and the following holds almost surely:
\[
\sum_{k=1}^{\infty} \left[ \eta \alpha_{k+1} \sum_{S(k)} \left\| x_k^i - x_k^j \right\|^2 + \gamma_{k+1} (f(s_k) - f(z^*)) \right] < \infty.
\]
Letting \(\theta(k) = \min(\alpha_k, \gamma_k)\), we can conclude that
\[
\sum_{k=1}^{\infty} \theta(k+1) \left[ \eta \sum_{S(k)} \left\| x_k^i - x_k^j \right\|^2 + (f(s_k) - f(z^*)) \right] < \infty,
\]
with $\sum_{k=1}^{\infty} \theta(k) = \infty$ (Assumption 5.4). Therefore, there is a subsequence such that

$$\lim_{k \to \infty} \sum_{S(n_k)} \|x_{n_k}^i - x_{n_k}^j\|^2 = 0$$

and $\lim_{k \to \infty} f(s(n_k)) = f(z^\ast)$ almost surely.

Since the number of spanning trees on a finite graph is finite, there exists a spanning tree $S$ which appears infinitely often in the sequence $\{S(n_k)\}$. Let us pick a further subsequence such that $S(n_k^1) = S$, then we have along this subsequence $\lim_{k \to \infty} \sum_{S} \|x_{n_k^1}^i - x_{n_k^1}^j\|^2 = 0$ and $\lim_{k \to \infty} f(s(n_k^1)) = f(z^\ast)$ almost surely. Since $S$ is a spanning tree of a connected graph we can conclude that $\lim_{k \to \infty} \|x_{n_k^1}^i - x_{n_k^1}^j\| = 0$ for all nodes $i$ and $j$ almost surely. Using the earlier proven bound in equation (4.5.4) and taking limit along the time subsequence $\{n_k^1\}$, we obtain

$$\lim_{k \to \infty} \sum_{i=1}^{m} \| x_{n_k^1}^i - s_{n_k^1}^i \| \leq \lim_{k \to \infty} 2 \left( \frac{mB + \delta}{m\delta} \right) \sum_{i < j} \| x_{n_k^1}^i - x_{n_k^1}^j \| = 0,$$

implying that we have almost surely

$$\lim_{k \to \infty} \| x_{n_k^1}^i - s(n_k^1) \| = 0 \quad \text{for all nodes } i. \quad (4.6.3)$$

Since $\lim_{k \to \infty} f(s_{n_k^1}) = f^\ast$ and the function $f$ is continuous we conclude that there exists a subsequence along which $s_{n_k^1}$ converges almost surely to a (random) point $x^\ast$ that lies in the set $X^\ast$. Without any loss of generality we can assume that the sequence $s_{n_k^1}$ itself converges to the limit point $x^\ast$ almost surely. From equation (4.6.3) we conclude that $x_{n_k^1}^i$ converges to $x^\ast$ for all $i$ almost surely. However from our conclusion earlier we know that $\|x_k - z^\ast\|^2$ converges almost surely for any $z^\ast \in X^\ast$. We can consider sample paths for which both $\|x_k - z^\ast\|^2$ converges for any $z^\ast \in X^\ast$ and $x_{n_k^1}^i$ converges to the corresponding realization $\tilde{x}^\ast \in X^\ast$ of the random point $x^\ast$. Then, by letting $z^\ast = x^\ast$, we can conclude that for each such realization, the realization of the sequence $\{x_{k}^i\}$ converges to the corresponding realization $\tilde{x}^\ast$ of the random point $x^\ast \in X^\ast$. Hence, the sequences $\{x_{k}^i\}, i = 1, \ldots, m$, converge almost surely to a common (random) point in the set $X^\ast$. \hfill \Box
4.7 Conclusion

In this chapter we considered the distributed optimization problem which is central to the current work. We considered a setup which allows for noisy communication links, stochastic subgradient errors and a time varying communication network. A two step size sequence approach to deal with the communication and subgradient noise was developed which uses the robustness properties of consensus based algorithms to establish almost sure convergence to the optimal solution. We also provided the unique condition relating the step size sequences which falls out of our analysis.
Chapter 5

Asynchronous Algorithms

In this chapter we deal with both the constrained consensus problem of Chapter 3 and the problem of distributed optimization, introduced in Chapter 4 in an asynchronous setting. Most of the work on the consensus problem deals with the unconstrained case when the variables on which the nodes need to agree are free to lie in the Euclidean space. We deal with the case when the variables which are local to nodes are also constrained to lie in closed convex sets. The constraint set for each local variable is private information to the node. The objective is to design an algorithm which is adapted to the time varying random nature of the underlying communication graph between nodes and guarantees asymptotic consensus on the local variables while maintaining the feasibility of each variable with respect to its constraint set. A distributed algorithm for this problem was proposed in [11]. However, the analysis there was restricted to the case of a deterministic network, with noiseless communication links. Unlike [11], here we consider the case when the communication graph is random and the communication links are noisy. In this case we extend the algorithm proposed in [11] by introducing a step size sequence that attenuates the communication noise. A new feature of the algorithm is that the step sizes of the agents are asynchronous and are a function of their local clocks.

Next we consider a distributed constrained stochastic optimization problem. A distributed optimization algorithm for the case when the objective functions are deterministic functions has been proposed in [11], but its convergence analysis was limited to two special cases: when the local constraint sets are identical and when the network is fully connected (requiring the nodes to use uniform weights). We consider a more general problem than [11], by fully studying the presence of local constraint sets and noisy communication along with the presence of stochastic errors in the evaluation of subgradients. Once again we consider a random communication network. In this case we need to introduce two step size sequences to damp out both communication noise and subgradi-
ent errors arising from considering the stochastic optimization problem. We prove that if the step size damping the subgradient error decays fast enough when compared to the step size attenuating the communication noise, then the algorithm converges to a common point in the optimal set with probability one. Our distributed optimization algorithm is asynchronous in nature and, for each agent, both the step sizes are functions of the agent’s local clock.

Our model of the random communication network is general enough to include both the gossip communication protocol of [51] and the broadcast protocol of [52]. We also consider the case when the algorithm is employed by using constant step sizes. As expected, in this case it is not possible to achieve convergence to the optimal set with probability one. Instead, we derive asymptotic error bounds on the iterates of the algorithm under some additional assumptions.

Consensus over noisy links in the lack of constraint sets has been studied in [37], [38] and [13] among others. In [7], the authors studied the distributed optimization problem in the presence of subgradient errors. However, the paper assumes a common constraint set and the absence of communication noise. A related asynchronous scheme for the distributed optimization problem is the Markov incremental algorithm proposed in [8]. According to this algorithm the index of agent updating the common decision vector is decided by the evolution of a Markov chain. A limitation of this algorithm is its serial nature. In [9], the authors consider a random network model for communication among agents, but the optimization problem is unconstrained and the proposed approach is synchronous in nature in the sense that the step sizes of the different agents are coordinated. Another relevant paper which considers the unconstrained random network model for consensus is [53].

The rest of the chapter is organized as follows. In Section 5.1 we setup the problem, state our algorithms for both the problem of constrained consensus and distributed optimization, and introduce our main assumptions. In Section 5.2 we state some results regarding asymptotic properties of some deterministic and random sequences which will be useful in deriving our result on asymptotic convergence and the asymptotic error bounds. In Section 5.3 we tackle the problem of constrained consensus. We prove the asymptotic convergence result of our algorithm. We also consider the case of constant step size and establish an asymptotic error bound for averaged iterates. Next, in Section 5.4 we provide an asymptotic convergence result for our asynchronous distributed optimiza-
tion algorithm, and derive an asymptotic error bound for the case of constant step sizes. Finally in Section 5.5 we provide the conclusion. This Chapter is based on the published work [54].

5.1 Network Model, Outline of Algorithms and Assumptions

In this section we introduce the main problems we consider. We provide our algorithms for these problems and discuss the assumptions which we use in our convergence analysis. We consider a setup with a set of \( m \) agents, which can be viewed as the node set \( V = \{1, \ldots, m\} \). We further assume that the possible communication links among agents are given by a bidirectional graph 
\( G = (V, E) \). The communication graph at time slot \( k \) is represented by the random graph \( \mathcal{W}(k) = (V, \mathcal{E}(k)) \), with \( \mathcal{E}(k) \subseteq E \). Clearly the random graph \( \mathcal{W}(k) \) is a subgraph of the graph \( G \). Note that \( \mathcal{W}(k) \) is not required to be a bidirectional graph. The use of random communication graphs enables us to include various communication protocols which are prevalent in the wireless network literature such as gossip based protocols [51] and broadcast based protocol [52].

In the current work we are mainly concerned with the problem of constrained consensus and distributed optimization in the presence of various uncertainties including noisy communication links, random communication graphs, and stochastic errors in the evaluation of subgradients of the objective function. We will model the communication events (when any two agents \( i \) and \( j \) communicate) as occurring at the ticks of a global virtual Poisson clock \( T_p \). It has been shown [51], [52], that this abstraction fits the framework of asynchronous computation in both the gossip and broadcast framework. Let us denote the local decision variable associated with node \( i \) at time \( k \) as \( x^i_k \). Each local decision variable \( x^i_k \) is restricted to lie in a local constraint set \( X_i \).

5.1.1 Constrained Consensus:

The constrained consensus problem introduced in Chapter 3 is to achieve asymptotic consensus on the local decision variables through information exchange with the neighboring nodes in the presence of the constraint sets. Alternatively, the problem can be cast as a quest for a distributed
algorithm for the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} \|x - P_{X_i}[x]\|^2 \\
\text{subject to} & \quad x \in \mathbb{R}^n.
\end{align*}
\] (5.1.1)

When the intersection \( \cap_{i=1}^{m} X_i \) is nonempty, a solution to the above problem is given by any vector \( x \) which lies in the intersection. Clearly, in this case the objective function value is zero, which is also the optimal value. A distributed algorithm for this problem was proposed in [11]. In the algorithm agent \( i \)'s local variable \( x^i_k \) evolves as follows:

\[
x^i_{k+1} = P_{X_i} \left[ \sum_{j=1}^{m} r_{ij,k+1} x^j_k \right],
\]

where \( r_{ij,k+1} \) denotes the weight assigned by node \( i \) to the estimate coming from node \( j \). A crucial assumption needed in the analysis in [11] was the requirement that if agent \( i \) receives data from agent \( j \) then \( r_{ij,k} \geq \eta > 0 \), uniformly in \( k \). We are interested in the case when the communication links are noisy and, hence, node \( i \) has access to a noise corrupted value of its neighbor’s local estimate. In this case it is detrimental to impose the requirement that \( r_{ij,k} \geq \eta \) since we need to asymptotically damp the impact of the noise. We formulate our algorithm for the general case when, at the \( k^{th} \) tick of the global Poisson clock \( T_p \), the communication graph \( W(k) = (V, E(k)) \) is random, i.e., the edge set \( E(k) \) is random. A similar formulation to ours was carried out in [55], but without local set constraints. We will assume that the noise is additive in nature. With this in consideration, let us define the random variables for all \( i \in V \) as

\[
v^i_{k+1} = x^i_k + a^i_{k+1} \sum_{j=1}^{m} a_{ij,k+1}[x^j_k + \xi_{ij,k+1} - x^i_k],
\] (5.1.2)

where \( \xi_{ij,k+1} \) is the additive noise on the link \( (i,j) \). The weights \( a_{ij,k+1} > 0 \) encode the random network and the relative weights node \( i \) imposes on the communicated variables from its neighboring nodes at instance \( k + 1 \). If a node \( i \) doesn’t receive any information from node \( j \) at instance \( k + 1 \), then \( a_{ij,k+1} = 0 \). Clearly \( a_{ij,k+1} = 0 \) for all \( k \), if the edges \( (i,j) \notin E(W(k+1)) \). We assign \( a_{ii,k} = 0 \), uniformly in \( k \). If an agent \( i \) doesn’t receive any information from its neighbors, then
\( v_{k+1}^i = x_k^i \). We define the matrix \( W_{k+1} \in \mathbb{R}^{m \times m} \) such that \([W_{k+1}]_{ij} = -a_{ij,k+1} \) for \( i \neq j \) and \([W_{k+1}]_{ii} = \sum_{j \neq i} a_{ij,k+1} \). Then, \( W_{k+1} \) has the property \( W_{k+1} \mathbf{1}_m = 0 \) for all \( k \). Under the new notation relation (5.1.2) can be written as

\[
v_{k+1}^i = x_k^i - \alpha_{k+1}^i \sum_{j=1}^m [W_{k+1}]_{ij} x_k^j - \alpha_{k+1}^i \sum_{j \neq i} [W_{k+1}]_{ij} \xi_{ij,k+1} \quad \text{for all } i \in V. \tag{5.1.3}
\]

Define the set \( U_k = \{ j \in V \mid [W_{k+1}]_{jl} < 0 \text{ for some } l \neq j \} \) and \( \Gamma_{i,k+1} = \sum_{t=1}^{k+1} \chi_{\{i \in U_t\}} \). Essentially the set \( U_{k+1} \) denotes the set of agents updating their iterates at instance \( k+1 \), and \( \Gamma_{i,k+1} \) denotes the number of times agent \( i \) has updated its value until time \( k+1 \). The asynchronous behavior of the algorithm becomes apparent in the way the agents use the step sizes \( \alpha_k^i \). We start with the assumption that all the agents know the functional form of the step sizes \( \alpha_k^i \), for example \( \alpha_k = \frac{1}{k^\theta} \) with a parameter \( \theta \). For an agent \( i \) updating at time \( k+1 \), the step size is given by \( \alpha_{k+1}^i = \alpha_{\Gamma_{i,k+1}} \).

Let us denote the total noise experienced by agent \( i \) at time \( k+1 \) in Eq. (5.1.3) by \( \xi_{k+1}^i \). Then, our algorithm for the constrained consensus problem is given as

\[
v_{k+1}^i = x_k^i - \alpha_{k+1}^i \sum_{j=1}^m [W_{k+1}]_{ij} x_k^j - \alpha_{k+1}^i \xi_{k+1}^i \xi_{k+1}^i \quad \text{for all } i \in V.
\tag{5.1.4}
\]

where \( P_{X_i} [\cdot] \) is the projection on the local constraint set \( X_i \). The local variables always satisfy \( x_k^i \subseteq X_i \), and if at any instant the node \( i \) doesn’t receive any information from its neighbors (\([W_{k+1}]_{ij} = 0 \) for all \( j \neq i \)), then \( x_{k+1}^i = x_k^i \).

### 5.1.2 Distributed Optimization:

We now discuss the problem of distributed optimization which was introduced in Chapter 4. As mentioned in Chapter 4 the objective of the agents is to cooperatively solve the following constrained
optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i=1}^{m} f_i(x) \\
\text{subject to} & \quad x \in X = \bigcap_{i=1}^{m} X_i,
\end{align*}
\]

where each \( f_i : \mathbb{R}^n \to \mathbb{R} \) is a convex function, representing the local objective function of agent \( i \), and each set \( X_i \subseteq \mathbb{R}^n \) is compact and convex, representing the local constraint set of agent \( i \). We propose the following update mechanism for each agent:

\[
v_{i}^{k+1} = x_{i}^{k} - \alpha_{k+1}^{i} \sum_{j=1}^{m} [W_{k+1}]_{ij} x_{j}^{k} - \alpha_{k+1}^{i} \xi_{k+1}^{i}
\]

\[
\xi_{k+1}^{i} := \sum_{j \neq i} [W_{k+1}]_{ij} \xi_{ij,k+1}
\]

\[
x_{i}^{k+1} = P_{X_i} \left[ v_{k+1}^{i} - \gamma_{k+1}^{i} \tilde{d}_{k+1}^{i} \chi_{\{i \in U_{k+1}\}} \right].
\]

The algorithm is an adjustment of the consensus algorithm in (5.1.4), where the agent \( i \) update of \( x_{i}^{k+1} \) is modified to account for the local objective function \( f_i \). We denote the noisy subgradient by \( \tilde{d}_{k}^{i} = d_{k}^{i} + \epsilon_{k}^{i} \) for all \( k \). Here, the vector \( d_{k}^{i} \) is a subgradient of the local objective function \( f_i \) and \( \epsilon_{k}^{i} \) is the error in the evaluation of the subgradient of \( f_i(x) \) at \( x = v_{k}^{i} \). A unique feature of our algorithm is the ability to handle both communication noise and subgradient error. The presence of two sources of noise makes it imperative to use two step size sequences \( \alpha_{k}^{i} \) and \( \gamma_{k}^{i} \) corresponding to the communication noise and subgradient error respectively. Once again the step size sequences employed by the agent \( i \) is given as \( \alpha_{k}^{i} = \alpha_{r_i,k} \) and \( \gamma_{k}^{i} = \gamma_{r_i,k} \) (see (5.2.1) for specific form of the step sizes); hence, both sequences are functions of the number of times agent \( i \) updates its variable.

### 5.1.3 Assumptions and Implications

In this section we introduce various assumptions which we use to prove convergence of our algorithms for both constrained consensus and the distributed optimization problem. We group the assumptions into three categories; namely, network assumptions, constraint sets and objective functions assumptions, and the assumptions on communication noise and subgradient errors. Typically
in our results we only require a subset of the following assumptions to hold. To prevent imposing blanket assumptions for our results, we explicitly refer the assumptions required in each of our result.

Network Assumptions. Our first assumption lists all the conditions on the underlying communication graph that will be used, not necessarily all at once. Recall that the graph \( G = (V, E) \) has the edge set \( E \) consisting of all possible communication links between nodes, and that \( \mathcal{W}(k) = (V, \mathcal{E}(k)) \) is the random communication graph at time \( k \). Also, recall that the matrices \( W_k \) are defined by

\[
[W_k]_{ij} = -a_{ij,k} \text{ for } i \neq j \text{ and } [W_k]_{ii} = \sum_{j \neq i} a_{ij,k}, \tag{5.1.6}
\]

where \( a_{ij,k} \) are the relative weights. We use the following conditions on the graph \( G \) and the weight matrices.

**Assumption 6.** Suppose that the following hold:

a) The graph \( G = (V, E) \) is bidirectional, connected and without self-loops i.e., \((i, i) \notin E\), for all \( i \).

b) For all edges \((i, j) \in \mathcal{E}(k)\), we have \( 0 < a_{ij,k} \leq \eta \) and \( a_{ij,k} = 0 \) otherwise.

c) The random matrix sequence \( \{W_k\} \) is independent identically distributed (i.i.d.). The expected matrix \( \bar{W} = \mathbb{E}[W_k] \) is positive semidefinite and the vector \( \mathbf{1}_m \) is the unique (up to scaling) nonzero vector satisfying \( \bar{W} \mathbf{1}_m = 0 \).

d) The probability \( p_i \) of update at any instance for each node \( i \) is strictly nonzero, i.e., \( p_i > 0 \).

By the definition of the weight matrices \( W_k \) in (5.1.6), Assumption 6-b implies that

\[
-\eta \leq [W_k]_{ij} < 0 \text{ for } (i, j) \in \mathcal{E}(k) \text{ and } [W_k]_{ij} = 0 \text{ for } (i, j) \notin \mathcal{E}(k). \tag{5.1.7}
\]

Note that Assumption 6-c does not require that the random communication graph \( \mathcal{W}(k) \) is connected at any instance. However, the expected graph \( \bar{W} = (V, \mathbb{E}[\mathcal{E}(k)]) \) is connected in view of the assumption on the vector \( \mathbf{1}_m \). As shown later in Lemmas 5.1.1 and 5.1.2, the expected weight matrix arising in gossip and broadcast communication protocol satisfy the positive semi-definiteness...
condition of Assumption 1-c. Furthermore, under the i.i.d condition of Assumption 6-c, the random variables \( \chi_{\{i \in U_{k+1}\}} \) are also i.i.d for any \( i \in V \). Thus, the stationary probability \( p_i \) of update at any instance by agent \( i \) is well defined, and we have \( p_i = \mathbb{E}[\chi_{\{i \in U_{k+1}\}}] \).

**Constraints sets and objective functions.** We here discuss the assumptions we make on the sets \( X_i \) and the functions \( f_i \).

**Assumption 7.** Let the following conditions hold:

a) The constraint sets \( X_i \subseteq \mathbb{R}^n \) are closed and convex. Their intersection set \( X = \cap_{i=1}^m X_i \) is nonempty.

b) The constraint sets \( X_i \) are compact.

c) The intersection set \( X \) has a nonempty interior, i.e., there exists a point \( \bar{z} \in X \) such that the \( \delta \)-ball \( B_\delta = \{x \in X : \|x - \bar{z}\| \leq \delta\} \subset X \).

d) The local objective functions \( f_i(x) \) are defined and convex over \( \mathbb{R}^n \).

e) We assume that the subgradients of each function \( f_i \) are uniformly bounded i.e., there is a scalar \( C > 0 \) such that \( \|d\| \leq C \) for all \( d \in \partial f_i(x), \) all \( x \in \mathbb{R}^n, \) and all \( i \).

When the sets \( X_i \) are compact, we will use the notation

\[
C_{X_i} = \max_{x,y \in X_i} \|x - y\| \quad \text{for all } i \in V.
\]

Under Assumption 7-d, it is known [28] that each function \( f_i(x) \) is differentiable almost everywhere. Wherever the functions are not differentiable, the subdifferential set \( \partial f_i \) is nonempty. Note that even when the sets \( X_i \) are compact, we still assume the subgradient boundedness for each \( f_i \) since the subgradients are evaluated at points \( v_{k+1}^i \), which need not lie in the set \( X_i \).

**Communication noise and subgradient error.** We now state the assumptions we use for the link noise and the stochastic subgradient errors. For this, let us denote the \( \sigma \)-algebra generated by the entire history of the algorithm up to time \( k \), as follows:

\[
F_k = \{x_{i,0}, W_l, \xi_{ij,l}, \epsilon_{i,l}; 1 \leq l \leq k, i, j \in V\}.
\]
Assumption 8. Let the following hold:

a) The noise is zero mean, $E[\xi_{ij,k+1} | F_k, W_{k+1}] = 0$ for any link $(i, j) \in \mathcal{E}(k + 1)$.

b) For any realization of the random graph $\mathcal{W}(k+1)$, the noise on link $(i, j)$ is independent of the noise on link $(i', j')$, where $(i, j) \neq (i', j')$.

c) There is a uniform bound on the noise variance, i.e., there is $\mu > 0$ such that for all $(i, j) \in \mathcal{E}(k + 1)$ and $k \geq 0$,

$$E[\|\xi_{ij,k+1}\|_2^2 | F_k, W_{k+1}] \leq \mu^2.$$ 

d) The error $\epsilon_{i,k+1}^i$ associated with the subgradient $d_{i,k+1}^i$ of the function $f_i(x)$ at $x = v_{k+1}^i$ is such that for all $k \geq 0$

$$E[\epsilon_{i,k+1}^i | F_k, W_{k+1}, v_{k+1}^i] = 0 \quad \text{for all } i \in U_{k+1}.$$ 

e) The subgradient errors satisfy the following bound

$$E[\|\epsilon_{i,k+1}^i\|_2^2 | F_k, W_{k+1}, v_{k+1}^i] \leq \nu^2,$$

for all $i \in U_{k+1}$ and $k \geq 0$.

For the algorithm in Eq. (5.1.5), we have $\xi_{i,k+1}^i = \sum_{j \neq i} [W_{k+1}]_{ij} \xi_{ij,k+1}$. Thus, as a consequence of Assumptions 8-a and 8-b on the noise process and relation (5.1.7), which holds under Assumption 6-b on the weights, we have

$$E[\|\xi_{k+1}^i\|_2^2 | F_k, W_{k+1}] = \sum_{j \neq i} E[\|W_{k+1}^2\|_{ij} \|\xi_{ij,k+1}\|_2^2 | F_k, W_{k+1}] \leq \bar{N} \eta^2 \mu^2, \quad (5.1.8)$$

where $\bar{N} = \max \{|N_i|\}$ and $N_i$ is the set of neighbors of agent $i$ in the graph $G = (V, E)$.

As a consequence of the subgradient norm and subgradient error boundedness (Assumptions 7-e and 8-e), it can be seen that for all $i \in U_{k+1}$ and $k \geq 0$,

$$E[\|d_{k+1}^i + \epsilon_{i,k+1}^i\|_2^2 | F_k, W_{k+1}, v_{k+1}^i] \leq (C + \nu)^2. \quad (5.1.9)$$

Here we have used Hölder’s inequality, which states that for random vectors $\mathbf{x}$ and $\mathbf{y}$ there holds
\[ \mathbb{E} [||x'y||] \leq \sqrt{\mathbb{E} [||x||^2]} \sqrt{\mathbb{E} [||y||^2]} \]. We now show that the two most widely studied communication protocols in the consensus literature satisfy our assumptions on the random matrix \( W_k \).

### 5.1.4 Gossip based communication protocol

In [51], a widely used model for asynchronous gossip based communication protocol is provided, which can be represented as follows. At each tick of the global Poisson clock \( T_p \), an agent represented by the random variable \( I_k \), wakes up with uniform probability. Then, the agent \( I_k \) selects a neighboring agent \( J_k \) with a stationary probability \( P_{I_k,J_k} \). Let us denote by \( W^g_k \) the random matrix \( W_k \) corresponding to this case. Then, the matrix \( W^g_k \) takes the following form:

\[
W^g_k = \frac{1}{2} (e_{I_k} - e_{J_k})(e_{I_k} - e_{J_k})^T \quad \text{with probability } \frac{1}{m} P_{I_k,J_k},
\]

where \( e_j \) is a vector with \( j^{th} \) component equal to 1 and the other components equal to 1. The probability of update for any agent \( i \) at any of the clock ticks is given by

\[
p_i = \frac{1}{m} \left[ 1 + \sum_{j \in N_i} P_{ji} \right].
\]

For the special case when the probability of selecting a neighboring agent is uniform, we have \( p_i = \frac{1}{m} \left[ 1 + \sum_{j \in N_i} \frac{1}{|N_j|} \right] \). Note that, in this case, when the underlying graph \( G = (V, E) \) is regular, the probability of update for all agents is the same and is equal to \( p = \frac{2}{m} \). We also have the following properties for the random matrix \( W^g_k \).

**Lemma 5.1.1.** [51] The random matrix \( W^g_k \) has the following properties:
Figure 5.2: Illustration of the Broadcast Protocol

a) \( \bar{W}_g = \mathbb{E}[W^g_k] \) is a symmetric positive semidefinite matrix such that \( \bar{W}_g \mathbf{1} = 0 \) and \( \mathbf{1}^T \bar{W}_g = 0 \).

b) \( R_g = \mathbb{E}[(W^g_k)^T W^g_k] = \bar{W}_g \).

Proof. We can explicitly compute the expected value of the random matrix \( W(k) \) as follows:

\[
\bar{W}_g = \mathbb{E}[W^g_k] = \frac{1}{2m} \sum_{(i,j) \in E} P_{ij}(e_i - e_j)(e_i - e_j)^T,
\]

which can be represented as \( \bar{W}_g = \frac{1}{2m}(D - (P + P')) \), where \( D \) is a diagonal matrix with entries \( D_i = \sum_{j \neq i} [P_{ij} + P_{ji}] \). Now, it can be verified that \( \bar{W}_g \mathbf{1} = 0 \) and \( \mathbf{1}^T \bar{W}_g = 0 \). Moreover since we have

\[
0 \leq (W^g_k)^T W^g_k = |W^g_k|^2 = W^g_k,
\]

upon taking the expectation, we obtain the desired result.

Observe that our notation is slightly different from that of [51], where the convergence properties of the recursion \( x_{k+1} = \bar{W}_k x_k \) is considered with \( \bar{W}_k = I - W^g_k \).

5.1.5 Broadcast based communication protocol

The consensus algorithm based on a broadcast based communication protocol was discussed in [52]. According to the broadcast protocol once a node \( i \) wakes up at the tick of its local clock, it broadcasts its local variable, which is successfully received by its neighbors. The random matrix \( W_{k+1} \) in this case is labeled \( W^b_k \). Formally, representing the random agent whose clock ticks in slot \( k \) by \( I_k \), the
matrix $W^b_k$ takes the following form:

$$W^b_k = \beta \sum_{j \in N_i} (e_j e_j^T - e_j e_i^T) \quad \text{with probability } 1/m.$$ 

The probability of update for any agent $i$ is given as $p_i = \frac{|N_i|}{m}$. If the graph $G$ is $r$-regular then the probability of update for all agents is uniformly equal to $p = \frac{r}{m}$. Furthermore, the following result holds.

**Lemma 5.1.2.** [56] Under the assumption that the graph $G$ is bidirectional and connected, the random matrix $W^b_k$ has the following properties:

a) $\bar{W}_b = \mathbb{E}[W^b_k]$ is a symmetric positive semidefinite matrix such that $\bar{W}_b 1 = 0$ and $1^T \bar{W}_b = 0$.

b) $R_b = \mathbb{E}[(W^b_k)^T W^b_k] = 2 \beta^2 \bar{W}_b$.

**Proof.** The expected matrix can be computed as

$$\bar{W}_b = \mathbb{E}[W^b_k] = \frac{\beta}{m} \sum_{i \in V} \sum_{j \in N_i} (e_j e_j^T - e_j e_i^T) = \frac{\beta}{m} L,$$

where $L$ is the Laplacian of the graph $G$. By the definition of the graph Laplacian matrix $L$, and the assumptions that the graph $G$ is bidirectional and connected (Assumption 6-a), we have that the matrix $\bar{W}_b$ is a positive semidefinite matrix which satisfies $\bar{W}_b 1 = 0$ and $1^T \bar{W}_b = 0$. It can also be seen that $R_b = \mathbb{E}[(W^b_k)^T W^b_k] = 2 \beta^2 \bar{W}_b$. \hfill $\square$

Recently in [56], a variant of the broadcast algorithm is presented including the possibility of collisions between simultaneous transmissions. In this case it can be seen that, if the graph is $r$-regular then $\bar{W}_{bc} = \beta p(1 - p)^r L$, which is a positive semidefinite matrix with $\bar{W}_{bc} 1 = 0$ and $1^T \bar{W}_{bc} = 0$.

### 5.2 Preliminary Results

In this section we provide various results which will be useful in proving our main results. The following result proven in [55] is helpful in establishing asymptotic error bounds associated with constant step size algorithms.
Lemma 5.2.1. ([55]) Let $\beta \in (0, 1)$, and let $\{d_k\}$ and $\{u_k\}$ be scalar sequences such that

$$d_k \leq \beta d_{k-1} + u_{k-1} \quad \text{for all } k \geq 1.$$  

Then,

$$\limsup_{k \to \infty} d_k \leq \frac{1}{1 - \beta} \limsup_{k \to \infty} u_k.$$

In the current work we restrict ourself to step sizes of the form $\frac{1}{\theta^q}$, where $\frac{1}{2} < \theta \leq 1$. Thus, in our algorithms (5.1.4) and (5.1.5), we will use

$$\alpha^i_k = \frac{1}{\Gamma_{\theta_1}^i}, \quad \gamma^i_k = \frac{1}{\Gamma_{\theta_2}^i},$$  

where $\frac{1}{2} < \theta_1, \theta_2 \leq 1$. In the following lemma, we present certain results regarding these forms of step size sequences.

Lemma 5.2.2. Under the i.i.d assumption on the random network $W(k)$ and $p_i > 0$ (Assumptions 6-c and 6-d), for any step size of the form $\frac{1}{\Gamma_{i,k}^q}$ with $\frac{1}{2} < \theta \leq 1$, the following hold with probability one:

$$\sum_{k=1}^{\infty} \mathbb{E}\left[ \frac{1}{\Gamma_{i,k}^q} \mid F_k \right] < \infty,$$

$$\sum_{k=1}^{\infty} \mathbb{E}\left[ \left| \frac{1}{\Gamma_{i,k}^q} - \frac{1}{p_{i,k}^q} \right| \mid F_k \right] < \infty \quad \text{for all } i \in V.$$

Proof. Note that according to our definition $\Gamma_{i,k} = \sum_{t=1}^{k} \chi_{E_i(t)}$, where $E_i(t) = \{i \in U(t)\}$, and $\chi_{E_i(t)}$ is the indicator function of the event that agent $i$ updates its local variable at instance $t$. Since according to our assumption the random graph sequence $\{W(k)\}$ is assumed to be drawn in an i.i.d fashion, this implies that the events $E_i(t)$ are i.i.d and $\mathbb{E}[\chi_{E_i(t)}] = \mathbb{P}\{i \in U(t)\} = p_i$ for each $i \in V$. By the law of iterated logarithms ([57], pages 476–479), for any $q > 0$, the following holds with probability 1, $\limsup_{k \to \infty} \frac{|\Gamma_{i,k} - kp_i|}{k^{\frac{q}{2} + q}} = 0$ for all $i \in V$. Hence, given any constant $c$ there exists a large enough $\tilde{k}$ with probability one such that $\frac{|\Gamma_{i,k} - kp_i|}{k^{\frac{q}{2} + q}} \leq c$ for all $k > \tilde{k}$. Then it can be
shown [58] that the following bound holds with probability one
\[
\frac{1}{\Gamma_{i,k}} \leq \frac{2}{kp_i} \quad \text{for all } k > \tilde{k}.
\] (5.2.2)

Now, we consider the term \( \left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i k} \right| \). Clearly, we have with probability one
\[
\left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i k} \right| = \frac{1}{\Gamma_{i,k} p_i k} \left| \Gamma_{i,k} - p_i k \right| \leq \frac{2}{p_i^2 k^2} \theta_{k^2} c k^{\frac{1}{2} + \theta}
\]
\[
= \frac{2c}{p_i^2 k^{\frac{3}{2} - \theta}}.
\]

Using the notation \( p = \min_i \{p_i\} \) we have \( \left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i k} \right| \leq \frac{2c}{p_i^{2}k^{\frac{3}{2} - \theta}} \). The above bound was derived for the case of gossip and broadcast communication in [41] and [58] respectively. Now for our general case we need a similar bound on the difference \( \left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i k} \right| \), where \( \theta \in (1/2, 1] \). By applying the mean value theorem to the function \( x^\theta \) we get
\[
\left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i k} \right| \leq \frac{\theta}{x^{1-\theta}} \left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i} \right|
\]
where \( x \) is between \( \frac{1}{\Gamma_{i,k}} \) and \( \frac{1}{p_i k} \). However since both \( \Gamma_{i,k} \leq k \), and \( p_i k \leq k \), we get \( \frac{1}{x} \leq k \). Thus using this as an upper bound and using the earlier bound, we get that the following bound holds with probability one for every \( i \in V \) and \( k \geq \tilde{k} \),
\[
\left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i k} \right| \leq \frac{2\theta c k^{1-\theta}}{k^{\frac{3}{2} + \theta - \theta} p_i^{2}} = \frac{2\theta c}{k^{\frac{3}{2} + \theta - \theta} p_i^{2}}.
\]

Then, clearly we have that with probability one
\[
\mathbb{E}_k \left[ \left| \frac{1}{\Gamma_{i,k}} - \frac{1}{p_i k} \right| \right] \leq \frac{2\theta c k^{1-\theta}}{k^{\frac{3}{2} + \theta - \theta} p_i^{2}} = \frac{2\theta c}{k^{\frac{3}{2} + \theta - \theta} p_i^{2}},
\]

57
for all $k \geq \tilde{k}$. Now summing up we obtain

$$
\sum_{k=0}^{\infty} \mathbb{E}_k \left[ \left| \frac{1}{\Gamma^\theta_{i,k} - \frac{1}{p_i^\theta k^\theta}} \right| \right] = \sum_{k=0}^{\tilde{k}} \mathbb{E}_k \left[ \left| \frac{1}{\Gamma^\theta_{i,k} - \frac{1}{p_i^\theta k^\theta}} \right| \right] + \sum_{k=\tilde{k}+1}^{\infty} \mathbb{E}_k \left[ \left| \frac{1}{\Gamma^\theta_{i,k} - \frac{1}{p_i^\theta k^\theta}} \right| \right] \\
\leq \sum_{k=0}^{\tilde{k}} \mathbb{E}_k \left[ \left| \frac{1}{\Gamma^\theta_{i,k} - \frac{1}{p_i^\theta k^\theta}} \right| \right] + \sum_{k=\tilde{k}+1}^{\infty} \frac{2\theta c}{k^{1+\theta-q} p^2}.
$$

However since $\theta \in (1/2, 1]$, and $q > 0$ is any arbitrary constant, we have $\sum_{k=\tilde{k}+1}^{\infty} \frac{2\theta c}{k^{1+\theta-q} p^2} < \infty$ with probability one. Thus, with probability one, $\sum_{k=0}^{\infty} \mathbb{E}_k \left[ \left| \frac{1}{\Gamma^\theta_{i,k} - \frac{1}{p_i^\theta k^\theta}} \right| \right] < \infty$. A similar argument can be carried out starting from Eq. (5.2.2) to show that with probability one, $\sum_{k=0}^{\infty} \mathbb{E}_k \left[ \frac{1}{\gamma^\theta_{i,k}(k)} \right] < \infty$.

We are now ready to discuss our main results regarding the asymptotic behavior of the algorithms given by Eqs. (5.1.4) and (5.1.5) under diminishing and constant step sizes.

### 5.3 Constrained Consensus

In this section we present our analysis of the constrained consensus algorithm (5.1.4) under both diminishing and constant step sizes. We follow the classical Lyapunov kind of analysis for stochastic systems. Our line of argument is as follows. First in Lemma 5.3.1 we derive a descent type relation involving a particular choice of a Lyapunov function. Then in Theorem 5.3.2 we show that under our various assumptions we can suitably apply the Lemma 2.3.2 to argue almost sure convergence of the local variables $x_i^k$ as generated by the algorithm (5.1.4) to a common random point in the intersection of the local constraint sets. Following this we consider the constant step size case in Theorem 5.3.3. It is well known that in this case we cannot achieve almost sure convergence of the algorithm. Here we derive asymptotic error bounds based on a metric which penalizes deviations from the consensus subspace.

We now provide some basic relations which will be valuable in deriving both the almost sure
convergence and the error bound for a constant step size. Let us introduce the notation for the joint state vector as $x_k = (x^T_{1,k}, \ldots, x^T_{m,k})^T$, where $x^i_k$ is the local variable at node $i$ constrained to the set $X_i$. Correspondingly, we denote the vector $v_k = (v^T_{1,k}, \ldots, v^T_{m,k})^T$, where $v^i_k$ is given by Eq. (5.1.4). For a fixed vector $z \in \mathbb{R}^n$, we denote $z \in \mathbb{R}^{mn}$ as the vector consisting of $m$ copies of the vector $z$, i.e., $z = 1_m \otimes z$. By definition any such vector $z$ belongs to the consensus subspace $C \subset \mathbb{R}^{mn}$. To get further insight in the problem, let us define the product constraint set $\tilde{X} = X_1 \times \cdots \times X_m$. Then, clearly $\tilde{X} \subset \mathbb{R}^{mn}$. According to this definition the algorithm given by Eq. (5.1.4) implies that $x_k \in \tilde{X}$ for all $k$. The constrained consensus problem can be equivalently thought of as the task of finding a point in the intersection set $\tilde{X} \cap C$. We introduce the diagonal matrix $Q = D(p^{\theta_1})$, where $\theta_1 \in (1/2, 1]$ and the matrix $Q = Q \otimes I_n$. Also, we define the diagonal matrix $\Lambda_k$, as $\Lambda_k = D(\alpha^k) \otimes I_n$. The matrix $W_k$ is defined as $W_k = W_k \otimes I_n$. Finally, we define the net noise vector by $\xi^T_k = (\xi^T_{1,k}, \ldots, \xi^T_{m,k})^T$. According to this notation and the algorithm in Eq. (5.1.4), we can write

$$v_{k+1} = [I - \tilde{\Lambda}_{k+1} W_{k+1}] x_k - \tilde{\Lambda}_{k+1} \xi_{k+1}. \quad (5.3.1)$$

For brevity of notation let us denote the conditional expectation

$$\mathbb{E} [\cdot | F_k] = \mathbb{E}_k [\cdot].$$

Then, we have the following result.

**Lemma 5.3.1.** Consider a Lyapunov function of the form $V_k = (x_k - z^*)^T Q (x_k - z^*)$, where $z^* = 1_m \otimes z^*$ for any point $z^* \in X$. Then, for the algorithm (5.1.4) under Assumption 7-a on the constraint sets and the zero mean assumption on the communication noise (Assumption 8-a), we have for all $k \geq 0$,

$$\mathbb{E}_k [V_{k+1}] \leq V_k + (x_k - z^*)^T \mathbb{E}_k [W^T_{k+1} \tilde{\Lambda}^2_{k+1} Q W_{k+1}] (x_k - z^*) - 2(x_k - z^*)^T \mathbb{E}_k [Q \tilde{\Lambda}_{k+1} W_{k+1}] (x_k - z^*) + \mathbb{E}_k [\xi^T_{k+1} \tilde{\Lambda}_{k+1} Q \xi_{k+1}].$$

**Proof.** Using the fact that $W_{k+1} z^* = 0$ which holds by the definition of $W_k$ in Eq. (5.1.6), from
relation (5.3.1) we have

\[
\mathbb{E}_k[(v_{k+1} - z^*)^T Q(v_{k+1} - z^*]) = V(x_k) + (x_k - z^*)^T \mathbb{E}_k \left[ W_{k+1}^{T} \hat{\Lambda}_{k+1}^{2} Q W_{k+1} \right] (x_k - z^*)
\]

\[
- 2(x_k - z^*)^T \mathbb{E}_k \left[ Q \hat{\Lambda}_{k+1} W_{k+1} \right] (x_k - z^*) + \mathbb{E}_k \left[ \xi_{k+1}^{T} \hat{\Lambda}_{k+1}^{2} Q \xi_{k+1} \right]
\]

\[
- 2\mathbb{E}_k \left[ ((I - \hat{\Lambda}_{k+1} W_{k+1})x_k - z^*)^T Q \hat{\Lambda}_{k+1} \xi_{k+1} \right].
\] (5.3.2)

Using the iterated expectation rule and \(\mathbb{E}[\xi_{k+1} | F_k, W_{k+1}] = 0\) (Assumption 8-a), we have

\[
\mathbb{E}_k \left[ (I - \hat{\Lambda}_{k+1} W_{k+1})x_k - z^*)^T Q \hat{\Lambda}_{k+1} \xi_{k+1} \right]
\]

\[
= \mathbb{E}_k \left[ (x_k - \hat{\Lambda}_{k+1} W_{k+1} x_k - z^*)^T Q \hat{\Lambda}_{k+1} \mathbb{E}[\xi_{k+1} | F_k, W_{k+1}] \right]
\]

\[
= 0.
\]

By the non-expansiveness property of the projection operator (cf. (2.3.1)), we obtain

\[
p_i^\theta_1 \| x^i_{k+1} - z^* \|^2 = p_i^\theta_1 \| P_{X_i} [v^i_{k+1}] - P_{X_i} [z^*] \|^2 \leq p_i^\theta_1 \| v^i_{k+1} - z^* \|^2.
\]

On summing this over index \(i \in V\), we get

\[
V_{k+1} = (x_{k+1} - z^*)^T Q(x_{k+1} - z^*) \leq (v_{k+1} - z^*)^T Q(v_{k+1} - z^*).
\]

The desired relation follows by taking the expectation conditioned on \(F_k\).

\[
\Box
\]

5.3.1 Convergence Result

The following theorem brings together our various assumptions and the preliminary results to provide conditions guaranteeing almost sure convergence of the algorithm.

**Theorem 5.3.2.** Let Assumption 6 on the network hold. Also, let Assumptions 7-a and 7-b on the constraint sets, and Assumptions 8-a, 8-b, and 8-c on the noise process hold. Let the step size \(\alpha_k^i\) be of the form \(\alpha_k^i = \frac{1}{\Gamma_{i,k} \theta_1}\), where \(\frac{1}{2} < \theta_1 \leq 1\). Then, the iterations of the algorithm given by
Eq. (5.1.4) satisfy with probability one for all $i$ and some random vector $z^* \in X$:

$$\lim_{k \to \infty} \| x^i_k - z^* \| = 0.$$  

Proof. The starting point of our analysis for almost sure convergence is the relation derived in Lemma 5.3.1. Let us focus on the term $-2(x_k - z^*)^T \mathbb{E}_k \left[ Q \tilde{\Lambda}_{k+1} W_{k+1} \right] (x_k - z^*)$, where $z^* = 1_m \otimes z^*$ for an arbitrary $z^* \in X$. According to our notation, we have $Q \tilde{\Lambda}_{k+1} = Q \Lambda_{k+1} \otimes I_n$, where $Q \Lambda_{k+1}$ is a diagonal matrix with diagonal entries given by $p^{\theta_1}_{i,k+1}$. Thus, we can write

$$p^{\theta_1}_{i,k+1} = \frac{p^{\theta_1}_{i,k}}{\Gamma_{i,k+1}^{\theta_1}} + p^{\theta_1}_{i,k+1} \left( \frac{1}{\Gamma_{i,k+1}^{\theta_1}} - \frac{1}{p^{\theta_1}_{i,k}(k+1)^{\theta_1}} \right).$$

Let us denote $H_{k,\theta_1} = D \left( \frac{1}{\Gamma_{i,k}^{\theta_1}} - \frac{1}{p^{\theta_1}_{i,k} \theta_1} \right)$ and $\tilde{H}_{k,\theta_1} = H_{k,\theta_1} \otimes I_n$. Then, we have

$$Q \tilde{\Lambda}_{k+1} W_{k+1} = \frac{1}{(k+1)^{\theta_1}} W_{k+1} + Q \tilde{H}_{k+1,\theta_1} W_{k+1}.$$

Since $W 1_m = 0$ (Assumption 6-c), it follows that

$$-2(x_k - z^*)^T \mathbb{E}_k \left[ Q \tilde{\Lambda}_{k+1} W_{k+1} \right] (x_k - z^*) = -2x_k^T \tilde{W} x_k \frac{1}{(k+1)^{\theta_1}}$$

$$- 2(x_k - z^*)^T \mathbb{E}_k \left[ Q \tilde{H}_{k+1,\theta_1} W_{k+1} \right] (x_k - z^*).$$

(5.3.3)

Now, using $p^{\theta_1}_i \leq 1$, $|[W_{k+1}]_{ij}| \leq \eta$ for $i \neq j$ and $|[W_{k+1}]_{ii}| \leq |N_i| \eta$ (cf. Eq. (5.1.6) as implied by Assumption 6-b), the boundedness of constraint sets (Assumption 7-b), we can upper bound the second term above as follows:

$$-2(x_k - z^*)^T \mathbb{E}_k \left[ Q \tilde{H}_{k+1,\theta_1} W_{k+1} \right] (x_k - z^*)$$

$$\leq 2 \sum_{E} \left| (x^i_k - z^*)^T (x^j_k - z^*) \right| \mathbb{E}_k \left[ |(W_{k+1})_{ij}| \right] |(H_{k+1,\theta_1})_{ij}|$$

$$\leq 2\eta \bar{N} \sum_{E} C_{X_i} X_{X_i} \mathbb{E}_k \left[ |(H_{k+1,\theta_1})_{ij}| \right],$$

(5.3.4)

where $C_{X_i}$ is the diameter of the set $X_i$ (i.e., $C_{X_i} = \max_{x,y \in X_i} \| x - y \|$) and $\bar{N}$ is the maximal node
degree in the graph $G$ (i.e., $\bar{N} = \max_{i} |N_i|$). Under zero mean, independent across the links, and bounded second moment assumptions on the link noise (Assumptions 8-a, 8-b, and 8-c), the term involving communication noise in Lemma 5.3.1 can be bound as follows:

$$
\mathbb{E}_k [\xi_{k+1}^T \bar{\Lambda}_{k+1}^2 Q \xi_{k+1}] = \sum_{i \in V} \mathbb{E}_k \left[ p_i^\theta_i (\alpha_{k+1})^2 \| \xi_{k+1} \|^2 \right] = \sum_{i \in V} \mathbb{E}_k \left[ p_i^\theta_i (\alpha_{k+1})^2 \mathbb{E}[\| \xi_{k+1} \|^2 | F_k, W_{k+1}] \right] \\
\leq \bar{N} \eta^2 \mu^2 \sum_{i \in V} \mathbb{E}_k \left[ (\alpha_{k+1})^2 \right] . \tag{5.3.5}
$$

Here, we have used the inequality in Eq. (5.1.8). Similarly we can derive the following bound

$$
(x_k - z^*)^T \mathbb{E}_k \left[ W_{k+1}^T \bar{\Lambda}_{k+1}^2 Q W_{k+1} \right] (x_k - z^*) = \sum_{E} (x_{k,j}^T - z^*)^T (x_{j,k} - z^*) \\
\times \mathbb{E} \left[ \sum_{i} [W_{k+1}]_{ij} [W_{k+1}]_{ij} p_i^\theta_i (\alpha_{k+1})^2 \right] \\
\leq \bar{N}^2 \eta^2 \sum_{E} C_{G_{ij}} \sum_{i \in V} \mathbb{E} \left[ (\alpha_{k+1})^2 \right] . \tag{5.3.6}
$$

Now substituting the bounds from Eqs. (5.3.3), (5.3.4), (5.3.5) and (5.3.6) in the relation of Lemma 5.3.1, we obtain

$$
\mathbb{E}_k [V_{k+1}] \leq V_k - \frac{2x_k^T \bar{W} x_k}{(k+1)^{\theta_1}} + 2 \bar{N} \eta \sum_{E} C_{G_{ij}} C_{G_{i,j'}} \mathbb{E}_k \left[ |H_{k+1,\theta_1}| \right] \\
+ \left( \bar{N} \eta^2 \mu^2 + \bar{N}^2 \eta^2 \sum_{E} C_{G_{ij}} C_{G_{i,j'}} \right) \sum_{i \in V} \mathbb{E}_k \left[ (\alpha_{k+1})^2 \right] .
$$

Note that from Lemma 5.2.2, we have the result that for the choice of step size $\alpha_{k+1} = \frac{1}{(k+1)^{\theta_1}}$ the last two terms are summable with probability one. Thus, we can apply the supermartingale convergence result of Lemma 2.3.2 to deduce that, with probability one, both the sequence $V_k = (x_k - z^*)^T Q (x_k - z^*)$ converges for any $z^* \in X$ and the following holds:

$$
\sum_{k=1}^{\infty} \frac{1}{(k+1)^{\theta_1}} x_k^T \bar{W} x_k < \infty . \tag{5.3.7}
$$

Since $\theta_1 \in (1/2, 1]$, we have $\sum_{k=1}^{\infty} \frac{1}{(k+1)^{\theta_1}} = \infty$. Moreover, since $\bar{W}$ is positive semidefinite (Assumption 6-c), from Eq. (5.3.7) it follows that with probability one $\liminf_{k \to \infty} x_k^T \bar{W} x_k = 0$. Thus,
with probability one, there exists a subsequence such that
\[
\lim_{k \to \infty} x_{n_k}^T \bar{W} x_{n_k} = 0.
\]

Now, under Assumption 6-c, the consensus subspace \( \mathcal{C} \) is the null space of the matrix \( \bar{W} \), hence for all agents \( i \) and \( j \) we have with probability one
\[
\lim_{k \to \infty} \| x_{n_k}^i - x_{n_k}^j \| = 0.
\]

Since the constraint sets are compact (Assumption 7-b), we can further pick a convergent subsequence such that \( \lim_{k \to \infty} \| x_{n_k}^i - \tilde{z}_i \| = 0 \) with probability one for all \( i \) and some random vector \( \tilde{z}_i \in X_i \). Since \( \lim_{k \to \infty} \| x_{n_k}^i - x_{n_k}^j \| = 0 \) almost surely, implying that \( \tilde{z}_1 = \ldots = \tilde{z}_m = \tilde{z}^* \). Hence, we can deduce that \( \lim_{k \to \infty} V_{n_k} = \lim_{k \to \infty} \sum_i p_i \theta_i \| x_{n_k}^i - \tilde{z}^* \|^2 = 0 \). But, \( V_k \) converges almost surely for any \( \tilde{z}^* \in X \), hence the sub-sequential limit is also the sequential limit. This implies that almost surely
\[
\lim_{k \to \infty} V_k = \lim_{k \to \infty} \sum_i p_i \theta_i \| x_k^i - \tilde{z}^* \|^2 = 0.
\]
Since \( p_i > 0 \), we have \( \lim_{k \to \infty} \| x_k^i - \tilde{z}^* \| = 0 \) almost surely for all \( i \) and a random point \( \tilde{z}^* \in X \). \( \Box \)

5.3.2 Constant step size error bound

In this section we focus on a constant step size algorithm, where \( \alpha_k^i = \alpha_i \) for all \( i \) and \( k > 0 \). Using constant step sizes does not give us almost sure convergence to the consensus subspace, but in this case we can provide an asymptotic error bound on the iterations of algorithm (5.1.4).

**Theorem 5.3.3.** Let Assumption 6 on the network hold. Also, let Assumptions 7-a and 7-b on the constraint set hold together with Assumptions 8-a, 8-b and 8-c on the link noise. Then, for the iterates generated by algorithm (5.1.4) we have the following asymptotic bound:

\[
\limsup_{T \to \infty} \mathbb{E} \left[ (\hat{x}_T - z^*)^T \bar{W} (\hat{x}_T - z^*) \right] \leq \frac{\eta \Delta \alpha \bar{N}}{\alpha} \sum_{i} C_{X_i} C_{X_i} + \frac{\bar{\alpha}^2}{2\alpha} \left[ m \bar{n} \eta^2 \mu^2 + \| \bar{R} \| \sum_{i=1}^m C_{X_i}^2 \right],
\]

where \( \hat{x}_T = \frac{1}{T} \sum_{k=0}^{T-1} x_k, \ R = \mathbb{E} \left[ W_{k+1}^T W_{k+1} \right], \ |R| \) is the norm of \( R \) induced by the Euclidean
vector norm, and \( z^* = 1_m \otimes z^* \) for any point \( z^* \in X \).

**Proof.** Proceeding as in Lemma 5.3.1 and in the bound of Eq. (5.3.5), we have that for constant step sizes \( \alpha_i \), the following relation holds:

\[
\mathbb{E}_k \left[ \|x_{k+1} - z^*\|^2 \right] \leq \|x_k - z^*\|^2 + \bar{\alpha}^2 (x_k - z^*)^T R(x_k - z^*) - 2(x_k - z^*)^T \tilde{\Lambda} \bar{W}(x_k - z^*) + m \bar{N} \eta^2 \mu^2 \bar{\alpha}^2. \tag{5.3.8}
\]

Using the definition of \( \tilde{\Lambda} \), we can derive the following estimate:

\[
-2(x_k - z^*)^T \tilde{\Lambda} \bar{W}(x_k - z^*) \leq -2\bar{\alpha}(x_k - z^*)^T \bar{W}(x_k - z^*) + 2\eta \Delta \bar{N} \sum_E C_X_i C_X_j, \tag{5.3.9}
\]

By substituting (5.3.9) in (5.3.8) and taking total expectation, we obtain

\[
2\bar{\alpha} \mathbb{E} \left[ (x_k - z^*)^T \bar{W}(x_k - z^*) \right] \leq \mathbb{E} \left[ \|x_k - z^*\|^2 \right] - \mathbb{E} \left[ \|x_{k+1} - z^*\|^2 \right] + 2\eta \Delta \bar{N} \sum_E C_X_i C_X_j + m \bar{N} \eta^2 \mu^2 \bar{\alpha}^2 + \bar{\alpha}^2 \mathbb{E} \left[ (x_k - z^*)^T R(x_k - z^*) \right]. \tag{5.3.10}
\]

Using the compactness condition (Assumption 7-b), we can bound \((x_k - z^*)^T R(x_k - z^*) \leq \|R\| \sum_{i=1}^m C_{X_i}^2\).

Using this bound and summing relations (5.3.10) from \( k = 0 \) to \( k = T - 1 \) for some \( T > 0 \), we have

\[
\frac{1}{T} \sum_{k=0}^{T-1} \mathbb{E} \left[ (x_k - z^*)^T \bar{W}(x_k - z^*) \right] \leq \frac{1}{2\bar{\alpha} T} \left[ \mathbb{E} \left[ \|x_0 - z^*\|^2 \right] - \mathbb{E} \left[ \|x_T - z^*\|^2 \right] \right] + \frac{\eta \Delta \bar{N}}{\bar{\alpha}} \sum_E C_X_i C_X_j + \frac{\bar{\alpha}^2}{2\bar{\alpha}} \left[ m \bar{N} \eta^2 \mu^2 + \|R\| \sum_{i=1}^m C_{X_i}^2 \right].
\]

Letting \( \bar{x}_T = \frac{1}{T} \sum_{k=0}^{T-1} x_k \), and using Jensen’s inequality and the convexity of the function \( g(y) = \)
we obtain
\[
E \left[ (\hat{x}_T - z^*)^T \bar{W} (\hat{x}_T - z^*) \right] \leq \frac{1}{2\alpha T} \left[ E \left[ \|x_0 - z^*\|^2 \right] - E \left[ \|x_T - z^*\|^2 \right] \right] + \frac{\eta \Delta_\alpha \tilde{N}}{\alpha} \sum_{E} C_{X_i} C_{X_j} + \frac{\tilde{\alpha}^2}{2\alpha} \left[ m \tilde{N} \eta^2 \mu^2 + \|R\| \sum_{i=1}^{m} C_{X_i}^2 \right].
\]

The result follows by taking the limsup as \( T \to \infty \) and noting that, by the compactness of the sets \( X_i \) (Assumption 7-b), the limsup of the first term on the right-hand side converges to 0, as \( T \to \infty \).

In the error bound of Theorem 5.3.3 the first error term, namely \( \frac{\eta \Delta_\alpha \tilde{N}}{\alpha} \sum_{E} C_{X_i} C_{X_j} \), is due to misaligned step sizes. In other words, this term would be zero if all step sizes \( \alpha_i \) take the same value (\( \Delta_\alpha = 0 \)). The second term in the error, \( \frac{\tilde{\alpha}^2}{2\alpha} m \tilde{N} \eta^2 \mu^2 \), is the contribution of the noise in the communication links, which would not be present if the links are perfect (\( \mu = 0 \)). Note that this term is of the order \( \frac{\tilde{\alpha}^2}{2\alpha} \) and, hence, diminishes with diminishing step sizes. The third term in the error \( \frac{\alpha^2}{2\alpha} \|R\| \sum_{i=1}^{m} C_{X_i}^2 \) can also be controlled by controlling the step sizes. In the case of gossip and broadcast protocols, we have \( R_g = \tilde{W}_g \) and \( R_b = 2\beta^2 \tilde{W}_b \), thus the last error term can be written in terms of the expected matrices \( \tilde{W}_g \) and \( \tilde{W}_b \), respectively.

## 5.4 Distributed Optimization

In this section we discuss the asymptotic properties of the algorithm given in Eq. (5.1.5) using both diminishing and constant step sizes. As mentioned earlier, a novel aspect of the proposed algorithm is the use of two step size sequences to damp communication noise and subgradient errors. As in the problem of constrained consensus, for the case of diminishing step sizes we can show almost sure convergence of the iterates to the optimal solution set \( X^* \subseteq X \). An interesting feature which emerges out of our analysis is the requirement that, for almost sure convergence, the step size corresponding to the subgradient error needs to decay to zero at a faster rate than the step size sequence corresponding to the communication noise.

Once again we use tools from Lyapunov analysis to study the convergence properties of the proposed algorithm. The crucial bound in Lemma 4.5.1 will prove to be useful here too. To derive this bound
we need to impose that the intersection of the constraint sets have a nonempty interior as stated in Assumption 7-c. This was also required to prove convergence of the alternating projection method to find the intersection of convex sets in [32]. It was also used in [11] to establish convergence of a distributed algorithm and the rate of convergence guarantees. We then proceed to derive a similar descent type inequality to Lemma 5.3.1 in two steps. First we derive an auxiliary bound in Lemma 5.4.1 which holds for diminishing stepsizes $\alpha_i^k$ and $\gamma_i^k$, then in Lemma 5.4.2 we refine the result of Lemma 5.4.1 for a specific choice of the form of the stepsizes to get the desired descent type inequality. We then proceed to argue almost sure convergence of the algorithm in Theorem 5.4.3 and Theorem 5.4.4. These theorems differ in their set of assumptions on the network and the step sizes employed by the agents. We then establish an asymptotic error bound for the algorithm in Theorem 5.4.5, when both the step sizes corresponding to the subgradient step and the alignment task are constant. In this case, under strong convexity assumption on the objective functions $f_i(x)$, we obtain a contractive property which enables us to apply Lemma 5.2.1 to derive the result.

### 5.4.1 Preliminary results

In this section we provide several results which we use when deriving our main result. We now provide a basic iterate relation for algorithm (5.1.5), which we use later on to establish convergence and error bounds.

**Lemma 5.4.1.** Let Assumption 6 on the network hold. Also, let Assumption 7 on the constraint sets and objective functions hold as well as Assumption 8 on the link noise and subgradient errors. Further, let the step size $\alpha_k$ be such that $\lim_{k \to \infty} \alpha_k = 0$. Then, for the iterates generated by algorithm (5.1.5) the following inequality holds for any $z^* \in X^*$ and for all large enough $k$,

$$
\mathbb{E} [\|x_{k+1} - z^*\|^2 \mid F_k, W_{k+1}] \leq \mathbb{E} [\|v_{k+1} - z^*\|^2 \mid F_k, W_{k+1}] + 2(C + \nu)^2 \sum_{i \in U_{k+1}} (\gamma_i^k)^2 \\
- 2 \sum_{i \in U_{k+1}} \Gamma_i \sum_{j=1}^m \sum_{r=1}^m \|x_{r}^j - x_{r}^k\|,
$$

where $z^* = \mathbf{1}_m \otimes z^*$, $s_k = \frac{1}{m} \sum_{i=1}^m P_X [x_i^k]$ and $K = \frac{mB+\delta}{m\delta}$.

**Proof.** By definition we have $x_{k+1} = P_{X_i} [v_{k+1} - \Gamma_{i,k+1}d_{k+1}^i \chi_{\{i \in U_{k+1}\}}]$. We see that for any $z^* \in$
\( X^* \subseteq X \) and all \( i \in V \),

\[
\|x^i_{k+1} - z^*\|^2 = \left\| P_{X^*}[v^i_{k+1} - \Gamma_{i,k+1}\tilde{d}^i_{k+1} \chi_{i \in U_{k+1}}] - P_{X^*}[z^*] \right\|^2 \\
\leq \left\| v^i_{k+1} - z^* - \Gamma_{i,k+1}\tilde{d}^i_{k+1} \chi_{i \in U_{k+1}} \right\|^2 \\
= \|v^i_{k+1} - z^*\|^2 + \left( (\gamma_{k+1}^i)^2 \left\| \tilde{d}^i_{k+1} \right\|^2 - 2\Gamma_{i,k+1}\tilde{d}^i_{k+1}(v^i_{k+1} - z^*) \right) \chi_{i \in U_{k+1}}.
\]

Taking conditional expectation with respect to the past information \( F_k \) and the matrix \( W_{k+1} \), we obtain for any \( z^* \in X^* \),

\[
\mathbb{E}[\|x^i_{k+1} - z^*\|^2 \mid F_k, W_{k+1}] \leq \mathbb{E}[\|v^i_{k+1} - z^*\|^2 \mid F_k, W_{k+1}] \\
+ 2(\gamma_{k+1}^i)^2 \mathbb{E}\left[ \left\| \tilde{d}^i_{k+1} \right\|^2 \mid F_k, W_{k+1} \right] \chi_{i \in U_{k+1}} \\
- 2\Gamma_{i,k+1}\mathbb{E}[d'_{i,k+1}(v^i_{k+1} - z^*) \mid F_k, W_{k+1}] \chi_{i \in U_{k+1}} \\
- 2\Gamma_{i,k+1}\mathbb{E}[\xi'_{i,k+1}(v^i_{k+1} - z^*) \mid F_k, W_{k+1}] \chi_{i \in U_{k+1}}. \quad (5.4.1)
\]

By Assumption 8-d we have \( \mathbb{E}[^i_{k+1} \mid F_k, W_{k+1}, v^i_{k+1}] = 0 \), thus we get

\[
\mathbb{E}[\xi'_{i,k+1}(v^i_{k+1} - z^*) \mid F_k, W_{k+1}] = \mathbb{E}[(v^i_{k+1} - z^*)^i \mathbb{E}[\xi'_{k+1} \mid F_k, W_{k+1}, v^i_{k+1}] \mid F_k, W_{k+1}] = 0.
\]

Also, using the fact that \( d'_{k+1} \) is a subgradient of \( f_i \) at \( v^i_{k+1} \) and the bound on the subgradient error of Eq. (5.1.9) (implied by Assumptions 7-e and 8-e), we have

\[
\mathbb{E}[\|x^i_{k+1} - z^*\|^2 \mid F_k, W_{k+1}] \leq \mathbb{E}[\|v^i_{k+1} - z^*\|^2 \mid F_k, W_{k+1}] + 2(\gamma_{k+1}^i)^2(C + \nu)^2 \chi_{i \in U_{k+1}} \\
- 2\Gamma_{i,k+1}\mathbb{E}[f_i(v^i_{k+1}) - f_i(z^*) \mid F_k, W_{k+1}] \chi_{i \in U_{k+1}}.
\]

Since \( f_i \) is a convex function, by Jensen’s inequality we have

\[-\mathbb{E}[f_i(v^i_{k+1}) \mid F_k, W_{k+1}] \leq -f(\mathbb{E}[v^i_{k+1} \mid F_k, W_{k+1}]).
\]

By the definition of \( v^i_{k+1} \) in Eq. (5.1.5) and \( \mathbb{E}[^i_{ij,k} \mid F_k, W_{k+1}] = 0 \) (Assumption 8-a), we have for
\[ i \in U_{k+1}, \]
\[ \mathbb{E}[v^i_{k+1} \mid F_k, W_{k+1}] = x^i_k - \alpha^i_{k+1} \sum_{j=1}^{m} [W_{k+1}]_{ij} x^j_k. \]

Letting \( y_{i,k} = x^i_k - \alpha^i_{k+1} \sum_{j=1}^{m} [W_{k+1}]_{ij} x^j_k \), we obtain
\[
\mathbb{E}[\|v^i_{k+1} - z^*\|_2^2 \mid F_k, W_{k+1}] \leq \mathbb{E}[\|v^i_{k+1} - z^*\|_2^2 \mid F_k, W_{k+1}] + 2(\gamma_{k+1}^i)^2 (C + \nu)^2 \chi_{\{i \in U_{k+1}\}}
- 2\Gamma_{i,k+1}[f_i(y_{i,k}) - f_i(z^*)] \chi_{\{i \in U_{k+1}\}}. 
\] (5.4.2)

Summing over all \( i \) and using vector notation yields
\[
\mathbb{E}[\|x_{k+1} - z^*\|_2^2 \mid F_k, W_{k+1}] \leq \mathbb{E}[\|v_{k+1} - z^*\|_2^2 \mid F_k, W_{k+1}] + 2(C + \nu)^2 \sum_{i \in U_{k+1}} (\gamma_{k+1}^i)^2 
- 2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1}[f_i(y_{i,k}) - f_i(z^*)] \chi_{\{i \in U_{k+1}\}}. 
\] (5.4.2)

Notice that \( s_k = \frac{1}{m} \sum_{i=1}^{m} P_X[x^i_k] \). Upon adding and subtracting the term \( \sum_{i \in U_{k+1}} f_i(s_k) \) in Eq. (5.4.2) we get
\[
\mathbb{E}[\|x_{k+1} - z^*\|_2^2 \mid F_k, W_{k+1}] \leq \mathbb{E}[\|v_{k+1} - z^*\|_2^2 \mid F_k, W_{k+1}] + 2(C + \nu)^2 \sum_{i \in U_{k+1}} \gamma_{k+1}^i (k + 1) 
- 2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1}[f_i(s_k) - f_i(z^*)] - 2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1}[f_i(y_{i,k}) - f_i(s_k)]. 
\] (5.4.3)

Let us now focus on the last term in Eq. (5.4.3), which we can bound as follows:
\[
- 2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1}[f_i(y_{i,k}) - f_i(s_k)] \leq 2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1} |f_i(y_{i,k}) - f_i(s_k)|. 
\]

By the convexity and subgradient boundedness of each \( f_i(x) \), we have
\[
|f_i(y_{i,k}) - f_i(s_k)| \leq C \|y_{i,k} - s_k\|. 
\] (5.4.4)
We now derive an upper bound on the term $\|y_{i,k} - s_k\|$. Since $\alpha_k \to 0$ and 
\[ y_{i,k} = x_k^i - \alpha_{k+1}^i \sum_{j=1}^m [W_{k+1}]_{ij} x_k^j, \]
the matrix $I_m - \alpha_{k+1}^i W_{k+1}$ is a stochastic matrix for all $k$ large enough. Therefore, it follows by the convexity of the norm function and the fact that $0 \leq 1 - \alpha_{k+1}^i [W_{k+1}]_{ij} \leq 1$ for all $i, j$ and all $k$ large enough
\[
\|y_{i,k} - s_k\| \leq \sum_{j=1}^m \|x_k^j - s_k\|. \tag{5.4.5}
\]

Next, let us consider $\|x_k^i - s_k\|$. Since $s_k = \frac{1}{m} \sum_{j=1}^m P_X[x_k^j]$, by adding and subtracting the term $P_X[x_k^i]$ inside the norm and using the convexity of norm function, we have
\[
\|x_k^i - s_k\| = \left\| x_k^i - \frac{1}{m} \sum_{j=1}^m P_X[x_k^j] \right\| \leq \|x_k^i - P_X[x_k^i]\| + \frac{1}{m} \sum_{j=1}^m \|P_X[x_k^i] - P_X[x_k^j]\|.
\]

By Lemma 4.5.1 we have that $\|x_k^i - P_X[x_k^i]\| \leq \frac{B}{m} \sum_{j=1}^m \|x_k^i - x_k^j\|$, and by the non-expansiveness property of projection $\|P_X[x_k^i] - P_X[x_k^j]\| \leq \|x_k^i - x_k^j\|$ (Lemma (2.3.1)). Hence, for all $i$,
\[
\|x_k^i - s_k\| \leq \frac{mB + \delta}{m\delta} \sum_{j=1}^m \|x_k^j - x_k^i\|. \tag{5.4.6}
\]
Combining relations in Eqs. (5.4.4), (5.4.5) and (5.4.6), we obtain for $k$ large enough,
\[
|f_i(y_{i,k}) - f_i(s_k)| \leq C \frac{mB + \delta}{m\delta} \sum_{j=1}^m \sum_{r=1}^m \|x_k^j - x_k^r\|.
\]

Therefore, for all $k$ large enough, we have
\[
-2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1}[f_i(y_{i,k+1}) - f_i(s_k)] \leq 2C \frac{mB + \delta}{m\delta} \sum_{i \in U_{k+1}} \Gamma_{i,k+1} \sum_{j=1}^m \sum_{r=1}^m \|x_k^j - x_k^r\|.
\]
Letting $K = \frac{mB + \delta}{m\delta}$ and substituting the preceding relation in Eq. (5.4.3), we obtain the desired result. □
Lemma 5.4.1 is true for any step size sequence $\gamma^i_k$. We next give a corollary of the lemma for the special case when $\alpha^i_k = \frac{1}{\Gamma_{1,k}^i}$ and $\gamma^i_k = \frac{1}{\Gamma_{2,k}^i}$ with $\frac{1}{2} < \theta_1, \theta_2 \leq 1$. Let us define $P = P \otimes I_n$, where $P$ is defined by $P = D \left( \frac{1}{p_i^1} \right)$.

**Lemma 5.4.2.** In addition to the assumptions in Lemma 5.4.1, let the step sizes be given by $\alpha^i_k = \frac{1}{\Gamma_{1,k}^i}$ and $\gamma^i_k = \frac{1}{\Gamma_{2,k}^i}$ with $\frac{1}{2} < \theta_1, \theta_2 \leq 1$. Then, the following relation holds for all large enough $k$, any $z^* \in X^*$, and $z^* = 1_m \otimes z^*$,

\[
E_k \left[ \|x_{k+1} - z^*\|^2 \right] \leq \|x_k - z^*\|^2 - \frac{2x_k^T PWx_k}{(k + 1)^\theta_1} + \left( \tilde{N}^2 \mu^2 + \tilde{N}^2 \eta^2 \sum_{i \in V} C_{X_i} C_{X_i'} \right) \sum_{i \in V} E_k \left[ (\alpha^i_{k+1})^2 \right] + 2\eta \tilde{N} \sum_{i \in V} C_{X_i} C_{X_i'} E_k \left[ \|H_{k+1,\theta_1}i\| \right] + 2(C + \nu)^2 E_k \left[ \sum_{i \in U_{k+1}} (\gamma^i_{k+1})^2 \right] - \frac{2}{(k + 1)^{\theta_2}} \sum_{i=1}^m p_i^{1-\theta_2} [f_i(s_k) - f_i(z^*)] + \frac{2CK}{(k + 1)^{\theta_2}} \sum_{j=1}^m \sum_{r=1}^m \|x_j^k - x_r^k\| \sum_{i=1}^m p_i^{1-\theta_2} + 2C \left( \|s_k - z^*\| + K \sum_{j=1}^m \sum_{r=1}^m \|x_j^k - x_r^k\| \right) E_k \left[ \sum_{i \in U_{k+1}} \|H_{k+1,\theta_2}i\| \right].
\]

**Proof.** Consider the result of Lemma 5.4.1, and focus on the term $E_k \left[ \|v_{k+1} - z^*\|^2 \right]$. Note that we can proceed as in Eq. (5.3.2) to get

\[E_k \left[ \|v_{k+1} - z^*\|^2 \right] = \|x_k - z^*\|^2 + (x_k - z^*)^T E_k \left[ W_{k+1}^T \tilde{A}_{k+1}^2 W_{k+1} \right] (x_k - z^*) - 2(x_k - z^*)^T E_k \left[ \tilde{A}_{k+1} W_{k+1} \right] (x_k - z^*) + \tilde{E}_k \left[ \xi_{k+1} \tilde{A}_{k+1}^2 \xi_{k+1} \right],
\]

where we have used the assumption that noise is zero mean. Proceeding similarly as in Eqs. (5.3.3) and (5.3.4) we can write

\[
\tilde{A}_{k+1} W_{k+1} = \frac{1}{(k + 1)^{\theta_1}} PW_{k+1} + \tilde{H}_{k+1,\theta_1} W_{k+1}
\]
to get the bound

\[-2(x_k - z^*)^j \mathbb{E}_k \left[ \tilde{\lambda}_{k+1} W_{k+1} \right] (x_k - z^*) \leq -\frac{2x'_k P \mathbb{W} x_k}{(k+1)^{\theta_1}} + 2\eta \bar{N} \sum_{E} C_{X_i} C_{X_j} \mathbb{E}_k [||H_{k+1, \theta_1}||],\]

(5.4.8)

where we use $P \mathbb{W} z^* = 0$, and $z^* P \mathbb{W} = 0$. Identically to Eqs. (5.3.5) and (5.3.6), we can bound the term

\[(x_k - z^*)^j \mathbb{E}_k \left[ W_{k+1} \tilde{\lambda}_{k+1}^2 W_{k+1} \right] (x_k - z^*) + \mathbb{E}_k [\tilde{\lambda}_{k+1}^2 \tilde{\lambda}_{k+1}] \leq \left( \bar{N} \eta^2 \mu^2 + \bar{N}^2 \eta^2 \sum_{E} C_{X_i} C_{X_j} \right) \sum_{i \in V} \mathbb{E}_k [(\alpha_{k+1}^i)^2].\]

(5.4.9)

Combining Eqs. (5.4.7), (5.4.8) and (5.4.9), we obtain

\[\mathbb{E}_k \left[ ||v_{k+1} - z^*||^2 \right] \leq ||x_k - z^*||^2 + \left( \bar{N} \eta^2 \mu^2 + \bar{N}^2 \eta^2 \sum_{E} C_{X_i} C_{X_j} \right) \sum_{i \in V} \mathbb{E}_k [(\alpha_{k+1}^i)^2] - \frac{2x'_k P \mathbb{W} x_k}{(k+1)^{\theta_1}} + 2\eta \bar{N} \sum_{E} C_{X_i} C_{X_j} \mathbb{E}_k [||H_{k+1, \theta_1}||].\]

(5.4.10)

We now focus on the last two terms in the relation of Lemma 5.4.1. Using the given forms of the step sizes, we can write

\[
\Gamma_{i,k+1} = \frac{1}{\Gamma_{i,k+1}^{\theta_2}} = \frac{1}{p_i^{\theta_2}(k+1)^{\theta_2}} + \left( \frac{1}{\Gamma_{i,k+1}^{\theta_2}} - \frac{1}{p_i^{\theta_2}(k+1)^{\theta_2}} \right).
\]

Define $[H_{k+1, \theta_2}]_i = \frac{1}{\Gamma_{i,k+1}^{\theta_2}} - \frac{1}{p_i^{\theta_2}(k+1)^{\theta_2}}$. Then, we can bound the terms under consideration as

\[-2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1} [f_i(s_k) - f_i(z^*)] + 2CK \sum_{i \in U_{k+1}} \Gamma_{i,k+1} \sum_{j=1}^{m} \sum_{r=1}^{\theta_2} \|x_k^j - x_r^j\| \leq -2 \frac{1}{(k+1)^{\theta_2}} \sum_{i \in U_{k+1}} \frac{1}{p_i^{\theta_2}} [f_i(s_k) - f_i(z^*)] + 2CK \sum_{i \in U_{k+1}} \frac{1}{p_i^{\theta_2}(k+1)^{\theta_2}} \sum_{j=1}^{m} \sum_{r=1}^{\theta_2} \|x_k^j - x_r^j\| + 2C \sum_{i \in U_{k+1}} \sum_{j=1}^{m} \sum_{r=1}^{\theta_2} \|s_k - z^*\| + 2CK \sum_{i \in U_{k+1}} \sum_{j=1}^{m} \sum_{r=1}^{\theta_2} \|H_{k+1, \theta_2}]_i \sum_{j=1}^{m} \sum_{r=1}^{\theta_2} \|x_k^j - x_r^j\|,
\]

where we have used $|f_i(s_k) - f_i(z^*)| \leq C \|s_k - z^*\|$ which follows from the bounded subgradient

71
condition (Assumption 7-e). Now, taking conditional expectation we have

$$
\mathbb{E}_k \left[ -2 \sum_{i \in U_{k+1}} \Gamma_{i,k+1} \left[ f_i(s_k) - f_i(z^*) \right] + 2CK \sum_{i \in U_{k+1}} \Gamma_{i,k+1} \sum_{j=1}^m \sum_{r=1}^m \left\| x^j - x^r \right\| \right]
$$

$$\leq -\frac{2}{(k+1)^{\theta_2}} \frac{1}{\gamma_{i,k+1}} \sum_{i=1}^m p_i^{1-\theta_2} \sum_{j=1}^m \sum_{r=1}^m \left\| x^j - x^r \right\| \right] \mathbb{E}_k \left[ \sum_{i \in U_{k+1}} [H_{k+1,\theta_2}]_i \right].
$$

The result follows from the above relation and Eq. (5.4.10).

### 5.4.2 Convergence Result

We are now ready to derive one of our main results regarding the almost sure convergence of the algorithm in Eq. (5.1.5). In this section we prove that almost sure convergence holds under two cases. In one case we assume that the quadratic form generated by the matrix $\overline{P} \overline{W}$ is positive everywhere except on the consensus subspace $\mathcal{C}$, where it vanishes. In this case, we can prove asymptotic convergence for the step size $\gamma_{i,k+1} = \frac{1}{\Gamma_{i,k+1}}$, i.e., $\theta_2 = 1$. We also require that $\alpha_{i,k} = \frac{1}{\Gamma_{i,k}}$ with the condition $\theta_1 \in (1/2, 1)$. Note that this is a stricter requirement than in the constrained consensus part, where $\theta_1$ could take the value 1. Thus, we need the step sizes $\gamma_{i,k}$ to decay at a faster rate than $\alpha_{i,k}$ for all $i$.

In the second case we assume that the random network has the property that the probability of update $p_i$ is the same for all agents $i$, i.e., $p_1 = \ldots = p_m = p$. In this case the earlier condition reduces to positive semi-definiteness of $\overline{W}$ together with the requirement that $\mathcal{N}(\overline{W}) = \mathcal{C}$. However in this case we are free to chose a wider class of step sizes $\gamma_{i,k} = \frac{1}{\Gamma_{i,k}}$. The faster decay constraint on $\gamma_{i,k}$ in this case arises as the requirement that $\frac{1+\theta_1}{2} < \theta_2 \leq 1$. Note that as illustrated in the section on Gossip and Broadcast communication protocols the requirement of a uniform update probability translates to the requirement that the graph $G$ is regular. In this case the update probability is $p = \frac{2}{m}$ for gossip and $p = \frac{r}{m}$ for broadcast algorithm, where $r$ is the common degree of each node.

The following theorem considers the first case. The theorem relies on the positive semi-definiteness
of the matrix $P\hat{W}$, for which we recall that $P = D\left(\frac{1}{p_i}\right)$ and $p_i$ is the probability of update at any instance by agent $i$.

**Theorem 5.4.3.** Let Assumption 6 on the network hold. Also, let Assumption 7 on the constraint sets and objective functions hold as well as Assumption 8 on the link noise and subgradient errors. Further, let the step sizes be such that $\gamma_k^i = \frac{1}{\Gamma_{i,k}}$ and $\alpha_k^i = \frac{1}{\Gamma_{i,k}}$ with $\theta_1 \in (1/2, 1)$. Assume that the matrix $P\hat{W}$ is positive semidefinite and that the vector $1_m$ is unique (up to scaling) nonzero vector such that $P\hat{W}1_m = 0$. Then, almost surely, the iterate sequences $\{x^i_k\}$ generated by algorithm (5.1.5) converge to a common random point $\tilde{z}^*$ in the optimal set $X^*$.

**Proof.** We use Lemma 5.4.2 with $\theta_2 = 1$ and obtain for any $z^* \in X^*$,

$$E_k \left[ \|x_{k+1} - z^*\|^2 \right] \leq \|x_k - z^*\|^2 - \frac{2}{(k+1)\theta_1} x_k^i P\hat{W} x_k - \frac{2mCK}{k+1} \sum_{j=1}^m \sum_{r=1}^m \|x^j_k - x^r_k\| + \varphi(k),$$

where $f(x) = \sum_{i=1}^m f_i(x)$, $f^*$ is the optimal value of the problem, $z^* = 1_m \otimes z^*$, and $\varphi(k)$ is given by

$$\varphi(k) = \left(\bar{N}\eta^2 \mu^2 + \bar{N}^2 \eta^2 \sum_{E} C_{X_j} C_{X_j'}\right) \sum_{i \in V} E_k \left[ (\alpha_{i+1}^i)^2 \right] + 2(C + \nu)^2 E_k \left[ \sum_{i \in U_{k+1}} \gamma_i^2 (k+1) \right]$$

$$+ 2\eta\bar{N} \sum_{E} C_{X_j} C_{X_j} E_k \left[ \|H_{k+1,\theta_2}\| \right] + 2C \left( \|s_k - z^*\| + K \sum_{j=1}^m \sum_{r=1}^m \|x^j_k - x^r_k\| \right)$$

$$\times E_k \left[ \sum_{i \in U_{k+1}} \|H_{k+1,\theta_2}\| \right].$$

Now, for each pair $\{i, j\} \in E$ in the graph $G = (V, E)$, we can find a path $i = s_1, \ldots, s_\ell = j$ from node $i$ to node $j$. Thus, each term $\|x_i^j - x_i^r\|$ can be bounded above by $\sum_{r=2}^\ell \|s_r, k - x_{s_r-1,k}\|$. Using this we arrive at the bound

$$\frac{2mCK}{(k+1)\theta_1} \sum_{i=1}^m \sum_{j=1}^m \|x_i^j - x_i^r\| \leq \frac{2mCK}{k+1} \left(\frac{m}{2}\right) \sum_{E} \|x_i^j - x_i^r\|. \quad (5.4.11)$$
For convenience of notation, we define \( K_1 = mCK \binom{m}{2} \), then for any constant \( \psi > 0 \) we have

\[
\frac{2K_1(k+1)^{\frac{\theta_1}{2}}}{\psi(k+1)} \frac{\psi}{(k+1)^{\frac{\theta_1}{2}}} \sum_{E} \| x_i^k - x_j^k \| \leq \frac{K_1^2}{\psi^2(k+1)^{2-\theta_1}} + \frac{\psi^2}{(k+1)^{\theta_1}} \sum_{E} \| x_i^k - x_j^k \|^2 = \frac{K_1^2}{\psi^2(k+1)^{2-\theta_1}} + \frac{\psi^2}{(k+1)^{\theta_1}} x_k' L x_k,
\]

where \( L = L \otimes I_n \), and \( L \) is the graph Laplacian for the bidirectional graph \( G \). Substituting this estimate in the relation for \( E_k \left[ \| x_{k+1} - z^* \|^2 \right] \), we obtain

\[
E_k \left[ \| x_{k+1} - z^* \|^2 \right] \leq \| x_k - z^* \|^2 + \frac{K_1^2}{\psi^2(k+1)^{2-\theta_1}} - \frac{1}{(k+1)^{\theta_1}} x_k' [2P\bar{W} - \psi^2L] x_k \leq 2\left[ f(s_k) - f^* \right] + \varphi(k).
\]

We next discuss how to choose \( \psi \) so that the matrix \( 2P\bar{W} - \psi^2L \) is positive semi-definite. Let the eigenvalues of both \( P\bar{W} \) and \( L \) be denoted in the increasing order \( 0 = \lambda_1(P\bar{W}) < \lambda_2(P\bar{W}) \leq \ldots \leq \lambda_m(P\bar{W}) \), and \( 0 = \lambda_1(L) < \lambda_2(L) \leq \ldots \leq \lambda_m(L) \). Note that according to our definition of the matrices \( P\bar{W} \) and \( L \), each of the eigenvalues \( \lambda_i \) has multiplicity \( n \). According to our assumption that \( 1_m \) is the unique null vector of the matrix \( P\bar{W} \), we have \( \lambda_2(P\bar{W}) > 0 \). Moreover, under the assumption that the graph \( G = (V, E) \) is connected, we also have \( \lambda_2(L) > 0 \). Since \( L1_m = 0 \) and \( P\bar{W}1_m = 0 \), it follows \( C \subset N(2P\bar{W} - \psi^2L) \). Any vector can be decomposed as \( x = a_1x_1 + a_2x_2 \), where \( x_1 \in C \) and \( x_2 \perp x_1 \), so that

\[
x' [2P\bar{W} - \psi^2L] x = a_2^2 x_2' [2P\bar{W} - \psi^2L] x_2.
\]

From the variational characterization of eigenvalues, we have for all vectors \( x_2 \perp C \),

\[
2\lambda_2(P\bar{W}) \| x_2 \|^2 \leq 2x_2' P\bar{W} x_2, \quad -\psi^2\lambda_m(L) \| x_2 \|^2 \leq -\psi^2 x_2' L x_2.
\]

Hence, we see that for any vector \( x \) and \( \psi > 0 \),

\[
(2\lambda_2(P\bar{W}) - \psi^2\lambda_m(L)) \| x_2 \|^2 \leq x' [2P\bar{W} - \psi^2L] x.
\]
Thus, for $\psi > 0$ such that $\psi < \sqrt{\frac{2\lambda_2(PW)}{\lambda_m(L)}}$, we see that $2PW - \psi^2L$ is positive semi-definite and its null-space coincides with the consensus space $C$, i.e., $x_k'[2PW - \psi^2L]x_k \geq 0$ with equality holding only for $x \in C$. Next, note that since $\theta_1 \in (1/2, 1)$, we have that $\sum_{k=1}^{\infty} \frac{K_1^2}{\psi^2(k+1)^{2-\theta_1}} < \infty$. Also, by Lemma 5.2.2 and the compactness assumption on the sets $X_i$, we deduce that $\sum_k \phi(k) < \infty$ with probability one. Now, we apply Robbins-Siegmund result (Lemma 2.3.2) and obtain that, with probability one, both $\|x_k - z^*\|^2$ converges for every $z^* \in X^*$ and the following holds:

$$\sum_{k=1}^{\infty} \left[\frac{1}{(k+1)^{\theta_1}} x_k'[2PW - \psi^2L] x_k + \frac{2[f(s_k) - f^*]}{k+1}\right] < \infty.$$  

Since $\frac{1}{k+1} < \frac{1}{(k+1)^{\theta_1}}$ for $\theta_1 < 1$, we have equivalently

$$\sum_{k=1}^{\infty} \frac{1}{k+1} \left[\phi_k x_k'[2PW - \psi^2L] x_k + [f(s_k) - f(z^*)]\right] < \infty.$$  

Since $\sum_{k=1}^{\infty} \frac{1}{k+1} = \infty$ it follows that, with probability one, there exists a subsequence such that $\lim_{k \to \infty} x_{n_k}'[2PW - \psi^2L] x_{n_k} = 0$ and $\lim_{k \to \infty} [f(s_{n_k}) - f^*] = 0$. By our choice of $\psi$, the matrix $2PW - \psi^2L$ is positive semi-definite and it vanishes only on the consensus subspace $C$; therefore, the sequence $x_{n_k}$ approaches the consensus subspace $C$ with probability one, i.e., $\lim_{k \to \infty} \|x_{i,n_k} - x_{j,n_k}\| = 0$ for every $i, j$ with probability one. On the other hand, using Eq. (5.4.6), we obtain

$$\sum_{i=1}^{m} \|x_{i,n_k} - s_{n_k}\| \leq \frac{mB + \delta}{\delta} \sum_{i=1}^{m} \sum_{j=1}^{m} \|x_{i,n_k} - x_{j,n_k}\|.$$  

(5.4.12)

Thus, with probability one we have for every $i$,

$$\lim_{k \to \infty} \|x_{i,n_k} - s_{n_k}\| = 0.$$  

(5.4.13)

Since $\lim_{k \to \infty} f(s_{n_k}) = f^*$ and the function $f$ is continuous, and since the sets $X_i$ are compact (Assumption 7-b), we conclude that there exists a subsequence along which $s_{n_k}$ converges almost surely to a (random) point $\tilde{z}^*$ that lies in the set $X^*$. Without any loss of generality, we can assume that the sequence $s_{n_k}$ itself converges to the limit point $\tilde{z}^*$ almost surely. By Eq. (5.4.13) it follows that $x_{i,n_k}$ converges to $\tilde{z}^*$ for all $i$ almost surely. Recall that $\|x_k - z^*\|^2$ converges almost surely
for any $z^* \in X^*$. Thus, we can consider sample paths for which both $\|x_k - z^*\|^2$ converges for any $z^* \in X^*$ and $x_{i,n_k}$ converges to the corresponding realization of $\tilde{z}^* \in X^*$. In this way, we can conclude that the sequence $\{x_k\}$ must converge to $1_m \otimes \tilde{z}^*$ almost surely, which implies that the sequences $\{x_i^k\}$, $i = 1, \ldots, m$, converge almost surely to a common (random) point in the set $X^*$.

We now provide a convergence result for the case when the probability of updates for each agents is the same. We have the following result.

**Theorem 5.4.4.** Let Assumption 6 on the network hold. Also, let Assumption 7 on the constraint sets and objective functions hold as well as Assumption 8 on the link noise and subgradient errors. Assume that $p_i = p$ for all $i$. Further, let the step sizes be such that $\gamma_k^i = \frac{1}{\Gamma_{i,k}^2}$ and $\alpha_k^i = \frac{1}{\Gamma_{i,k}^2}$, where $\theta_2 > \frac{1+\theta_1}{2}$ and $\theta_1 \in (1/2, 1)$. Then, almost surely, the iterate sequences $\{x_k\}$ of algorithm (5.1.5) converge to a common random point in the optimal set $X^*$.

**Proof.** We use Lemma 5.4.2 where $p_i = p$ for all $i$, and we obtain

$$
E_k \left[ \|x_{k+1} - z^*\|^2 \right] \leq \|x_k - z^*\|^2 - \frac{2p^{-\theta_1}}{(k+1)^{\theta_1}}x_k^*Wx_k - \frac{2p^{-\theta_2}}{(k+1)^{\theta_2}}[f(s_k) - f^*] \tag{5.4.14}
$$

$$
+ \frac{2p^{-\theta_2}CK}{(k+1)^{\theta_2}} \sum_{j=1}^{m} \sum_{r=1}^{m} \left\| x_j^k - x_r^k \right\| + \varphi(k),
$$

where $\varphi(k)$ denotes the remaining terms, $f(x) = \sum_i f_i(x)$, and $f^*$ is the optimal value of the problem. Proceeding similarly as in the derivation of Eq. (5.4.11), we arrive at the following bound:

$$
\frac{2p^{-\theta_2}CK}{(k+1)^{\theta_2}} \sum_{i=1}^{m} \sum_{j=1}^{m} \left\| x_i^k - x_j^k \right\| \leq \frac{2p^{-\theta_2}CK}{(k+1)^{\theta_2}} \binom{m}{2} \sum_{E} \left\| x_i^k - x_j^k \right\|^2.
$$

Let us define $K_2 = p^{-\theta_2} CK \binom{m}{2}$, so we can write

$$
\frac{2K_2}{\psi(k+1)^{\theta_2}} \sum_{(i,j) \in E} \left\| x_i^k - x_j^k \right\| \leq \frac{K_2^2}{\psi^2(k+1)^{2\theta_2-\theta_1}} + \frac{\psi^2}{(k+1)^{\theta_1}} \sum_{E} \left\| x_i^k - x_j^k \right\|^2
$$

$$
= \frac{K_2^2}{\psi^2(k+1)^{2\theta_2-\theta_1}} + \frac{\psi^2}{(k+1)^{\theta_1}} x_k^Lx_k.
$$

76
Substituting this relation in Eq. (5.4.14) we obtain

\[
\mathbb{E}_k \left[ \|x_{k+1} - z^*\|^2 \right] \leq \|x_k - z^*\|^2 + \frac{K^2_2}{\psi^2(k+1)^{2\theta_2 - \theta_1}} - \frac{1}{(k+1)^{\theta_1}} x_k' \left[ 2p^{-\theta_1} \bar{W} - \psi^2 \bar{L} \right] x_k \\
- \frac{2p^{1-\theta_2}}{(k+1)^{\theta_2}} [f(s_k) - f^*] + \varphi(k).
\]

Now, the condition \(\theta_2 > \frac{1+\theta_1}{2}\) implies that \(\sum_{k=0}^{\infty} \frac{K^2_2}{\psi^2(k+1)^{2\theta_2 - \theta_1}} < \infty\). We choose \(\psi\) such that \(0 < \psi < \sqrt{\frac{2p^{-\theta_1} x_2(\bar{W})}{\lambda_{\min}(\bar{L})}}\), so that the matrix \(2p^{-\theta_1} \bar{W} - \psi^2 \bar{L}\) is positive semidefinite. The rest of the proof follows from similar arguments as in Theorem 5.4.3.

5.4.3 Constant step size error bound

In this section we derive asymptotic error bounds when each agent uses constant step sizes \(\alpha_i\) and \(\gamma_i\). The main additional assumption we need the requirement that the local agent functions \(f_i(x)\) are continuously differentiable and strongly convex, i.e., for all \(i \in V\),

\[
(\nabla f_i(x) - \nabla f_i(y))^T (x - y) \geq \sigma_i \|x - y\|^2 \quad \text{for all } x, y.
\]

Note that the convexity parameter \(\sigma_i\) of each agent is allowed to be different. This problem was also considered in [59] in the absence of local constraint sets \(X_i\) on the decision variables.

**Theorem 5.4.5.** Let Assumption 6 on the network hold. Let also Assumptions 7 and 8 hold. Further assume that each function \(f_i\) is continuously differentiable and strongly convex with a constant \(\sigma_i > 0\). Also, assume that the step size \(\alpha_i\) is such that \(\alpha_i \leq \frac{1}{\eta |N_i|}\) and the step size \(\gamma_i\) satisfies \(\gamma_i < \frac{1}{2\sigma_i}\) for all \(i\). Then, for the iterate sequences \(\{x_k^i\}\) of the algorithm in Eq. (5.1.5), we have the following asymptotic bound:

\[
\limsup_{k \to \infty} \mathbb{E} \left[ \|x_k - z^*\|^2 \right] \leq \frac{\epsilon_1}{2\gamma_p p} + \epsilon_2.
\]
where $z^* = 1_m \otimes z^*$ for any $z^* \in X^*$, and

$$
\epsilon_1 = 2(C + \nu)^2 \sum_{i \in V} p_i \gamma_i^2 + 2m \Delta_{\gamma,i} CC + 2 \Delta_{\gamma,i} (1 - p) \sum_{i \in V} C_i^2 \\
+ (1 - 2\gamma) \left( 2\eta \Delta_{\gamma,i} N \sum_E C_i C_{i,j} + m \eta \mu^2 \alpha^2 + \alpha^2 \|R\| \sum_i C_i^2 \right)
$$

$$
\epsilon_2 = \frac{\gamma^2 C^2 \lambda_m(L) \tilde{K}^2}{\gamma \sigma p (1 - 2\gamma) \alpha \lambda_2(W)}, \quad \tilde{K} = m \frac{B + \delta}{2\delta} \binom{m}{2}.
$$

**Proof.** From Eq. (5.4.1) we have the following relation

$$
\mathbb{E}[\|x^i_{k+1} - z^*\|^2 | F_k, W_{k+1}] \leq \mathbb{E}[\|v^i_{k+1} - z^*\|^2 | F_k, W_{k+1}] \\
+ 2\gamma^2 \mathbb{E} \left[ \|\tilde{d}^i_{k+1}\|^2 | F_k, W_{k+1} \right] \chi_{\{i \in U_{k+1}\}} \\
- 2\gamma_i \mathbb{E}[d^i_{i,k+1}(v^i_{k+1} - z^*) | F_k, W_{k+1}] \chi_{\{i \in U_{k+1}\}},
$$

(5.4.15)

with $d^i_k = \nabla f_i(v^i_k)$ and $\tilde{d}^i_k$ is a noisy gradient. Under the assumption of strong convexity of the function $f_i$, we obtain

$$
d'_{i,k+1}(v^i_{k+1} - z^*) = (d^i_{k+1} - \nabla f_i(z^*))'(v^i_{k+1} - z^*) + \nabla f_i(z^*)'(v^i_{k+1} - z^*) \\
\quad \geq \sigma_i \|v^i_{k+1} - z^*\|^2 + \nabla f_i(z^*)'(v^i_{k+1} - z^*).$$

When $i \in U_{k+1}$, we add and subtract the term $-2\gamma_i \nabla f_i(z^*)' s_k$, where $s_k = \frac{1}{m} \sum_{i=1}^m P_X[x^i_k]$ to get

$$
-2\gamma_i d'_{i,k+1}(v^i_{k+1} - z^*) \leq -2\gamma_i \sigma_i \|v^i_{k+1} - z^*\|^2 - 2\gamma_i \nabla f_i(z^*)'(s_k - z^*) + 2\gamma_i \nabla f_i(z^*)'(s_k - v^i_{k+1}).
$$

Then, letting $y_{i,k} = \mathbb{E}[v^i_{k+1} | F_k, W_{k+1}]$ and using the gradient boundedness, we obtain

$$
-2\gamma_i \mathbb{E}[d'_{i,k+1}(v^i_{k+1} - z^*) | F_k, W_{k+1}] \leq -2\gamma_i \sigma_i \mathbb{E}[\|v^i_{k+1} - z^*\|^2 | F_k, W_{k+1}] \\
- 2\gamma_i \nabla f_i(z^*)'(s_k - z^*) + 2\gamma_i C \|y_{i,k} - s_k\|.
$$
Thus, plugging back in Eq. (5.4.15), we get the relation

\[
\mathbb{E}[\|x_{i,k+1}^i - z^*\|^2 | F_k, W_{k+1}] \leq \mathbb{E}[\|v_{i,k+1}^i - z^*\|^2 | F_k, W_{k+1}] + 2\gamma_i^2(C + \nu)^2 \mathbb{E}_i \chi_{i \in U_{k+1}} + 2 \gamma_i \sigma_i (C \|y_{i,k} - s_k\| - \nabla f_i(z^*)'(s_k - z^*)) \chi_{i \in U_{k+1}}.
\]

Using the fact that \(v_{i,k+1}^i = x_k^i\) for \(i \notin U_{k+1}\), the preceding relation can equivalently be written as

\[
\mathbb{E}[\|x_{k+1}^i - z^*\|^2 | F_k, U_{k+1}] \leq (1 - 2\gamma_i \sigma_i) \mathbb{E}[\|v_{k+1}^i - z^*\|^2 | F_k, W_{k+1}] + 2\gamma_i^2(C + \nu)^2 \mathbb{E}_i \chi_{i \in U_{k+1}} + 2 \gamma_i \sigma_i (C \|y_{i,k} - s_k\| - \nabla f_i(z^*)'(s_k - z^*)) \chi_{i \in U_{k+1}}.
\]

Now, taking conditional expectation with respect to the history \(F_k\), we obtain

\[
\mathbb{E}_k[\|x_{k+1}^i - z^*\|^2] \leq (1 - 2\gamma_i \sigma_i) \mathbb{E}_k[\|v_{k+1}^i - z^*\|^2] + 2\gamma_i^2(C + \nu)^2 + 2 \gamma_i \sigma_i \|x_k^i - z^*\|^2 (1 - p)
+ 2 \gamma_i \sigma_i \|y_{i,k} - s_k\| \mathbb{E}_i \chi_{i \in U_{k+1}} - 2 \gamma_i \sigma_i \nabla f_i(z^*)'(s_k - z^*),
\]

where \(p = \min_i \{p_i\}\). Letting \(\Delta_{\gamma,p} = \max_i \{\gamma_ip_i\} - \min_j \{\gamma jp_j\}\) and similarly defining \(\Delta_{\gamma,\sigma}\), we have

\[-\gamma_i \sigma_i \leq -\gamma p + \Delta_{\gamma,p} \text{ and } \gamma_i \sigma_i \leq \gamma \sigma + \Delta_{\gamma,\sigma} \text{. Therefore, it follows}
\]

\[
\mathbb{E}_k[\|x_{k+1}^i - z^*\|^2] \leq (1 - 2\gamma \sigma) \mathbb{E}_k[\|v_{k+1}^i - z^*\|^2] + 2\gamma_i^2(C + \nu)^2 + 2 \gamma \sigma (1 - p) \|x_k^i - z^*\|^2
+ 2 \gamma \sigma \mathbb{E}_k[\|y_{i,k} - s_k\| \mathbb{E}_i \chi_{i \in U_{k+1}} - 2 \gamma \sigma \nabla f_i(z^*)'(s_k - z^*) + 2 \Delta_{\gamma,p} CCX + 2 \Delta_{\gamma,\sigma} (1 - p) C_{X_i}^2.
\]

Summing over all \(i\) we obtain

\[
\mathbb{E}_k[\|x_{k+1} - z^*\|^2] \leq (1 - 2\gamma \sigma) \mathbb{E}_k[\|v_{k+1} - z^*\|^2] + 2(C + \nu)^2 \sum_{i \in V} \gamma_i^2 + 2 \gamma \sigma (1 - p) \|x_k - z^*\|^2
+ 2 \gamma \sigma \mathbb{E}_k \left[ \sum_{i \in U_{k+1}} \|y_{i,k} - s_k\| \right] + 2 \Delta_{\gamma,\sigma} (1 - p) \sum_{i \in V} C_{X_i}^2 - 2 \gamma \sigma \nabla f(z^*)'(s_k - z^*) + 2 m \Delta_{\gamma,p} CCX.
\]

Using relations in (5.4.7) and (5.4.9), where we only use the bound on the term with noise, we can
see that
\[ E_k \left( \|v_{k+1} - z^*\|^2 \right) = \|x_k - z^*\|^2 - 2(x_k - z^*)'\tilde{A}\tilde{W}(x_k - z^*) + \alpha^2 x_k'R x_k + m\tilde{N}\eta^2\mu^2\alpha^2. \]

Further, we have
\[ -2(x_k - z^*)'\tilde{A}\tilde{W}(x_k - z^*) \leq -2\alpha(x_k - z^*)'\tilde{W}(x_k - z^*) + 2\eta\Delta N\sum_E C_{X_i}C_{X_j}, \]
and \( \alpha^2 x_k'R x_k \leq \alpha^2\|R\|\sum_i C_{X_i}^2 \). We also have \( \nabla f(z^*)' (s_k - z^*) \geq 0 \) since \( z^* \) is the optimal point of the objective \( f(x) = \sum_{i \in V} f_i(x) \) and \( s_k \in X \) (recall that \( s_k \) is the average sum of the projections \( P_X[x_{k'}^i] \)). Thus, by neglecting this term and using \( 1 - 2\gamma\sigma < 1 \), we obtain
\[
E_k[\|x_{k+1} - z^*\|^2] \leq (1 - 2\gamma\sigma p)E_k[\|x_k - z^*\|^2] + 2(C + \nu)^2 \sum_{i \in V} p_i\gamma_i^2 + 2\gamma CE_k \left( \sum_{i \in U_{k+1}} \|y_{i,k} - s_k\| \right) \\
+ 2m\Delta_{s,p}CC_X + 2\Delta_{\gamma,\sigma}(1 - p) \sum_{i \in V} C_{X_i}^2 + (1 - 2\gamma\sigma)\alpha^2\|R\|\sum_i C_{X_i}^2 \\
+ (1 - 2\gamma\sigma) \left( 2\eta\Delta N\sum_E C_{X_i}C_{X_j} + m\tilde{N}\eta^2\mu^2\alpha^2 \right). 
\]

We can write the preceding relation more compactly as:
\[ E_k[\|x_{k+1} - z^*\|^2] \leq (1 - 2\gamma\sigma p)E_k[\|x_k - z^*\|^2] + 2\gamma CE_k \left( \sum_{i \in U_{k+1}} \|y_{i,k} - s_k\| \right) + \epsilon_1, \tag{5.4.17} \]
where \( \epsilon_1 \) includes all the other terms. Note that from Eqs. (5.4.5) and (5.4.6) we have
\[
\sum_{i \in U_{k+1}} \|y_{i,k} - s_k\| \leq \frac{mB + \delta}{\delta} \sum_{i \in U_{k+1}} \sum_{j=1}^m \tilde{W}_{ij}(k+1) \sum_{r=1}^m \|x_{k}^j - x_{r}^j\|. 
\]
Since \( \alpha_i \leq \frac{1}{\eta|N_i|} \) for all \( i \), the matrix \( \tilde{W}(k+1) \) is stochastic. Thus, similar to the derivation of relation (5.4.11), we can see that
\[
\sum_{i \in U_{k+1}} \|y_{i,k} - s_k\| \leq m \frac{mB + \delta}{\delta} \sum_{i=1}^m \sum_{j=1}^m \|x_{k}^i - x_{k}^j\| \leq 2\tilde{K} \sum_{E} \|x_{k}^j - x_{k}^j\|. 
\]
where $\tilde{K} = m \frac{mB + 6}{2m} \binom{m}{2}$. We further have

$$2 \tilde{K} \sum_{E} \| x_k^i - x_k^j \| = \frac{2 \tilde{K}}{\psi} \sum_{E} \psi \| x_k^i - x_k^j \| \leq \frac{\tilde{K}^2}{\psi^2} + \psi^2 \sum_{E} \| x_k^i - x_k^j \|^2 = \frac{\tilde{K}^2}{\psi^2} + \psi^2 x_k^L x_k,$$

where $\psi > 0$ is an arbitrary scalar to be chosen later. When substituted back in Eq. (5.4.17) this yields

$$E_k[\| x_k+1 - z^* \|^2] \leq (1 - 2\gamma\sigma p) E_k[\| x_k - z^* \|^2] + \epsilon_1 + 2\tilde{C} \frac{\tilde{K}^2}{\psi^2} - (1 - 2\gamma\sigma) x_k \left[ 2\tilde{C} \tilde{W} - 2 \frac{\tilde{C} \psi^2}{1 - 2\gamma\sigma} L \right] x_k.$$

According to the condition on step sizes $\gamma_i$, we have $1 - 2\gamma\sigma > 0$. Moreover, under the connectivity of the network, we have $\lambda_2(\tilde{W}) > 0$ and $\lambda_2(L) > 0$. Then, it can be seen that for any $\psi$ such that $0 < \psi \leq \sqrt{\frac{(1 - 2\gamma\sigma) \alpha \lambda_2(\tilde{W})}{\tilde{C} \lambda_m(L)}}$, the matrix $2\tilde{C} \tilde{W} - 2 \frac{\tilde{C} \psi^2}{1 - 2\gamma\sigma} L$ is positive semi-definite. Neglecting that term and taking full expectation, we obtain

$$E[\| x_k+1 - z^* \|^2] \leq (1 - 2\gamma\sigma p) E[\| x_k - z^* \|^2] + \epsilon_1 + 2\tilde{C} \frac{\tilde{K}^2}{\psi^2}.$$

Now we can apply the result of Lemma 5.2.1 to deduce

$$\limsup_{k \to \infty} E \left[ \| x_k - z^* \|^2 \right] \leq \frac{\epsilon_1}{2\gamma\sigma p} + \frac{\tilde{C} \tilde{K}^2}{\gamma\sigma p \psi^2}.$$

We can minimize the error by letting $\psi = \sqrt{\frac{(1 - 2\gamma\sigma) \alpha \lambda_2(\tilde{W})}{\tilde{C} \lambda_m(L)}}$, which when substituted in the preceding relation yields the desired result.

Note that the condition $\gamma_i < \frac{1}{2\sigma}$ can be easily implemented in a distributed manner as each agent knows its own objective function. The error $\epsilon_2 = \frac{\tilde{C}^2 \lambda_m(L) \tilde{K}^2}{\gamma\sigma p (1 - 2\gamma\sigma) \alpha \lambda_2(\tilde{W})}$ emphasizes the effect of the network structure on the asymptotic error. It can be seen that networks with large eigenvalue $\lambda_2(\tilde{W})$ have reduced asymptotic error. To see the effect of step size on the error $\epsilon_2$, let us assume that the step sizes $\gamma_i = \gamma$ for all $i$. Then, the error $\epsilon_2$ reduces to $\epsilon_2 = \frac{\gamma \lambda_m(L) \tilde{K}^2}{\sigma p (1 - 2\gamma\sigma) \alpha \lambda_2(\tilde{W})}$, which evidently decreases with decreasing step size $\gamma$. Note that for the special case when all the step sizes $\alpha_i$ and $\gamma_i$ are chosen to be equal across the agents, the probability of update for all agents
is $p$, and the functions $f_i$ have the same convexity parameter $\sigma$, the upper bound reduces to
\[
\epsilon = \frac{m(C+\nu)^2}{\sigma} + \frac{1}{p} \left( \frac{1}{2\gamma \sigma} - 1 \right) (m\bar{\eta}^2 \mu^2 + \alpha^2 \|R\| \sum_i C_i^2 \gamma_i) + \epsilon_2.
\] It is clear that the first term is the contribution of the subgradient error and the term involving $\mu^2$ is the contribution of the communication noise. As noted earlier, for gossip and broadcast algorithms the norm $\|R\|$ can be written in terms of $\lambda_m(W)$. We can choose to minimize the total error bound by choosing the various step sizes carefully. Another interesting fact is that the convexity parameters affect the error bound inversely. This implies that the asymptotic error is smaller when the objective functions have a higher curvature as characterized by their convexity parameter.

## 5.5 Conclusion

In this Chapter we considered the problem of reaching agreement on a set of local variables and the problem of minimizing the sum of local objective functions when the local variables are constrained to local convex constraint sets. We proposed algorithms for these problems which can be applied over a random communication network. We showed that our model of the random communication network is general enough to include the widely used gossip and broadcast based communication protocols arising in wireless networks. Further, our algorithms are robust to the presence of communication noise and errors in the evaluation of subgradients of the objective functions. This generality allows us to consider the distributed stochastic optimization problem in our framework. We established conditions under which we can guarantee almost sure convergence of our algorithms, and provided asymptotic error bounds when almost sure convergence cannot be achieved.
Chapter 6

Distributed Bregman Distance Algorithms for Min-Max Optimization

In Chapters 4 and 5 we presented distributed synchronous and asynchronous algorithms respectively for minimizing an objective function which is composed of a sum of local convex objective functions. These algorithms were motivated by subgradient descent algorithms and used the power of distributed averaging to guarantee convergence. The contribution of this chapter is twofold. We extend the distributed algorithm based on subgradient descent algorithms to Bregman descent algorithms and we consider a min-max problem formulation which doesn’t fit the separable structure of the cost function considered earlier.

We are interested in finding an optimal decision variable which minimizes the worst case loss incurred to any agent. This is based on the idea of min-max fairness in a resource allocation problem [60], whereas the formulation in Chapters 4 and 5 deals with minimizing the sum total loss incurred by agents. This Chapter is based on [61] and [62].

The rest of the chapter is organized as follows. In Section 6.1 we formally state our problem of interest and reformulate it in a way which makes the problem more suitable to derive distributed algorithms. Next, in Section 6.2 we present our discussion on a distributed Bregman distance based algorithm which utilizes the exact penalty function approach. In this section we state our algorithm and assumptions and derive the convergence result. In Section 6.3 we present an alternative algorithm which builds upon the primal-dual algorithm of Arrow-Hurwicz-Uzawa [63]. This algorithm paves a way to handle problem formulations in which the network plays a min-max game against an exogenous signal. This aspect is discussed in Section 6.4. Finally, in Section 6.5 we present an example of min-max power allocation in a cellular network and provide simulation results showing the convergence behavior of both the exact penalty approach and the primal-dual approach.

In this Chapter we additionally use the symbol \( \langle x, y \rangle \) to denote the inner-product \( x^T y \) between any two vectors \( x \) and \( y \).
6.1 Problem Formulation

In this Chapter we restrict ourself to the synchronized network model as discussed in Chapters 3 and 4. For ease of reference we state the problem setup again. We are given a set of $m$ agents, which can be viewed as the node set $V = \{1, \ldots, m\}$, and we use the terms node and agent interchangeably. We assume that the time is discrete and use $k = 0, 1, \ldots$ to denote the time instances. The agents communicate with each other over a time-varying communication network. At any time $k$, the communications among the agents are represented by a directed graph $G(k) = (V, E(k))$ with an edge-set $E(k)$ that has a link $(i, j) \in E(k)$ if and only if agent $i$ receives information from agent $j$ at time $k$.

We consider solving a distributed multi-agent optimization problem subject to local agent communications, where the agents want to cooperatively solve the following problem:

\[
\min_{x \in X} \max_{i \in V} f_i(x). \tag{6.1.1}
\]

Here each $f_i : \mathbb{R}^n \to \mathbb{R}$ is a convex function, representing a local objective function known only by agent $i$. The set $X \subseteq \mathbb{R}^n$ is a closed and convex set known by all agents. In this chapter we relax the requirement that $\mathbb{R}^n$ is equipped with the Euclidean norm. We assume that it is equipped with a general vector norm $\| \cdot \|$, with a corresponding dual norm $\| \cdot \|_*$. The goal is to develop a distributed algorithm for solving the constrained optimization problem in (6.1.1). We are interested in the case when the agents’s objective functions $f_i$ are not necessarily differentiable. As in previous chapters we allow the local objective functions $f_i$ to take the form of the following stochastic optimization:

\[
f_i(x) = \mathbb{E}_{\omega_i}[F_i(x, \omega_i)] + \Omega_i(x), \tag{6.1.2}
\]

where the expectation is taken with respect to the distribution of the random variable $\omega_i$, and the term $\Omega_i(x)$ is a regularization term included to improve the generalization ability [43]. A more general problem formulation can be arrived at by considering the constraint set $X$ as the intersection of local constraint sets as $X = \cap_{i=1}^m X_i$. This formulation was considered in Chapters 3 and 4 and was extended to the case of noisy communication, noisy gradients and in the asynchronous setting in Chapter 5. Though, the current approach can be significantly extended to the distributed
constraint sets setting, in the current work we restrict ourself to the common constraint set \( X \).

We now reformulate the problem (6.1.1) which makes casts it in a more suitable form for developing distributed algorithms.

### 6.1.1 Problem Reformulation

The min-max-problem in (6.1.1) is a convex problem, as the function \( f(x) = \max_{i \in V} f_i(x) \) is convex since pointwise maximum of convex functions preserves convexity [28], Proposition 1.2.4, page 30. However, the form of the min-max problem is not suitable for distributed optimization over the agent network. In our approach, we find it useful to use an epigraph representation of the min-max problem in (6.1.1). In particular, we let \( \eta \in \mathbb{R} \) and re-cast problem (6.1.1) in an equivalent form:

\[
\begin{align*}
\text{minimize} & \quad \eta \\
\text{subject to} & \quad f_i(x) \leq \eta \quad \forall x \in X, \eta \in \mathbb{R}, \text{ and } i \in V.
\end{align*}
\]

In this formulation, the decision variables are \( x \) and \( \eta \). We assume throughout the paper that the optimal value of the problem is finite and we denote its optimal value by \( \eta^* \). An optimal decision \( x^* \in X \) for the problem satisfies \( f_i(x^*) \leq \eta^* \) for all \( i \). We will denote the set of optimal decisions by \( X^* \). Note that \( X^* \) is the optimal set for the original min-max problem (6.1.1), while \( \eta^* \) is its optimal value.

We provide two algorithms for the distributed min-max problem. The first algorithm is based on an exact penalty function based approach and the second algorithm is a primal-dual algorithm. Both the algorithms employ Bregman distance functions.

### 6.2 Exact Penalty Function Approach

We further transform the problem in (6.1.3) by penalizing the constraints to obtain the following problem:

\[
\begin{align*}
\min_{x \in X, \eta \in \mathbb{R}} & \quad \eta + \sum_{i=1}^{m} r_i g_i(x, \eta), \\
\text{where} & \quad g_i(x, \eta) = \max \{ 0, f_i(x) - \eta \}
\end{align*}
\]
is a penalty function and $r_i > 0$ is a penalty parameter for violating the constraint $f_i(x) \leq \eta$. This problem has a form suitable for our development of a distributed algorithm, as its objective can be written as $\sum_{i=1}^{m}(\eta/m + r_i g_i(x, \eta))$ and each summand can be viewed as an objective function of agent $i$.

Under certain conditions on problem (6.1.3) and the penalty parameters $r_i$, the solutions of the penalized problem (6.2.1) are also the solutions of the constrained problem (6.1.3). We next discuss these conditions and show that they are satisfied. The problem of using penalty functions to recover the exact solution of the constrained problem was dealt in [64]. There, a general constrained convex optimization problem was considered, and necessary and sufficient conditions were derived for its equivalence to a penalized problem. The result in [64], when specialized to the problem at hand implies that a solution of the penalized problem (6.2.1) is also a solution of the constrained problem (6.1.3) if and only if $r_i$ is strictly greater than the optimal dual variable corresponding to the $i^{th}$ constraint in (6.1.3). Note that in the agent setting this would imply that agent $i$ should choose its own $r_i$ based on knowledge of an optimal dual variable associated with its constraint $f_i(x) \leq \eta$. Let us consider a dual problem to (6.1.3), by introducing its Lagrangian function, defined as:

$$L(x, \eta, \mu) = \eta + \sum_{i=1}^{m} \mu_i (f_i(x) - \eta), \quad (6.2.2)$$

where $\mu = (\mu_1, \ldots, \mu_m)'$ is the vector of dual variables satisfying $\mu_i \geq 0$ for all $i \in V$. The dual problem is

$$\max_{\mu \geq 0} q(\mu) \quad \text{with} \quad q(\mu) = \inf_{x \in X, \eta \in \mathbb{R}} L(x, \eta, \mu). \quad (6.2.3)$$

It can be easily checked that the Slater condition is satisfied for problem (6.1.3) and, hence, there is no duality gap between the primal problem (6.1.3) and its dual (6.2.3). Furthermore, the set of dual optimal solutions is nonempty and bounded. The bound for the dual optimal variables can be found by rewriting the Lagrangian function (6.2.2) as $L(x, \eta, \mu) = (1 - \sum_{i=1}^{m} \mu_i) \eta + \sum_{i=1}^{m} \mu_i f_i(x)$. Then, we note that $\inf_{\eta \in \mathbb{R}} L(x, \eta, \mu) = -\infty$ when $\sum_{i=1}^{m} \mu_i \neq 1$, implying that $q(\mu) = -\infty$ when $\sum_{i=1}^{m} \mu_i \neq 1$. Thus, the domain of the dual function $q$ is the set of multipliers $\mu \geq 0$ such that
\[ \sum_{i=1}^{m} \mu_i = 1, \text{ implying that the optimal multipliers } \mu_i^* \text{ must satisfy} \]

\[ \sum_{i=1}^{m} \mu_i^* = 1. \quad (6.2.4) \]

Thus, when the penalty parameters satisfy \( r_i > 1 \) for all \( i \), then the problems in (6.2.1) and (6.1.3) are equivalent. Note that in the multiagent case, the agents just need to know that their individual penalty coefficient is greater than 1. We focus on solving problem (6.2.1).

### 6.2.1 Equivalence Between Epigraph and Penalty Problem Formulations

Our first lemma establishes an important relation between the optimal solutions of the min-max problem formulation in (6.1.3) and its penalized counterpart in (6.2.1). In the proof of the lemma, we use the saddle-point theorem characterizing the optimal solutions of the problem (6.1.3) and its dual problem (6.2.3), as given for example in [28], Proposition 6.2.4, page 360.

**Theorem 6.2.1.** *(Saddle-Point Theorem)* The pair \((z^*, \mu^*)\) with \(z^* = (x^*, \eta^*) \in X \times \mathbb{R}\) and \(\mu^* \geq 0\) is a primal-dual optimal solution pair if and only if the following relation holds:

\[
L(z^*, \mu) \leq L(z^*, \mu^*) \leq L(z, \mu^*),
\]

for all \(z = (x, \eta) \in X \times \mathbb{R}\) and \(\mu \geq 0\), i.e., \((z^*, \mu^*)\) is a saddle-point of the Lagrangian function \(L(z, \mu)\).

Now, we state the lemma.

**Lemma 6.2.2.** Let \(\eta^* = \min_{x \in X} \max_{i \in V} f_i(x)\) and \(r_i > 1\) for all \(i\). Then, for \(g_i(x, \eta) = \max\{0, f_i(x) - \eta\}\) we have

\[
\sum_{i=1}^{m} r_i g_i(x, \eta) + \eta \geq \eta^* \quad \text{for all } x \in X \text{ and } \eta \in \mathbb{R}.
\]

Furthermore, equality holds in the above expression if and only if \(\eta = \eta^*\) and \(x = x^*\) for an optimal solution \(x^*\) of the problem \(\min_{x \in X} \max_{i \in V} f_i(x)\).

**Proof.** Consider the definition of the Lagrangian in (6.2.2). Then, it follows that we have \(\eta^* = L(z^*, \mu^*)\). Now, for the given \(x \in X\) and \(\eta\), let us define the dual variables \(\mu_i\) such that \(\mu_i = r_i\) if
\[ f_i(x) - \eta \geq 0, \text{ and } \mu_i = 0 \text{ if } f_i(x) - \eta < 0 \text{ or compactly } \mu_i = r_i 1_{\{f_i(x) \geq \eta\}}. \] Then, clearly we have
\[
\sum_{i=1}^{m} r_i \max\{f_i(x) - \eta, 0\} + \eta = \sum_{i=1}^{m} \mu_i(f_i(x) - \eta) + \eta = L(z, \mu).
\]

We need to prove that \( L(z, \mu) - L(z^*, \mu^*) \geq 0 \). From Theorem 6.2.1, we have \( -L(z^*, \mu^*) \geq -L(z, \mu^*) \), implying
\[
L(z, \mu) - L(z^*, \mu^*) \geq L(z, \mu) - L(z, \mu^*) = \sum_{i=1}^{m} (\mu_i - \mu_i^*)(f_i(x) - \eta)
\]
\[
= \sum_{i=1}^{m} (r_i - \mu_i^*) 1_{\{f_i(x) \geq \eta\}}(f_i(x) - \eta) - \sum_{i=1}^{m} \mu_i^* 1_{\{f_i(x) < \eta\}}(f_i(x) - \eta) \geq 0,
\]
where we have used the decomposition \( \mu_i^* = \mu_i^* 1_{\{f_i(x) \geq \eta\}} + \mu_i^* 1_{\{f_i(x) < \eta\}} \) and used the facts that \( 0 \leq \mu_i^* \leq 1 \) (see (6.2.4)) and \( r_i > 1 \) for all \( i \).

We now prove the second assertion. By the definition of min-max solution we have \( f_i(x^*) \leq \eta^* \) for all \( i \), so that \( \sum_{i=1}^{m} r_i \max\{f_i(x^*) - \eta^*, 0\} + \eta^* = \eta^* \). Thus, we just need to prove the “only if” part. For this, let us assume that for some \( \bar{x} \in X \) and \( \bar{\eta} \) we have
\[
\sum_{i=1}^{m} r_i \max\{f_i(\bar{x}) - \bar{\eta}, 0\} + \bar{\eta} = \eta^*.
\]
(6.2.5)

Since \( \sum_{i=1}^{m} r_i \max\{f_i(\bar{x}) - \bar{\eta}, 0\} \geq 0 \), it follows \( \bar{\eta} \leq \eta^* \). Let us assume that \( \bar{\eta} < \eta^* \), then for the equality to hold we must have \( f_j(\bar{x}) > \bar{\eta} \) for some \( j \). Thus, \( f_i^*(\bar{x}) > \bar{\eta} \) for \( i^* = \text{argmax}_i f_i(\bar{x}) \). By \( \eta^* = \min_{x \in X} \max_{i \in V} f_i(x) \) we have \( f_i^*(\bar{x}) \geq \eta^* \) implying \( f_i^*(\bar{x}) - \bar{\eta} \geq \eta^* - \bar{\eta} > 0 \). Since \( r_i > 0 \), it follows that \( r_i^*(f_i^*(\bar{x}) - \bar{\eta}) > \eta^* - \bar{\eta} \). Therefore, \( \sum_{i=1}^{m} r_i \max\{f_i(\bar{x}) - \bar{\eta}, 0\} + \bar{\eta} > \eta^* \), which is contradics (6.2.5) so we must have \( \bar{\eta} = \eta^* \) in (6.2.5). This, however, by \( r_i > 0 \) yields \( f_i(\bar{x}) \leq \eta^* \) for all \( i \), thus showing that \( \bar{x} \) is a min-max solution. \( \square \)

### 6.2.2 Algorithm

Here, we present a distributed multi-agent algorithm for solving the penalty reformulation (6.2.1) of the min-max problem, where a penalty function \( g_i(x, \eta) \) is associated with an agent \( i \) and \( r_i > 1 \) for all \( i \). Let \( x_j^k \) and \( \eta_j^k \) be the decision variables of agent \( j \) at time \( k \), which are agent \( j \) estimates of an optimal solution \( x^* \) and the optimal value \( \eta^* \) of the problem. Recall that the agents’
communications at time $k$ are represented with a graph $G(k) = (V, E(k))$, where $(i, j) \in E(k)$ if agent $i$ receives estimates $x^j(k)$ and $\eta^j(k)$ from agent $j$. To capture this information exchange, let $N_i(k)$ denote the set of neighbors of agent $i$, i.e., $N_i(k) = \{j \in V \mid (i, j) \in E(k)\}$. Let us introduce the strongly convex function $\omega_x : X \to \mathbb{R}$, with the convexity parameter $\sigma_x$. Let us also introduce strongly convex function $\omega_\eta : \mathbb{R} \to \mathbb{R}$, with the convexity parameter $\sigma_\eta$. Let us denote the Bregman distance functions generated by these strongly convex functions by $V_x(\cdot, \cdot)$, and $V_\eta(\cdot, \cdot)$ respectively, i.e.

$$V_x(y, z) = \omega_x(y) - \left[ \omega_x(z) + \nabla \omega_x(z)^T (y - z) \right]$$

$$V_\eta(y, z) = \omega_\eta(y) - \left[ \omega_\eta(z) + \nabla \omega_\eta(z)^T (y - z) \right].$$

Upon receiving the information from its neighbors, each agent $i$ performs an intermittent adjustment of its estimates as follows:

$$\begin{bmatrix}
\tilde{x}^i_k \\
\tilde{\eta}^i_k
\end{bmatrix} = \sum_{j \in N_i(k)} w_{ij}(k) \begin{bmatrix} x^j_k \\ \eta^j_k \end{bmatrix},$$

(6.2.6)

where $w_{ij}(k) \geq 0$ is a weight that agent $i$ assigns to its neighbor $j \in N_i(k)$. For a compact representation of relation (6.2.6), let $w_{ij}(k) = 0$ for all $j \not\in N_i(k)$ and introduce a matrix $W_k$ with entries $w_{ij}(k)$. With this, the intermittent adjustment in (6.2.6) can be written as follows:

$$v^i_k = \begin{bmatrix}
\tilde{x}^i_k \\
\tilde{\eta}^i_k
\end{bmatrix} = \sum_{j=1}^{m} [W_k]_{ij} \begin{bmatrix} x^j_k \\ \eta^j_k \end{bmatrix}.$$

(6.2.7)

After the intermittent adjustment, each agent takes a step toward minimizing its own penalty function through an adjustment:

$$x^i_{k+1} = \arg\min_{y \in X} \left[ \alpha_k r_i (\nabla_x g_i(v^i_k) + \epsilon^i_k, y) + V_x(y, \tilde{x}^i_k) \right]$$

$$\eta^i_{k+1} = \arg\min_{s \in \mathbb{R}} \left[ \alpha_k \left\{ \frac{1}{m} + r_i \nabla_\eta g_i(v^i_k), s \right\} + V_\eta(s, \tilde{\eta}^i_k) \right],$$

(6.2.8)
where $r_i > 1$ and $\alpha_k > 0$ is a stepsize. The brackets $\langle x, y \rangle$ denote the standard inner product $x^T y$. The notation $\nabla_x g_i(x, \eta)$ denotes the term $\nabla f_i(x) \mathbf{1}_{\{f_i(x) \geq \eta\}}$ and $\nabla_\eta g_i(x, \eta)$ denotes the term $-\mathbf{1}_{\{f_i(x) \geq \eta\}}$, where $\mathbf{1}_{\{f_i(x) \geq \eta\}}$ is the indicator function which takes the value 1 when $f_i(x) \geq \eta$ and 0 otherwise. The first update involves taking a step along an erroneous subgradient of $\nabla_x g_i(v^i_k)$, where $\epsilon^i_k$ is a subgradient error. Let's take a closer look at that update. The agent $i$ objective function $g_i(x, \eta) = \max \{0, f_i(x) - \eta\}$ is not differentiable at the point $(x, \eta)$ where $f_i(x) - \eta = 0$. Then, a subgradient of the function $g_i$ at $(x, \eta)$ is given by

$$\nabla g_i(x, \eta) = \begin{bmatrix} \nabla f_i(x) \\ -1 \end{bmatrix} \mathbf{1}_{\{f_i(x) \geq \eta\}},$$

where $\nabla f_i(x)$ denotes a subgradient of the function $f_i$ at $x$. Since each function $f_i$ is assumed to be convex over the entire space $\mathbb{R}^n$, the subdifferential set $\partial f_i(x)$ is nonempty for all $x$ and $i \in V$ [28], Proposition 4.2.1. Thus, the function $g_i$ also has a nonempty subdifferential set at any point $(x, \eta)$.

The initial points $x^0_i \in X$ and $\eta^0_i$ may be selected randomly with the distribution independent of any other sources of randomness in the algorithm. The subgradient error $\epsilon^i_k$ is assumed to be stochastic in order to tackle the general form of the objective function as in (6.1.2), where the subgradient $\nabla f_i(x)$ is not readily available to us. We adopt a standard approach in stochastic optimization by using an unbiased estimate $\nabla f_i(x) + \bar{\epsilon}^i_k$ of the subgradient, where $\bar{\epsilon}^i_k$ is a zero mean random variable. Thus, in (6.2.8) we have

$$\epsilon^i_k = \bar{\epsilon}^i_k \mathbf{1}_{\{f_i(\tilde{x}^i_k) \geq \tilde{\eta}^i_k\}}.$$

If $\mathbb{R}^n$ is equipped with Euclidean norm and the Bregman distance functions are chosen as $\omega_x = \|x\|^2$ and $\omega_\eta = \eta^2$, then the algorithm reduces to:

$$\begin{bmatrix} x^i_{k+1} \\ \eta^i_{k+1} \end{bmatrix} = P_{X \times \mathbb{R}} \left[ v^i_k - \alpha_k r_i \left( \nabla g_i(v^i_k) + \begin{bmatrix} \epsilon^i_k \\ 0 \end{bmatrix} \right) \right].$$

90
6.2.3 Assumptions

Our assumptions on the network are similar to those in [7], and those in Chapter 4. We assume that there is no communication noise present. Hence, unlike Chapter 4 we do not need to include a step size to suppress communication noise. The assumptions ensure that agent’s local variables are properly diffused over the communication networks in time.

**Assumption 9.** For the weight matrices and the communication graphs, we assume the following:

a) *(Weights Rule)* There exists a scalar $0 < \gamma < 1$ such that $[W_k]_{ii} \geq \gamma$ for all $i$ and $k$, and $[W_k]_{ij} \geq \gamma$ if $[W_k]_{ij} > 0$.

b) *(Doubly Stochasticity)* The matrix $W_k$ is doubly stochastic for all $k$, i.e., $\sum_{j=1}^{m}[W_k]_{ij} = 1$ and $\sum_{i=1}^{m}[W_k]_{ij} = 1$.

c) *(Connectedness)* There exists an integer $B \geq 1$ such that the graph $(V, \cup_{\tau=KB-1}^{(k+1)B-1} E(\tau))$ is strongly connected for every $k$.

Our assumptions on the subgradient and the subgradient noise is same as in Chapter 4. However the norm is understood to be the dual $\|\cdot\|_*$ to the general norm $\|\cdot\|$ on $\mathbb{R}^n$. We restate these assumptions for ease of reference. We specifically impose the following assumptions on the subgradients $\nabla f_i(x)$ and the errors $\bar{\epsilon}_k^i$.

**Assumption 10.** Let the following hold:

a) The subgradients of each $f_i$ are bounded over the set $X$, i.e., there is a scalar $C > 0$ such that $\|\nabla f_i(x)\|_* \leq C$ for all $\nabla f_i(x) \in \partial f_i(x)$, all $x \in X$ and all $i \in V$.

b) The subgradient errors $\bar{\epsilon}_k^i$ when conditioned on the point $x = \bar{x}_k^i$ of the subgradient $\nabla f_i(x)$ evaluation are zero mean, i.e., $E[\bar{\epsilon}_k^i | \bar{x}_k^i] = 0$ for all $i \in V$ and $k \geq 0$, with probability 1.

c) There is a scalar $\nu > 0$ such that $E[\|\bar{\epsilon}_k^i\|_*^2 | \bar{x}_k^i] \leq \nu^2$ for all $i \in V$ and $k \geq 0$, with probability 1.

Basically, under Assumptions 10-b and 10-c, the iterations $\{x_k^i\}, i \in V$, of the algorithm in (6.2.7)–(6.2.8) form a Markov process. In what follows, we will use $F_k$ to denote the past iterates of the algorithm (6.2.8), i.e.,

$$F_k = \{x_t^i, \eta_t^i, i \in V, t = 0, 1, \ldots, k\} \quad \text{for } k \geq 0.$$
Note that, given $F_k$, the iterates $\tilde{x}_k^i$ and $\tilde{\eta}_k^i$ in (6.2.7) are deterministic. In view of this, as a consequence of the subgradient norm and subgradient error boundedness (Assumptions 10-a and 10-c), it can be seen that with probability 1,

$$E \left[ \| \nabla f_i(x) + \tilde{\epsilon}_k^i \|^2 \right| F_k \right] \leq (C + \nu)^2 \quad \text{for all } i \in V \text{ and } k \geq 0.$$  

Also, note that as a result of Assumption 10-a we have

$$\| \nabla g_i(\tilde{v}_k^i) \|_\ast = \| \nabla f_i (\tilde{x}_k^i) \|_\ast \mathbf{1}_{\{f_i(\tilde{x}_k^i) \geq \tilde{\eta}_k^i \}} \leq C, \quad \text{for all } i \in V \text{ and } k \geq 0.$$  

This and the zero-mean error assumption (Assumption 10-b) yield

$$E[\tilde{\epsilon}_k^i \mid F_k] = E[\tilde{\epsilon}_k^i \mathbf{1}_{\{f_i(\tilde{x}_k^i) \geq \tilde{\eta}_k^i \}} \mid F_k] = E \left[ \tilde{\epsilon}_k^i \right| F_k \right] \mathbf{1}_{\{f_i(\tilde{x}_k^i) \geq \tilde{\eta}_k^i \}} = 0.$$  

Similarly, as a result of Assumption 10-c we have with probability 1,

$$E \left[ \| \tilde{\epsilon}_k^i \|^2 \right| F_k \right] = E \left[ \| \tilde{\epsilon}_k^i \|^2 \mathbf{1}_{\{f_i(\tilde{x}_k^i) \geq \tilde{\eta}_k^i \}} \right| F_k \right] = E \left[ \| \tilde{\epsilon}_k^i \|^2 \right| F_k \right] \mathbf{1}_{\{f_i(\tilde{x}_k^i) \geq \tilde{\eta}_k^i \}} \leq \nu^2.$$  

This, in turn implies that with probability 1 for all $i \in V$ and $k \geq 0$,

$$E \left[ \| \nabla x g_i(\tilde{z}_k^i) + \epsilon_k^i \|^2 \right| F_k \right] \leq (C + \nu)^2.$$  

(6.2.10)

Applying Jensen’s inequality, we find that

$$E \left[ \| \nabla x g_i(\tilde{z}_k^i) + \epsilon_k^i \|^2 \right| F_k \right] \leq (C + \nu)^2.$$  

(6.2.11)

By the definition, a Bregman function is convex in the first variable. We further make the assumption on the choice of Bregman distance functions that requires convexity with respect to the second variable. We depend on this assumption when showing the convergence of our algorithms.

**Assumption 11.** We assume that both the Bregman distance functions $V_x(x, z)$ and $V_\eta(\eta, \zeta)$ are convex in their second arguments $z$ and $\zeta$, for every fixed $x$ and $\eta$, respectively.
6.2.4 Preliminary Results

We now show some preliminary relations for the iterates of the algorithm. We start with a result related to the instantaneous averages of the iterates $x^i_k$ and $\eta^i_k$ over $i \in V$, defined by

$$\hat{x}_k = \frac{1}{m} \sum_{i=1}^{m} x^i_k, \quad \hat{\eta}_k = \frac{1}{m} \sum_{i=1}^{m} \eta^i_k$$ for all $k \geq 0$.

Before we state our result we discuss a result from [7] which will be helpful in proving our result. In [7] distributed algorithms have been considered of the form

$$x^i_{k+1} = P_X \left[ \hat{x}^i_k - \gamma_k \tilde{d}^i_k \right]. \quad (6.2.12)$$

There in Theorem 6.1 it has been shown that, for the network model satisfying Assumption 9, if the perturbation term $\tilde{d}^i_k$ is bounded in the sense $E \left[ \| \tilde{d}_k^i \|^2 | F_k \right] \leq K$, the set $X$ is closed and convex and the step size satisfies $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$, then we have

$$\sum_{k=0}^{\infty} \alpha_k \| \hat{x}_k - x^j_k \| < \infty, \quad \text{for all } j. \quad (6.2.13)$$

We now prove a similar result for our algorithm.

**Lemma 6.2.3.** Let Assumptions 9 and 10 hold, and let the step sizes satisfy $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$. Then, for the iterates of the algorithm (6.2.8) we have with probability 1 for all $j \in V$,

$$\sum_{k=0}^{\infty} \alpha_k \| \hat{x}_k - x^j_k \| < \infty, \quad \sum_{k=0}^{\infty} \alpha_k |\hat{\eta}_k - \eta^j_k| < \infty.$$

**Proof.** Let us denote the noisy subgradient as $\tilde{d}^i_k = \nabla_x g_i(v^i_k) + \epsilon^i_k$. Applying the optimality condition for (6.2.8) we get

$$\langle \alpha_k r^i_k \tilde{d}^i_k + \nabla \omega_x(x^i_{k+1}) - \nabla \omega_x(\hat{x}^i_k), y - x^i_{k+1} \rangle \geq 0 \quad \text{for all } y \in X.$$

Since $\hat{x}^i_k \in X$, by letting $y = \hat{x}^i_k$ we have

$$\langle \alpha_k r^i_k \tilde{d}^i_k + \nabla \omega_x(x^i_{k+1}) - \nabla \omega_x(\hat{x}^i_k), \hat{x}^i_k - x^i_{k+1} \rangle \geq 0.$$
This implies that

\[
\alpha_k r_i (\tilde{d}_k^i)'(\tilde{x}_k^i - x_{k+1}^i) \geq (x_{k+1}^i - \tilde{x}_k^i)'(\nabla \omega(x_{k+1}^i) - \nabla \omega(\tilde{x}_k^i)).
\]  

(6.2.14)

Since \( \omega(\cdot) \) is a strongly convex function with convexity parameter \( \sigma_x \) we have

\[
(x_{k+1}^i - \tilde{x}_k^i)'(\nabla \omega(x_{k+1}^i) - \nabla \omega(\tilde{x}_k^i)) \geq \sigma_x \| \tilde{x}_k^i - x_{k+1}^i \|^2.
\]

Using this fact and the Hölder’s inequality in (6.2.14) we have

\[
\alpha_k r_i \| \tilde{d}_k^i \|_s \| \tilde{x}_k^i - x_{k+1}^i \| \geq \sigma_x \| \tilde{x}_k^i - x_{k+1}^i \|^2.
\]

This yields

\[
\| \tilde{x}_k^i - x_{k+1}^i \| \leq \frac{\| \tilde{d}_k^i \|_s}{\sigma_x},
\]

which implies that, upon taking conditional expectation on \( F_k \) we have

\[
\mathbb{E} \left[ \| \tilde{x}_k^i - x_{k+1}^i \| \mid F_k \right] \leq \alpha_k r_i \frac{\mathbb{E} \left[ \| \tilde{d}_k^i \|_s \mid F_k \right]}{\sigma_x} \leq \alpha_k r_i \frac{C + \nu}{\sigma_x}.
\]  

(6.2.15)

Let us now write the iterates as follows

\[
x_{k+1}^i = \tilde{x}_k^i + \alpha_k \epsilon_k^i,
\]  

(6.2.16)

where \( \epsilon_k^i \) is an error term. From the relation obtained in (6.2.15) we have that \( \mathbb{E}[\| \epsilon_k^i \| \mid F_k] \) is bounded. Now we can apply the result of Theorem 6.1 in [7] to infer that

\[
\sum_{k=0}^{\infty} \alpha_k \| \tilde{x}_k^i - x_k^i \| < \infty \text{ for all } i.
\]

A similar analysis proves the assertion that \( \sum_{k=0}^{\infty} \alpha_k \| \tilde{\eta}_k^j - \eta_k^j \| < \infty \text{ for all } j. \)

We are now ready to prove our convergence result for the algorithm (6.2.8).
6.2.5 Analysis of the Algorithm

The steps involving the convergence proof are similar in nature to that of convergence proofs in Chapters 4 and 5. We use techniques from Lyapunov analysis to prove convergence of our algorithm. The first step involves establishing a descent inequality and the second step involves applying the Robbins-Seigmund convergence result to the inequality. For a choice of Lyapunov function as a sum of Bregman distance functions the following descent condition can be established. We also let

\[ z^i_k = \begin{bmatrix} x^i_k \\ \eta^i_k \end{bmatrix}, \quad \hat{z}_k = \begin{bmatrix} \hat{x}_k \\ \hat{\eta}_k \end{bmatrix}. \]  

(6.2.17)

Lemma 6.2.4. Let Assumptions 9 and 10 hold. Then, for algorithm (6.2.8) we have with probability 1 for any \( z^* = (x^*, \eta^*) \in X^* \times \{\eta^*\} \) and any \( k \geq 0 \),

\[
\sum_{i=1}^{m} \mathbb{E}_k [V_x(x^*, x^i_{k+1}) + V_\eta(\eta^*, \eta^i_{k+1})] \leq \sum_{i=1}^{m} (V_x(x^*, x^i_k) + V_\eta(\eta^*, \eta^i_k)) + \alpha_k^2 m \left( \frac{\bar{r}^2 (C + \nu)^2}{2\sigma_x} + \frac{(\frac{1}{m} + \bar{r})^2}{2\sigma_\eta} \right)
- \alpha_k \left( \sum_{i=1}^{m} r_i \eta_i^i + \hat{\eta}_k - \eta^* \right)
+ \alpha_k \bar{r} \left( C \sum_{j=1}^{m} \| \hat{x}_k - x^i_k \| + \sum_{j=1}^{m} \| \hat{\eta}_k - \eta^i_k \| \right),
\]

where \( \bar{r} = \max_{i \in V} r_i \).

Proof. By definition of the Bregman distance function we have

\[ V_x(x^*, x^i_{k+1}) - V_x(x^*, \tilde{x}^i_k) = w_x(\tilde{x}^i_k) - w_x(x^i_{k+1}) - \nabla w_x(x^i_{k+1})(x^* - x^i_{k+1}) + \nabla w_x(\tilde{x}^i_k)(x^* - \tilde{x}^i_k). \]

This implies

\[
V_x(x^*, x^i_{k+1}) - V_x(x^*, \tilde{x}^i_k) = \langle \nabla w_x(\tilde{x}^i_k) - \nabla w_x(x^i_{k+1}), x^* - x^i_{k+1} \rangle - V_x(x^i_{k+1}, \tilde{x}^i_k). \]  

(6.2.18)
Now, applying the optimality condition in (6.2.8), since \( x^* \in X \) we have

\[
\langle \alpha_k r_i \tilde{d}_k^i + \nabla \omega_x(x^i_{k+1}) - \nabla \omega_x(\tilde{x}^i_k), x^* - x^i_{k+1} \rangle \geq 0,
\]

where once again we have used the notation \( \tilde{d}_k^i = \nabla g_i(v_k^i) + \epsilon_k^i \). This implies that

\[
\langle \nabla \omega_x(\tilde{x}^i_k) - \nabla \omega_x(\tilde{x}_{k+1}^i), x^* - x^i_{k+1} \rangle \leq \langle \alpha_k r_i \tilde{d}_k^i, x^* - x^i_{k+1} \rangle.
\]

Upon substituting in (6.2.18) we obtain

\[
V_x(x^*, x^i_{k+1}) - V_x(x^*, \tilde{x}_k^i) \leq \alpha_k r_i \langle \tilde{d}_k^i, x^* - x_{k+1}^i \rangle - V_x(x^i_{k+1}, \tilde{x}_k^i) = \alpha_k r_i \langle \tilde{d}_k^i, x^* - \tilde{x}_k^i \rangle + \alpha_k r_i \langle \tilde{d}_k^i, \tilde{x}_k^i - x_{k+1}^i \rangle - V_x(x^i_{k+1}, \tilde{x}_k^i). (6.2.19)
\]

Due to the strong convexity of the Bregman function we have

\[
V_x(x^i_{k+1}, \tilde{x}_k^i) \geq \frac{\sigma_x}{2} \| x^i_{k+1} - \tilde{x}_k^i \|^2. \tag{6.2.20}
\]

Also, note that applying Hölder’s inequality we have \( \alpha_k r_i \langle \tilde{d}_k^i, x^* - x_{k+1}^i \rangle \leq \alpha_k r_i \| \tilde{d}_k^i \| \| x^* - x_{k+1}^i \| \). Using this fact, and applying the known inequality \( 2ab \leq a^2 + b^2 \) we get

\[
\alpha_k r_i \| \tilde{d}_k^i \| \| x^i_{k} - x^i_{k+1} \| = 2 \frac{\alpha_k r_i}{\sqrt{2}\sigma_x} \| \tilde{d}_k^i \| \sqrt{\frac{\sigma_x}{2}} \| x^i_{k} - x^i_{k+1} \| \leq \frac{\alpha_k r_i^2}{2\sigma_x} \| \tilde{d}_k^i \|^2 \| x^i_{k+1} - \tilde{x}_k^i \|^2 + \frac{\sigma_x}{2} \| x^i_{k} - \tilde{x}_k^i \|^2. \tag{6.2.21}
\]

Upon substituting (6.2.20) and (6.2.21) in (6.2.19) we arrive at

\[
V_x(x^*, x^i_{k+1}) - V_x(x^*, \tilde{x}_k^i) \leq \alpha_k r_i \langle \tilde{d}_k^i, x^* - \tilde{x}_k^i \rangle + \alpha_k r_i^2 \| \tilde{d}_k^i \|^2 \| x^i_{k+1} - \tilde{x}_k^i \|^2 + \alpha_k r_i^2 \frac{(C + \nu)^2}{2\sigma_x}.
\]

Taking the conditional expectation with respect to the history \( F_k \) we obtain

\[
\mathbb{E}_k \left[ V_x(x^*, x^i_{k+1}) \right] \leq V_x(x^*, \tilde{x}_k^i) - \alpha_k r_i \mathbb{E}_k [\tilde{d}_k^i], \tilde{x}_k^i - x^*] + \alpha_k r_i^2 \frac{(C + \nu)^2}{2\sigma_x}.
\]
Now, since $E_k[\tilde{d}_k^i] = \nabla_x g_i(v_k^i)$, we get

$$E_k \left[ V_x (x^*, x_{k+1}^i) \right] \leq V_x (x^*, \tilde{x}_k^i) - \alpha_k r_i (\nabla_x g_i(v_k^i), \tilde{x}_k^i - x^*) + \alpha_k^2 r_i^2 \frac{(C + \nu)^2}{2\sigma_x}. \quad (6.2.22)$$

Proceeding similarly, we can derive the following inequality for the iterates involving the min-max value estimate

$$E_k \left[ V_{\eta} (\eta^*, \eta_{k+1}^i) \right] \leq V_{\eta} (\eta^*, \tilde{\eta}_k^i) - \alpha_k \left( \frac{1}{m} + r_i \nabla_{\eta} g_i(v_k^i) \right) \cdot (\tilde{\eta}_k^i - \eta^*) + \alpha_k \frac{(\frac{1}{m} + r_i)^2}{2\sigma_{\eta}}, \quad (6.2.23)$$

where we have used the fact that $|\nabla_{\eta} g_i(v_k^i)| \leq 1$. By the convexity of $g_i$ and the subgradient property, we have

$$\nabla g_i(v_k^i)'(v_k^i - z^*) \geq g_i(v_k^i) - g_i(z^*) = g_i(v_k^i).$$

Here the equality follows by $g_i(z^*) = \max \{0, f_i(x^*) - \eta^*\}$ and relation $f_i(x^*) - \eta^* \leq 0$ which holds for all $i$ at any optimal point $x^*$ of the problem. Upon adding the inequalities (6.2.22) and (6.2.23) and using the convexity of $g_i$, we have for any $z^* \in X^* \times \{\eta^*\}$, any $k \geq 0$ and $i \in V$,

$$E_k \left[ V_x (x^*, x_{k+1}^i) + V_{\eta} (\eta^*, \eta_{k+1}^i) \right] \leq V_x (x^*, \tilde{x}_k^i) + V_{\eta} (\eta^*, \tilde{\eta}_k^i) - \alpha_k \frac{1}{m} (\tilde{\eta}_k^i - \eta^*) - \alpha_k r_i g_i(v_k^i) + \alpha_k^2 \left( \frac{r_i^2 (C + \nu)^2}{2\sigma_x} + \frac{(\frac{1}{m} + r_i)^2}{2\sigma_{\eta}} \right). \quad (6.2.24)$$

Now, by Assumption 11 on the convexity of the Bregman function and the doubly stochasticity of the weight matrix $W_k$ (Assumption 9-b) we have

$$\sum_{i=1}^m V_x (x^*, x_{k+1}^i) = \sum_{i=1}^m V_x \left( x^*, \sum_{j=1}^m [W_k]_{ij} x_{k}^j \right) \leq \sum_{i=1}^m \sum_{j=1}^m [W_k]_{ij} V_x (x^*, x_{k}^j) = \sum_{j=1}^m V_x (x^*, x_{k}^j).$$

Similarly,

$$\sum_{i=1}^m V_{\eta} (\eta^*, \eta_{k+1}^i) = \sum_{i=1}^m V_{\eta} \left( \eta^*, \sum_{j=1}^m [W_k]_{ij} \eta_{k}^j \right) \leq \sum_{i=1}^m \sum_{j=1}^m [W_k]_{ij} V_{\eta} (\eta^*, \eta_{k}^j) = \sum_{j=1}^m V_{\eta} (\eta^*, \eta_{k}^j).$$
Thus, summing inequalities in (6.2.24) over \( i \in V \) we obtain for any \( z^* \in X^* \times \{ \eta^* \} \) and all \( k \geq 0 \),

\[
\sum_{i=1}^{m} \mathbb{E}_k \left[ V_x(x^*, x_{k+1}^i) + V_\eta(\eta^*, \eta_{k+1}^i) \right] \leq \sum_{i=1}^{m} \left( V_x(x^*, x_k^i) + V_\eta(\eta^*, \eta_k^i) \right) - \alpha_k \frac{1}{m} \sum_{i=1}^{m} (\tilde{\eta}_k^i - \eta^*) \\
- \alpha_k \sum_{i=1}^{m} r_i g_i(v_k^i) + \alpha_k^2 m \left( \frac{\bar{r}^2(C + \nu)^2}{2\sigma_x} + \frac{(1/m + \bar{r})^2}{2\sigma_\eta} \right), \tag{6.2.25}
\]

where \( \bar{r} = \max_{i \in V} r_i \). Now, recalling the definition of \( \tilde{\eta}_k^i \) and the doubly stochasticity of the matrix \( W_k \), we have

\[
\sum_{i=1}^{m} \tilde{\eta}_k^i = \sum_{i=1}^{m} \sum_{j=1}^{m} [W_k]_{ij} \eta_k^j = \sum_{j=1}^{m} \eta_k^j = m\tilde{\eta}_k,
\]

which when substituted in (6.2.26) yields

\[
\sum_{i=1}^{m} \mathbb{E}_k \left[ V_x(x^*, x_{k+1}^i) + V_\eta(\eta^*, \eta_{k+1}^i) \right] \leq \sum_{i=1}^{m} \left( V_x(x^*, x_k^i) + V_\eta(\eta^*, \eta_k^i) \right) - \alpha_k (\tilde{\eta}_k - \eta^*) \\
- \sum_{i=1}^{m} \alpha_k r_i g_i(v_k^i) + \alpha_k^2 m \left( \frac{\bar{r}^2(C + \nu)^2}{2\sigma_x} + \frac{(1/m + \bar{r})^2}{2\sigma_\eta} \right), \tag{6.2.26}
\]

Next, recalling \( z_k \) from (6.2.17), upon adding and subtracting the term \( \alpha_k \sum_{i=1}^{m} r_i g_i(z_k) \) in (6.2.26), we get

\[
\sum_{i=1}^{m} \mathbb{E}_k \left[ V_x(x^*, x_{k+1}^i) + V_\eta(\eta^*, \eta_{k+1}^i) \right] \leq \sum_{i=1}^{m} \left( V_x(x^*, x_k^i) + V_\eta(\eta^*, \eta_k^i) \right) \\
+ \alpha_k^2 m \left( \frac{\bar{r}^2(C + \nu)^2}{2\sigma_x} + \frac{(1/m + \bar{r})^2}{2\sigma_\eta} \right) - \alpha_k \left( \sum_{i=1}^{m} r_i g_i(z_k) + \tilde{\eta}_k - \eta^* \right) + \alpha_k \sum_{i=1}^{m} r_i \left| g_i(z_k) - g_i(v_k^i) \right|. \tag{6.2.27}
\]

Next, consider the term \( \sum_{i=1}^{m} r_i \left| g_i(z_k) - g_i(v_k^i) \right| \). By the definition of \( g_i \) and relation \( \max\{a, 0\} -
max\{b,0\} \leq |a - b| \text{ valid for any two scalars } a \text{ and } b, \text{ we have the following:}

\[ |g_i(\hat{z}_k) - g_i(v^i_k)| \leq |f_i(\hat{x}_k) - f_i(\hat{x}^i_k) - \hat{\eta}_k + \eta^i_k| \]
\[ \leq |f_i(\hat{x}_k) - f_i(\hat{x}^i_k)| + |\hat{\eta}_k - \eta^i_k| \]
\[ \leq C \|\hat{x}_k - \hat{x}^i_k\| + |\hat{\eta}_k - \eta^i_k| \]
\[ \leq \sum_{j=1}^{m}|W_k|_{ij} \left[ C \|\hat{x}_k - x^i_k\| + |\hat{\eta}_k - \eta^i_k| \right], \]

where in the first inequality we use the definition of \(v^i_k\), while in the last inequality we use the subgradient boundedness assumption for \(f_i\) and the definition of the variables \(\hat{x}^i_k\) and \(\eta^i_k\) in (6.2.7).

Therefore, by using the doubly stochasticity of \(W_k\), we obtain

\[ \sum_{i=1}^{m} r_i |g_i(\hat{z}_k) - g_i(v^i_k)| \leq \bar{r}C \sum_{j=1}^{m}\|\hat{x}_k - x^i_k\| + \bar{r} \sum_{j=1}^{m}|\hat{\eta}_k - \eta^i_k|. \]

Now, substituting back in (6.2.27) we get the desired result.

We are now ready to prove our main convergence result. The result essentially states that under suitable conditions on the step size \(\alpha_k\), all the agents’ estimates converge to a common optimal point. Moreover, the agents’ estimates of the min-max value also converge to the optimal value of the problem.

**Theorem 6.2.5.** Let Assumptions 9 and 10 hold. Moreover, let the Bregman distances in the algorithm (6.2.8) be chosen to satisfy Assumption 11. Assume that min-max problem (6.1.1) has a nonempty optimal solution set \(X^* \subseteq X\). Let the step sizes satisfy \(\sum_{k=0}^{\infty} \alpha_k = \infty\) and \(\sum_{k=0}^{\infty} \alpha_k^2 < \infty\).

Then for all \(i \in V\), the agents’ iterates \(x^i_k\) and \(\eta^i_k\) generated by algorithm (6.2.8) are such that with probability 1: the decision variables \(x^i_k\) converge to a common optimal (random) point \(x^* \in X^*\) and the estimates \(\eta^i_k\) converge to the optimal value \(\eta^*\) of the min-max problem.

**Proof.** Our analysis is based on applying the Robbins-Siegmund result from Lemma 2.3.2 to the inequality derived in Lemma 6.2.4. By our assumption on the step sizes we have \(\sum_{k=0}^{\infty} \alpha_k^2 < \infty\), which trivially implies that \(\alpha_k^2 m \left( \frac{\nu^2(C+\nu)^2}{2\sigma_x} + \frac{1}{2\sigma_\eta} \left( \frac{1+\nu}{\bar{r}} \right)^2 \right) < \infty\). Furthermore, by virtue of Lemma 6.2.3 we
have
\[
\sum_{k=0}^{\infty} \alpha_k \bar{r} \sum_{j=1}^{m} \left( C \left\| \hat{x}_k - x^j_k \right\| + |\hat{\eta}_k - \eta^j_k| \right) < \infty. \tag{6.2.28}
\]
In addition, by virtue of Lemma 6.2.2 we have that
\[
\sum_{i=1}^{m} r_i g_i(\hat{z}_k) + \hat{\eta}_k - \eta^* \geq 0 \quad \text{for all } k \geq 0.
\]
Thus, we can apply Lemma 2.3.2 and infer that with probability 1,
\[
\sum_{i=1}^{m} \left( V_x(x^*, x^j_k) + V_\eta(\eta^*, \eta^j_k) \right) \quad \text{converges for every } (x^*, \eta^*) \in X^* \times \{ \eta^* \}
\]
and
\[
\sum_{k=0}^{\infty} \alpha_k \left( \sum_{i=1}^{m} r_i g_i(\hat{z}_k) + \hat{\eta}_k - \eta^* \right) < \infty.
\]
Now, since \( \sum_{k=0}^{\infty} \alpha_k = \infty \), there exists a subsequence indexed as \{k_\ell\} such that the following hold with probability 1,
\[
\lim_{\ell \to \infty} \left( \sum_{i=1}^{m} r_i g_i(\hat{z}_{k_\ell}) + \hat{\eta}_{k_\ell} - \eta^* \right) = 0,
\]
\[
\lim_{\ell \to \infty} \left\| \hat{x}_{k_\ell} - x^j_{k_\ell} \right\| = 0 \quad \text{and} \quad \lim_{\ell \to \infty} |\hat{\eta}_{k_\ell} - \eta^j_{k_\ell}| = 0 \quad \text{for all } j. \tag{6.2.29}
\]
Since \( \sum_{i=1}^{m} \left( V_x(x^*, x^j_k) + V_\eta(\eta^*, \eta^j_k) \right) \) converges for every \((x^*, \eta^*) \in X^* \times \{ \eta^* \}\) we have that the sequence \( \sum_{i=1}^{m} \left( V_x(x^*, x^j_k) + V_\eta(\eta^*, \eta^j_k) \right) \) is bounded for every \((x^*, \eta^*) \in X^* \times \{ \eta^* \}\). Note that we have from Assumption 11 on the convexity of the chosen Bregman function and the strong convexity property of the Bregman functions
\[
\frac{1}{m} \sum_{i=1}^{m} \left( V_x(x^*, x^j_k) + V_\eta(\eta^*, \eta^j_k) \right) \geq V_x(x^*, \hat{x}_k) + V_\eta(\eta^*, \hat{\eta}_k)
\]
\[
\geq \frac{\sigma_x}{2} \left\| x^* - \hat{x}_k \right\|^2 + \frac{\sigma_\eta}{2} \left\| \eta^* - \hat{\eta}_k \right\|^2.
\]
This implies that the sequences \{\hat{x}_k\} and \{\hat{\eta}_k\} are also bounded. Thus, along a further subsequence, which without loss of generality we can let it be indexed by the same index set \{k_\ell, \ell = 1, 2, \ldots\}, with probability 1 we have \( \lim_{\ell \to \infty} \hat{x}_{k_\ell} = \hat{x} \) and \( \lim_{\ell \to \infty} \hat{\eta}_{k_\ell} = \hat{\eta} \). Moreover, with probability 1 the
limit points satisfy
\[ \sum_{i=1}^{m} r_i g_i(\bar{x}) + \bar{\eta} - \eta^* = 0. \] (6.2.30)

From Lemma 6.2.2 it follows that \( \bar{x} \) is an optimal solution to the min-max problem with probability 1 and \( \bar{\eta} = \eta^* \). In view of (6.2.30), with probability 1, we further have \( x_{k_t}^j \rightarrow \bar{x} \) and \( \eta_{k_t}^j \rightarrow \bar{\eta} \) for all \( j \). Let \( \tilde{z} = (\bar{x}, \bar{\eta}) \), then we have \( \lim_{\ell \rightarrow \infty} \| z_{k_\ell}^j - \tilde{z} \| = 0 \) with probability 1 for all \( j \), which implies that \( \lim_{\ell \rightarrow \infty} \sum_{i=1}^{m} \left( V_x(\bar{x}, x_{k_\ell}^i) + V_\eta(\bar{\eta}, \eta_{k_\ell}^i) \right) = 0 \). However, we have shown that the sequence \( \sum_{i=1}^{m} \left( V_x(x^*, x_{k}^i) + V_\eta(\eta^*, \eta_{k}^i) \right) \) converges with probability 1 for any \((x^*, \eta^*) \in X^* \times \{\eta^*\}\), which in view of \((\bar{x}, \bar{\eta}) \in X^* \times \{\eta^*\}\), implies that \( \lim_{k \rightarrow \infty} \sum_{i=1}^{m} \left( V_x(\bar{x}, x_{k}^i) + V_\eta(\bar{\eta}, \eta_{k}^i) \right) = 0 \) with probability 1 for all \( j \). Because of the strong convexity of Bregman functions we have
\[
\sum_{i=1}^{m} \left( V_x(\bar{x}, x_{k}^i) + V_\eta(\bar{\eta}, \eta_{k}^i) \right) \geq \sum_{i=1}^{m} \left( \frac{\sigma_x}{2} \| \bar{x} - x_{k}^i \|^2 + \frac{\sigma_\eta}{2} \| \bar{\eta} - \eta_{k}^i \|^2 \right),
\]
which implies that \( x_{k}^j \rightarrow \bar{x} \) and \( \eta_{k}^j \rightarrow \bar{\eta} \) with probability 1 for all \( j \). \( \Box \)

### 6.3 Primal-Dual Algorithm

In this section we present a distributed primal-dual algorithm which is based on the notion of generalized distance. The primal-dual algorithm that we study is motivated by the classical work of Arrow-Hurwicz-Uzawa [63]. Recently the authors in [65] studied primal-dual algorithms for approximate saddle-points by considering standard Euclidean norm. The use of Bregman distance for a saddle-point problem was studied in [26].

As in the previous section we work with the strongly convex function \( \omega_x : X \rightarrow \mathbb{R} \), with the convexity parameter \( \sigma_x \), and the strongly convex function \( \omega_\eta : \mathbb{R} \rightarrow \mathbb{R} \) with the convexity parameter \( \sigma_\eta \). In addition, we introduce a strongly convex function \( \omega_\mu : \mathbb{R} \rightarrow \mathbb{R} \), with convexity parameter \( \sigma_\mu \). Let us denote the Bregman distance functions generated by these strongly convex functions by \( V_x(\cdot, \cdot), V_\eta(\cdot, \cdot), \) and \( V_\mu(\cdot, \cdot) \), respectively. Recall that the Lagrangian of the min-max problem under consideration is given as

\[ L(x, \eta, \mu) = \left( 1 - \sum_{i=1}^{m} \mu_i \right) \eta + \sum_{i=1}^{m} \mu_i f_i(x). \] (6.3.1)
Then the proposed primal-dual algorithm is as follows.

\[
\begin{align*}
  x^{i+1}_{k+1} &= \arg\min_{y \in X} \left[ \alpha_k \mu_k^i \langle d_k^i + \epsilon_k^i, y \rangle + V_x(y, \tilde{x}_k^i) \right] \\
  \eta^{i+1}_{k+1} &= \arg\min_{s \in D} \left[ \alpha_k (1/m - \mu_k^i) s + V_\eta(s, \tilde{\eta}_k^i) \right] \\
  \mu^{i+1}_{k+1} &= \arg\min_{\zeta \in I} \left[ \alpha_k \left( \tilde{\eta}_k^i - f_i(\tilde{x}_k^i) \right) \zeta + V_{\mu}(\zeta, \mu_k^i) \right],
\end{align*}
\]

(6.3.2)

where \( d_k^i \) is a subgradient of \( f_i(x) \) evaluated at \( \tilde{x}_k^i \). The set \( D \subset \mathbb{R} \) is a compact set which serves as an estimate of the min-max optimal value and the set \( I \) is the unit interval \([0, 1]\). Here, once again we have used the notation \( \tilde{x}_k^i \) and \( \tilde{\eta}_k^i \) as defined in (6.2.7).

### 6.3.1 Analysis of the Algorithm

Our line of analysis is similar to the analysis of the Bregman distance based algorithm using exact penalty approach (6.2.8). First we state a lemma, similar to Lemma 6.2.3 which states that for the primal-dual algorithm the local iterates converge to their respective mean trajectory. Let us define the modified \( \sigma \)-field as

\[
F_k = \{ x^i_t, \eta^i_t, \mu^i_t, i \in V, t = 0, 1, \ldots, k \} \quad \text{for } k \geq 0.
\]

In what follows the conditional expectation \( \mathbb{E}_k[.] = \mathbb{E}[. \mid F_k] \) is with respect to the modified \( \sigma \)-field.

**Lemma 6.3.1.** Let Assumptions 9 and 10 hold, and let the step sizes satisfy \( \sum_{k=0}^\infty \alpha_k^2 < \infty \). Then, for the iterates of the algorithm (6.3.2) we have with probability 1 for all \( j \in V \),

\[
\sum_{k=0}^\infty \alpha_k \| \hat{x}_k - \hat{x}_k^j \| < \infty, \quad \sum_{k=0}^\infty \alpha_k |\hat{\eta}_k - \eta_k^j| < \infty.
\]

**Proof.** The proof is similar to the proof of Lemma 6.2.3 and makes use of the fact that \( \mu_k^i \in I \) and \( \eta_k^i \in D \) belong to bounded sets for all \( i \). \qed

Next, we establish a descent condition similar to Lemma 6.2.4 for a chosen Lyapunov function.
Let us denote the composite function
\[
V(z^*, \mu^*, z_k, \mu_k) = \sum_{i=1}^{m} \left[ V(x^*, x^i_k) + V(\eta^*, \eta^i_k) + V(\mu^*, \mu^i_k) \right],
\]  
(6.3.3)
where we have denoted \(z_k = (x^1_k, \ldots, x^m_k, \eta^1_k, \ldots, \eta^m_k)\) and \(\mu_k = (\mu^1_k, \ldots, \mu^m_k)\). The vectors \(z^* = (x^*, \eta^*)\) is an optimal decision variable and optimal min-max value pair. Then we can establish the following descent condition.

**Lemma 6.3.2.** Let Assumptions 9 and 10 hold. Further, let the Bregman distances are chosen to satisfy the Assumption 11. Then, for the iterates of the algorithm (6.3.2) the following descent condition holds

\[
\mathbb{E}_k \left[ V(z^*, \mu^*, z_{k+1}, \mu_{k+1}) \right] \leq V(z^*, \mu^*, z_k, \mu_k) + \alpha_k^2 \left( \frac{m(C + \nu)^2}{2\sigma_x} + \sum_{i=1}^{m} \frac{(1/m - \mu^i_k)^2}{2\sigma_\mu} \right) + \alpha_k \left[ L(\hat{z}_k, \mu^*) - L(z^*, \mu_k) \right] + 2C\alpha_k \sum_{j=1}^{m} \left\| \hat{x}_k - x^j_k \right\| + \alpha_k^2 \left( \frac{2 - 1/m}{m} \sum_{j=1}^{m} \left\| \hat{\eta}_k - \eta^j_k \right\| \right).
\]

*Proof.* The proof of this lemma closely resembles that of Lemma 6.2.4. Using the optimality conditions in (6.3.2) and proceeding as in the proof of Lemma 6.2.4, we arrive at

\[
\mathbb{E}_k \left[ V_x(x^*, x^i_{k+1}) \right] \leq V_x(x^*, \tilde{x}^i_k) - \alpha_k \mu^i_k (\mathbb{E}_k [\tilde{d}^i_k], \tilde{x}^i_k - x^*) + \alpha_k^2 \left( \frac{C + \nu)^2}{2\sigma_x} \right).
\]

Now, since \(\mathbb{E}_k [\tilde{d}^i_k] = d^i_k\), and \(d^i_k\) is a subgradient of \(f_i(x)\) at \(\tilde{x}_k^i\) we get

\[
\mathbb{E}_k \left[ V_x(x^*, x^i_{k+1}) \right] \leq V_x(x^*, \tilde{x}^i_k) - \alpha_k \mu^i_k (f_i(\tilde{x}^i_k) - f_i(x^*)) + \alpha_k^2 \left( \frac{C + \nu)^2}{2\sigma_x} \right). \tag{6.3.4}
\]

A similar analysis gives us the inequality for the minmax estimate iterates \(\{\eta^i_k\}\)

\[
V_\eta(\eta^*, \eta^i_{k+1}) \leq V_\eta(\eta^*, \tilde{\eta}^i_k) - \alpha_k (1/m - \mu^i_k) (\tilde{\eta}^i_k - \eta^*) + \alpha_k^2 \left( \frac{1/m - \mu^i_k)^2}{2\sigma_\eta} \right) \tag{6.3.5}
\]

103
and the inequality for the dual variable iterates \( \{ \mu_k^i \} \)

\[
V_\mu(\mu^i_{k+1}, \mu^i_k) \leq V_\mu(\mu^i, \mu^i_k) + \alpha_k(f_i(\tilde{x}^i_k) - \tilde{\eta}^i_k)(\mu^i_k - \mu^i_k) + \alpha_k^2 \frac{(f_i(\tilde{x}^i_k) - \tilde{\eta}^i_k)^2}{2\sigma_\mu}. \tag{6.3.6}
\]

Summing up equations (6.3.4) and (6.3.5), and summing the result over the index \( i \) we have

\[
\sum_{i=1}^m \mathbb{E}_k \left[ V_x(x^*, x_{k+1}^i) + V_\eta(\eta^*, \eta_{k+1}^i) \right] \leq \sum_{i=1}^m \left[ V_x(x^*, \tilde{x}^i_k) + V_\eta(\eta^*, \tilde{\eta}^i_k) \right] + \alpha_k^2 \left( \frac{(C + \nu)^2}{2\sigma_x} + \sum_{i=1}^m \frac{(1/m - \mu_k^i)^2}{2\sigma_\eta} \right) \\
- \alpha_k \sum_{i=1}^m \left[ \mu_k^i(f_i(\tilde{x}^i_k) - f_i(x^*)) + (1/m - \mu_k^i)(\tilde{\eta}^i_k - \eta^*) \right].
\]

On adding and subtracting the term \( \alpha_k \sum_{i=1}^m (\mu_k^i f_i(\tilde{x}^i_k) + (1/m - \mu_k^i) \tilde{\eta}^i_k) \), we get

\[
\sum_{i=1}^m \mathbb{E}_k \left[ V_x(x^*, x_{k+1}^i) + V_\eta(\eta^*, \eta_{k+1}^i) \right] \leq \sum_{i=1}^m \left[ V_x(x^*, \tilde{x}^i_k) + V_\eta(\eta^*, \tilde{\eta}^i_k) \right] + \alpha_k^2 K \\
- \alpha_k \left[ \sum_{i=1}^m (\mu_k^i f_i(\tilde{x}^i_k) - f_i(x^*)) + (1 - \sum_{i=1}^m \mu_k^i)(\tilde{\eta} - \eta^*) \right] \\
+ \alpha_k \left[ \sum_{i=1}^m \mu_k^i f_i(\tilde{x}^i_k) - f_i(x^*) \right] + \alpha_k \sum_{i=1}^m \left( \frac{(1/m - \mu_k^i)^2}{2\sigma_\eta} \right) \cdot \eta^i_k - \tilde{\eta}^i_k, 
\]

where for ease of notation we have used \( K = \left( \frac{(C + \nu)^2}{2\sigma_x} + \sum_{i=1}^m \frac{(1/m - \mu_k^i)^2}{2\sigma_\eta} \right) \). Using the definition of the Lagrangian in (6.3.1) this can be equivalently written as

\[
\sum_{i=1}^m \mathbb{E}_k \left[ V_x(x^*, x_{k+1}^i) + V_\eta(\eta^*, \eta_{k+1}^i) \right] \leq \sum_{i=1}^m \left[ V_x(x^*, \tilde{x}^i_k) + V_\eta(\eta^*, \tilde{\eta}^i_k) \right] + \alpha_k^2 K \\
- \alpha_k \left[ L(\hat{x}_k, \mu_k) - L(z^*, \mu_k) \right] \\
+ \alpha_k \left[ \sum_{i=1}^m \mu_k^i f_i(\tilde{x}^i_k) - f_i(x^*) \right] + \sum_{i=1}^m \left( \frac{(1/m - \mu_k^i)(\tilde{\eta}^i_k - \eta^i_k)}{2\sigma_\eta} \right).
\]

Since under our assumption the Bregman distance functions are convex in the second argument, using the doubly stochasticity of the weight matrix, the above relation can be equivalently written
Hence, on summing over index $i$ as

\[
\sum_{i=1}^{m} \mathbb{E}_k \left[ V_k (x^*, x^i_{k+1}) + V_\eta (\eta^*, \eta^i_{k+1}) \right] \leq \sum_{i=1}^{m} \left[ V_k (x^*, x^i_k) + V_\eta (\eta^*, \eta^i_k) \right] + \alpha_k^2 K
\]

\[
- \alpha_k \left[ L (\dot{x}_k, \mu_k) - L (x^*, \mu_k) \right]
\]

\[
+ \alpha_k \left[ \sum_{i=1}^{m} \mu_k^i (f_i (\ddot{x}_k^i) - f_i (\ddot{x}_k)) + \sum_{i=1}^{m} (1/m - \mu_k^i) (\ddot{\eta}_k - \ddot{\eta}_k^i) \right],
\]

this is similar to the proof of Lemma 6.2.4. Let us now consider the equation (6.3.6). We have

\[
f_i (\ddot{x}_k^i) - \ddot{\eta}_k^i = (f_i (\ddot{x}_k) - \ddot{\eta}_k) + [f_i (\ddot{x}_k^i) - f_i (\ddot{x}_k) - \ddot{\eta}_k^i + \ddot{\eta}_k].
\]

Thus, the term $\alpha_k (f_i (\ddot{x}_k^i) - \ddot{\eta}_k^i) (\mu_k^i - \mu_i^*)$ in (6.3.6) can be equivalently written as

\[
\alpha_k (f_i (\ddot{x}_k^i) - \ddot{\eta}_k^i) (\mu_k^i - \mu_i^*) = \alpha_k (\mu_k^i - \mu_i^*) (f_i (\ddot{x}_k) - \ddot{\eta}_k) + \alpha_k (\mu_k^i - \mu_i^*) \left[ f_i (\ddot{x}_k) - f_i (\ddot{x}_k) - \ddot{\eta}_k^i + \ddot{\eta}_k \right].
\]

Further on adding and subtracting $1/m \alpha_k \ddot{\eta}_k$ in the term $\alpha_k (\mu_k^i - \mu_i^*) (f_i (\ddot{x}_k) - \ddot{\eta}_k)$ above we arrive at,

\[
\alpha_k (\mu_k^i - \mu_i^*) (f_i (\ddot{x}_k) - \ddot{\eta}_k) = \alpha_k \left( \mu_k^i f_i (\ddot{x}_k) + (1/m - \mu_k^i) \ddot{\eta}_k \right) - \alpha_k \left( \mu_i^* f_i (\ddot{x}_k) + (1/m - \mu_i^*) \ddot{\eta}_k \right).
\]

Hence, on summing over index $i$ and using the definition of the Lagrangian (6.3.1) we get

\[
\sum_{i=1}^{m} \alpha_k (\mu_k^i - \mu_i^*) (f_i (\ddot{x}_k) - \ddot{\eta}_k) = \alpha_k (L (\dot{x}_k, \mu_k) - L (\dot{x}_*, \mu^*)).
\]

On summing up equation (6.3.6) over the index $i$ and using (6.3.8) and (6.3.9) we arrive at

\[
\sum_{i=1}^{m} V_\mu (\mu_i^*, \mu_{i,k+1}^*) \leq \sum_{i=1}^{m} V_\mu (\mu_i^*, \mu_k^i) + \alpha_k (L (\dot{x}_k, \mu_k) - L (\dot{x}_*, \mu^*))
\]

\[
+ \alpha_k \sum_{i=1}^{m} (\mu_k^i - \mu_i^*) \left[ f_i (\ddot{x}_k) - f_i (\ddot{x}_k) - \ddot{\eta}_k^i + \ddot{\eta}_k \right] + \alpha_k^2 \sum_{i=1}^{m} \frac{(f_i (\ddot{x}_k^i) - \ddot{\eta}_k^i)^2}{2 \sigma_\mu}.\]

We now establish some simple bounds which simplifies our analysis. Let us consider the term

\[
\alpha_k \left[ \sum_{i=1}^{m} \mu_k^i (f_i (\ddot{x}_k) - f_i (\ddot{x}_k)) + \sum_{i=1}^{m} (1/m - \mu_k^i) (\ddot{\eta}_k - \ddot{\eta}_k^i) \right]
\]

105
in (6.3.7). The first term here can be bound as
\[
\alpha_k \sum_{i=1}^{m} \mu_k^i (f_i(\tilde{x}_k^i) - f_i(\bar{x}_k^i)) \leq C \alpha_k \sum_{i=1}^{m} \sum_{j=1}^{m} [W_k]_{ij} \| \tilde{x}_k - x_k^j \| \leq C \alpha_k \sum_{j=1}^{m} \| \tilde{x}_k - x_k^j \|,
\]
where \( C \) is a uniform bound on the Lipschitz constants for the functions \( f_i \). The second term is bounded as
\[
\alpha_k \sum_{i=1}^{m} (1/m - \mu_k^i)(\hat{\eta}_k - \hat{\eta}_k^i) \leq (1 - 1/m) \alpha_k \sum_{i=1}^{m} \| \hat{\eta}_k - \hat{\eta}_k^i \| \leq (1 - 1/m) \alpha_k \sum_{i=1}^{m} \sum_{j=1}^{m} [W_k]_{ij} \| \hat{\eta}_k - \eta_k^i \| \leq (1 - 1/m) \alpha_k \sum_{j=1}^{m} \| \hat{\eta}_k - \eta_k^i \|,
\]
where we have used the fact that \( \mu_k^i \in [0,1] \) and the double stochasticity of the weight matrix. A similar bound can be derived for the error term in (6.3.10) as
\[
\alpha_k \sum_{i=1}^{m} (\mu_k^i - \mu_i^*) [f_i(\bar{x}_k^i) - f_i(\tilde{x}_k^i) - \hat{\eta}_k + \tilde{\eta}_k] \leq \alpha_k \sum_{i=1}^{m} |f_i(\bar{x}_k^i) - f_i(\tilde{x}_k^i)| + \alpha_k \sum_{i=1}^{m} \| \hat{\eta}_k - \tilde{\eta}_k^i \| \leq C \alpha_k \sum_{j=1}^{m} \| \tilde{x}_k - x_k^j \| + \alpha_k \sum_{j=1}^{m} \| \hat{\eta}_k - \eta_k^j \|,
\]
where once again we use the fact that both \( \mu_k^i \in [0,1] \) and \( \mu_i^* \in [0,1] \). Thus, we have the following coupled set of equations
\[
\sum_{i=1}^{m} E_k \left[ V(x^*, x_{k+1}^i) + V_\eta(\eta^*, \eta_{k+1}^i) \right] \leq \sum_{i=1}^{m} \left[ V(x^*, x_k^i) + V_\eta(\eta^*, \eta_k^i) \right] + \alpha_k^2 K
- \alpha_k \left[ L(\tilde{z}_k, \mu_k) - L(z^*, \mu_k) \right] + C \alpha_k \sum_{j=1}^{m} \| \tilde{x}_k - x_k^j \|
+ (1 - 1/m) \alpha_k \sum_{j=1}^{m} \| \hat{\eta}_k - \eta_k^j \|, \quad (6.3.11)
\]
106
and
\[
\sum_{i=1}^{m} V_{\mu}(\mu^*_{i}, \mu_{k+1}) \leq \sum_{i=1}^{m} V_{\mu}(\mu^*_{i}, \mu_{k}) + \alpha_k (L(\tilde{x}_k, \mu_k) - L(\tilde{z}_k, \mu^*))
\]
\[
+ C\alpha_k \sum_{j=1}^{m} \|\tilde{x}_k - x^k_j\| + \alpha_k \sum_{j=1}^{m} \|\tilde{\eta}_k - \eta^k_j\| + \alpha_k \sum_{i=1}^{m} \frac{(f_i(\tilde{x}_k^i) - \tilde{\eta}_k^i)^2}{2\sigma_{\mu}}. \tag{6.3.12}
\]

We get the desired result by adding the equations (6.3.11), and (6.3.12). \hfill \square

We next prove our asymptotic convergence result for the algorithm (6.3.2).

**Theorem 6.3.3.** Let us assume that the Lagrangian (6.3.1) admits a unique saddle-point \((z^*, \mu^*)\). Further, let Assumptions 9 and 10 hold and the Bregman distances \(V_x(\cdot, \cdot)\) and \(V_\eta(\cdot, \cdot)\) in the algorithm (6.3.2) are chosen to satisfy Assumption 11. Moreover assume that the convex constraint sets \(X\) and \(D\) are compact and that min-max problem (6.1.1) has an optimal solution set \(X^* \subset X\). Let the step sizes satisfy \(\sum_{k=0}^{\infty} \alpha_k = \infty\) and \(\sum_{k=0}^{\infty} \alpha^2_k < \infty\). Then, the agents’ iterates \(x^i_k\) and \(\eta^i_k\) generated by algorithm (6.3.2) are such that with probability 1: the decision variables \(x^*_k\) converge to the common optimal point \(x^* \in X^*\) for all \(i\) and the estimates \(\eta^*_k\) converge to the optimal value \(\eta^*\) of the min-max problem. Also the agents’ dual iterates \(\mu^i_k\) converge to the corresponding optimal dual variable \(\mu^*_i\).

**Proof.** Our proof follows our earlier methodology of applying the Robbins-Siegmund result from Lemma 2.3.2 to the inequality derived in Lemma 6.3.2. By our assumption on the step sizes we have \(\sum_{k=0}^{\infty} \alpha^2_k < \infty\), which trivially implies that \(\sum_{k=0}^{\infty} \alpha_k^2 m (C + \nu)^2 \sigma_x < \infty\). Since, the projection step in algorithm (6.3.2) constrains the dual variables \(\mu^i_k\) to lie in the interval \([0, 1]\), this implies that \(\sum_{k=0}^{\infty} \alpha^2_k \sum_{i=1}^{m} \frac{(1/m - \mu^i_k)^2}{2\sigma_\mu} < \infty\) almost surely. Moreover, as the constraint set \(X\) and the set \(D\) are compact and convex this implies that the term \(f_i(\tilde{x}_k^i) - \tilde{\eta}_k^i\) is bounded for all \(i\) and \(k\) which implies that \(\sum_{k=0}^{\infty} \alpha^2_k \sum_{i=1}^{m} \frac{(f_i(\tilde{x}_k^i) - \tilde{\eta}_k^i)^2}{2\sigma_\mu} < \infty\) almost surely. By virtue of Lemma 6.3.1 we have
\[
\sum_{k=0}^{\infty} \alpha_k \left( 2C \sum_{j=1}^{m} \|\tilde{x}_k - x^k_j\| + (2 - 1/m) \|\tilde{\eta}_k - \eta^k_j\| \right) < \infty. \tag{6.3.13}
\]
Also by the Saddle-Point Theorem 6.2.1 we have that \(L(\tilde{z}_k, \mu^*) - L(z^*, \mu_k) \geq 0\). Thus, we can apply Lemma 2.3.2 and infer that with probability 1, \(V(z^*, \mu^*, \tilde{z}_k, \mu_k)\) converges for the optimal
decision vector $z^* = (x^*, \eta^*) \in X^* \times \{\eta^*\}$, and optimal dual variable $\mu^*$, and

$$\sum_{k=0}^{\infty} \alpha_k (L(\hat{z}_k, \mu^*) - L(z^*, \mu_k)) < \infty.$$ 

Now, since $\sum_{k=0}^{\infty} \alpha_k = \infty$, without loss of generality we can use Cantor diagonalization-type argument to select a subsequence such that the following hold with probability 1,

$$\lim_{\ell \to \infty} (L(\hat{z}_{k_{\ell}}, \mu^*) - L(z^*, \mu_{k_{\ell}})) = 0,$$

$$\lim_{\ell \to \infty} \|\hat{x}_{k_{\ell}} - x^i_{k_{\ell}}\| = 0 \quad \text{and} \quad \lim_{\ell \to \infty} |\hat{\eta}_{k_{\ell}} - \eta^i_{k_{\ell}}| = 0 \quad \forall j. \quad (6.3.14)$$

Due to the compactness of the constraint set $X$, the set $D$ and the interval $I$ we can choose convergent subsequences $\{\hat{z}_{k_{\ell}}\}$ and $\{\mu_{k_{\ell}}\}$ with probability 1, which without loss of generality we can let it be indexed by the same index set $\{k_{\ell}, \ell = 1, 2, \ldots\}$, such that $\lim_{\ell \to \infty} \hat{z}_{k_{\ell}} = \bar{z}$ and $\lim_{\ell \to \infty} \mu_{k_{\ell}} = \bar{\mu}$, where $\bar{z} = (\bar{x}, \bar{\eta}) \in X \times D$. Moreover we have that the limit points satisfy

$$L(\bar{z}, \mu^*) - L(z^*, \bar{\mu}) = 0.$$

Note that the saddle-point property of the solution $(z^*, \mu^*)$, states that

$$L(\bar{z}, \mu^*) \geq L(z^*, \mu^*) \geq L(z^*, \bar{\mu}),$$

which implies that

$$L(\bar{z}, \mu^*) = L(z^*, \mu^*) = L(z^*, \bar{\mu}).$$

This in turn implies that due to the uniqueness of the saddle-point of the Lagrangin we have $\bar{z} = z^*$ and $\bar{\mu} = \mu^*$. From (6.3.14) we have $\lim_{\ell \to \infty} \|z^i_{k_{\ell}} - z^*\| = 0$ with probability 1 for all $j$, which implies that $\lim_{\ell \to \infty} V(z^*, \mu^*, z_{k_{\ell}}, \mu_{k_{\ell}}) = 0$. However, we have shown that the sequence $V(z^*, \mu^*, z_{k}, \mu_{k})$ converges with probability 1. This implies that $\lim_{k \to \infty} V(z^*, \mu^*, z_k, \mu_k) = 0$ with probability 1 for all $j$. Owing to the strong convexity of the Bregman functions we have

$$V(z^*, \mu^*, z_k, \mu_k) \geq \sum_{i=1}^{m} \left( \frac{\sigma_x}{2} \|x^* - x^i_k\|^2 + \frac{\sigma_\eta}{2} \|\eta^* - \eta^i_k\|^2 + \frac{\sigma_\mu}{2} \|\mu^*_i - \mu^i_k\|^2 \right),$$

108
which implies that the primal variables $x^j_k \to x^*$ and $\eta^j_k \to \eta^*$ with probability 1 for all $j$, and the dual variables $\mu^j_k$ converge to the corresponding optimal dual variables $\mu^*_j$ with probability one for all $j$.

A simple point to note is that for the Bregman functions $V_\mu(y, z)$ we do not require the additional property of convexity in the second argument $z$ for every fixed $y$, which was the case for Bregman functions $V_x(y, z)$ and $V_\eta(y, z)$.

6.4 Min-Max Game Against Exogenous Player

In this section we consider the case when the network of cooperative agents need to solve a min-max optimal game against an exogenous player. The exogenous player is a malicious agent/nature which adversely affects the cost of each agent. Let us denote the action of the adversarial agent by $\xi$. We also require that the feasible set of allowable actions $\xi$ is a compact set denoted as $\Theta$. The objective is to solve

$$\min_{x \in X} \max_{\xi \in \Theta} \sum_{i=1}^{m} f_i(x, \xi).$$

(6.4.1)

This can be thought of as the robust version of the problem earlier considered in Chapters 4 and 5, where we considered the optimization problem

$$\min_{x \in X} \sum_{i=1}^{m} \mathbb{E}_\xi [f_i(x, \xi)].$$

(6.4.2)

In certain cases when it is desired to model the unknown signal $\xi$ as lying in an uncertainty set $\Theta$ the robust version of the problem (6.4.1) is more suitable. The problem (6.4.1) could alternatively be thought of as a zero sum game between the exogenous player and the network. To guarantee the existence of a min-max optimal solution to (6.4.1) we impose the following assumption on the problem.

Assumption 12.

a) The cost functions $f_i(x, \xi)$ are convex in $x$ for every fixed value of $\xi \in \Theta$, and concave in $\xi$ for every fixed $x \in X$. 

109
b) The constraint sets $X$ and $\Theta$ are convex and compact.

Under the above assumption the minmax problem (6.4.1) admits a solution set $X^*, \Theta^*$ such that for any $x^* \in X^*$, and $\xi^* \in \Theta^*$ we have the saddle-point property [28]

$$\sum_{i=1}^{m} f_i(x^*, \xi) \leq \sum_{i=1}^{m} f_i(x^*, \xi^*) \leq \sum_{i=1}^{m} f_i(x, \xi^*),$$  \hspace{1cm} (6.4.3)

where $x$ and $\xi$ are arbitrary points in the feasible set.

6.4.1 Algorithm

Once again we denote the Bregman distance functions as $V_x(\cdot, \cdot)$ for the set $X$ and $V_\xi(\cdot, \cdot)$ for the set $\Theta$. We propose the following algorithm for the min-max problem (6.4.1).

$$x_{i+1}^k = \arg\min_{y \in X} \left[ \alpha_k \langle \nabla_x f_i(\tilde{x}_i^k, \tilde{\xi}_i^k), y \rangle + V_x(y, \tilde{x}_i^k) \right]$$

$$\xi_{i+1}^k = \arg\min_{\zeta \in \Theta} \left[ -\alpha_k \langle \nabla_\xi f_i(\tilde{x}_i^k, \tilde{\xi}_i^k), \zeta \rangle + V_\xi(\zeta, \tilde{\xi}_i^k) \right],$$  \hspace{1cm} (6.4.4)

where $\nabla_x$ denotes the partial derivative operator with respect to the variable $x$ and $\nabla_\xi$ denotes the partial derivative operator with respect to the variable $\xi$. The initial conditions $x_0^i$ and $\xi_0^i$ satisfy $x_0^i \in X$ and $\xi_0^i \in \Theta$ for all $i$. It is also assumed that the constraint sets $X$ and $D$ are common knowledge for each agent $i \in V$.

The analysis of the algorithm follows along similar lines to that of the algorithm (6.3.2). This can be seen in light of the fact that the primal-dual algorithm in (6.3.2) computes a saddle-point of the Lagrangian function (6.2.2) whereas the algorithm (6.4.4) computes a saddle-point of the problem (6.4.1). A major difference between the algorithms is the fact that in (6.3.2) the agents update their own local dual variables $\mu_k^i$ whereas in the algorithm (6.4.4) the agents update the whole vector $\xi$. Note that there is no stochasticity in the current formulation unless we consider a stochastic model of the network. The final result is that asymptotically the agents estimates $x_{k}^i$ and $\xi_{k}^i$ converge to a common min-max optimal point $x^*$ and $\xi^*$. We formalize the statement in the following theorem.

**Theorem 6.4.1.** Let us assume that the assumptions on the network (Assumption 9) hold and
further Assumption 12, guaranteeing a min-max solution holds. Moreover, assume that the Bregman
distance functions \( V_x(y, z) \) and \( V_\xi(\xi, \zeta) \) in the algorithm (6.4.1) are convex in their second arguments
\( z \) and \( \zeta \) respectively for every fixed \( y \) and \( \xi \). If the step size \( \alpha_k \) in the algorithm (6.4.1) are chosen
to satisfy \( \sum_{k=0}^{\infty} \alpha_k = \infty, \sum_{k=0}^{\infty} \alpha_k^2 < \infty \) then the local variables \((x_k^i, \xi_k^i)\) converge to a common
saddle-point solution \((x^*, \xi^*)\) for the min-max problem (6.4.1), for all \( i \).

Proof. The proof proceeds similarly to the proof of Theorem 6.3.3 by considering the Lyapunov
function \( \sum_{i=1}^{m} (V_x(x^*, x_k^i) + V_\xi(\xi^*, \xi_k^i)) \).

---

6.5 Uplink Power Control

In this section we show the suitability of our algorithms (6.2.8) and (6.3.2) to achieve a min-max
fair allocation of utility in a cellular network. We will keep our discussion brief and refer the readers
to [66] for a general discussion on the power allocation problem. We will be using the formulation
discussed in [17] for our purposes.

There are \( m \) mobile users (MU) in neighboring cells communicating with their respective base
stations (BS) using a common wireless channel. Let \( p_i \) denote the power used by MU \( i \) to commu-
nicate with its base station. Due to the shared nature of the wireless medium the total received
SINR at BS \( i \) is given by

\[
\gamma_i(\bar{p}, \bar{h}_i) = \frac{p_i h_{i,i}^2}{\sigma_i^2 + \sum_{j \neq i} p_j h_{i,j}^2},
\]

where \( h_{i,j} \) is the channel coefficient between MU \( j \) and BS \( i \), and \( \sigma_i^2 \) is the receiver noise variance.
The vector containing power variables \( p_i \) is denoted \( \bar{p} \) and the vector of channel coefficients at BS \( i \)
is denoted \( \bar{h}_i \). The power variables are constrained to a maximum value of \( p_t \), i.e., \( p_i \leq p_t \) for all \( i \).

Let \( U_i(\gamma_i(\bar{p}, \bar{h}_i)) \) be the utility derived by BS \( i \) and \( V(p_i) \) be a cost function penalizing excessive
power. Then, we are interested in finding an allocation that minimizes the worst case loss to any
agent \( i \). In other words, we wish to solve the following problem:

\[
\min_{\bar{p} \in \Pi} \max_{i \in V} \left[ V(p_i) - U_i(\gamma_i(\bar{p}, \bar{h}_i)) \right],
\]

where \( \Pi = \{ \bar{p} \in \mathbb{R}^m \mid p_i \leq p_t \text{ for all } i \} \) and \( p_t \) is the maximum power. We consider the logarithmic
utility function $U_i(x) = \log(x)$. In this case, using the transformation $p_i = e^{x_i}$, it can be shown that the preceding problem can be cast in the form of (6.1.1) with

$$f_i(x) = \log \left( \sigma_i^2 h_{i,i}^{-2} e^{-x_i} + \sum_{j \neq i} h_{i,i}^{-2} h_{j,i}^2 e^{x_j - x_i} \right) + V(e^{x_i}),$$

and $X = \{x | x_i \leq \log(p_i) \text{ for all } i\}$.

### 6.5.1 Simulation Result

In our simulations, we considered a cellular network of 16 square cells of the same size. Within each cell, the MU is randomly located and the base station is located at the center of the cell. The network under consideration is shown in Figure 6.1. The channel coefficient $h_{i,j}$ is assumed to decay as the fourth power of the distance between the MU $j$ and the BS $i$. The shadow fading is assumed to be log-normal with variance 0.1. The receiver noise variance $\sigma_i^2$ is taken to be 0.01. The cost of the power is modeled as $V(p_i) = 10^{-3} p_i$. The step size is taken to be $\alpha_k = \frac{50}{p_{\max}}$ for the exact penalty algorithm and $\alpha_k = \frac{4}{p_{\max}}$ for the distribtued primal-dual algorithm. The parameter $r_i$ in the algorithm (6.2.8) is chosen to be 1.3 for all $i$. For the purpose of this simulation all the Bregman distance generating functions involved were chosen to be Euclidean norm functions. Figure 6.3 shows the behavior of the algorithms (6.2.8) and (6.3.2) for this problem. The optimal min-max allocation was computed using a centralized gradient descent scheme for the penalized problem (6.2.1).

### 6.6 Conclusion

In this Chapter we presented distributed algorithms for min-max allocation problem in networks. We formulated our algorithms in the general setting of Bregman distances. Specifically we discussed two algorithm. The first algorithm uses a non-differentiable penalty function to translate the min-max problem to a format which is suitable for distributed algorithms. The second algorithm is based on the primal-dual iterative update scheme. In both these algorithms we allow the presence of stochastic subgradient noise. We further provided a brief discussion on the min-max problem in the presence of exogenous disturbance.
Figure 6.1: The circles denote the base stations. The dotted lines denote the communication links between adjacent BSs. The cross denotes the MUs. The bold lines connect each MU to its respective BS.

Figure 6.2: The final iterate values after 4000 iterations of the algorithms. The plot shows the allocation achieved by Centralized gradient descent, Distributed Exact Penalty Algorithm, Distributed Primal Dual Algorithm, and Centralized Primal-Dual Algorithm.
Figure 6.3: The plot shows the convergence in $l_2$ norm of the distributed primal-dual algorithms’ dual vector to the optimal dual vector computed using the centralized primal-dual algorithm.
Chapter 7

Application to Distributed Supervised Learning

The current interest in distributed learning fuels from the need to make inferences based on a deluge of large scale and distributed data sets. In such scenarios the classical assumptions that there is a central server with access to all the data doesn’t hold. In its most general setting the distributed learning problem is to perform a regression or classification task when there is a network of computational nodes which have access to streaming data. In some cases, we can hope to store the data at each of these nodes so that any learning algorithm can perform multiple passes over these training sets. This is the problem definition of a distributed batch learning problem. However in many cases it is desirable to perform computation in an online manner when each of the nodes update their model parameters sequentially and communicate with other nodes to arrive at a jointly optimal solution. This is the framework of distributed online learning.

At this point we would like to emphasize the need to study distributed first order methods based on gradient/subgradient descent. Gradient descent methods are by far the most widely used approaches for both the batch and online learning approach. The power of this approach can be gauged from the fact that Adaboost, which is one of the most popular techniques in machine learning, was also shown to be a restatement of the coordinate descent based approach in optimization when applied to a suitably chosen loss function [67, 68]. An important factor which influences the development of distributed algorithms is the assumption about the underlying architecture of the network. A popular architecture is the shared memory architecture. The commercially available multicore CPU’s on desktop computers fall in this category. In this case there are multiple processors which operate on a common memory space where each of the processors can update a common parameter vector. This scenario is considered in [1,69]. We are more interested in the case when there is no such central node which can communicate with all nodes. In our setting which we established in Chapter 4 and 5 the different local nodes have their own local storage and
can communicate over a time varying communication network.

In this chapter we utilize the results developed in Chapter 3, Chapter 4 and 5 to provide a framework for distributed solutions to some problems arising in statistical learning.

The rest of this Chapter is organized as follows. In Section 7.1 we provide a brief introduction to the supervised learning problem and formalize the notions of batch learning and online learning. Then, in Section 7.2 we provide our discussion on the distributed batch learning problem and adapt our algorithm for the case of distributed batch learning in a support vector machine framework and provide simulation results. In Section 7.3 we discuss the distributed online learning case. We adapt our algorithm for distributed online learning in the setting of support vector machines and provide simulation result. Finally in Section 7.4 we conclude our discussion.

7.1 Supervised Learning Problem

In the classic supervised machine learning problem we are given a set of training data $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. This data is drawn from an unknown underlying distribution $P_{X \times Y}$. The problem of learning is to come up with a function $g(x, \beta^*)$ from a given parameterized set of functions $g(x, \beta)$, where $\beta$ is a parameter lying in the parameter set $\Lambda$. The function $g(x, \beta^*)$ is chosen so as to approximate the underlying relationship between $x$ and $y$. This can be mathematically modeled as the task of minimizing the following risk functional

$$\min_{\beta \in \Lambda} R(\beta) = \min_{\beta \in \Lambda} \left[ \int L(y, g(x, \beta)) dP(x, y) + \Omega(\beta) \right], \quad (7.1.1)$$

where $\Omega(\beta)$ is a regularization term. Some typical examples of the loss function used in various machine learning tasks are as follows:

1. Least-squares regression: $x \in \mathbb{R}^n, y \in \mathbb{R}$, $g(x, \beta) := \beta^t x$, and $L(y, g(x, \beta)) = (y - \beta^t x)^2$.

2. Classification with Hinge-loss: $x \in \mathbb{R}^n, y \in \{-1, +1\}$, $g(x, \beta) = \beta^t x$, and $L(y, g(x, \beta)) = \max\{0, 1 - y(\beta^t x)\}$.

3. Logistic regression: $x \in \mathbb{R}^n, y \in \mathbb{R}$, $g(x, \beta) = \exp(-y(\beta^t x))$, and $L(y, g(x, \beta)) = \log(1 + \exp(-y(\beta^t x)))$. 

116
Some typical choices of the regularization term include:

1. $l_1$-regularization: $\Omega(\beta) = c \|\beta\|_1$. The use of $l_1$-regularization is desired when the feature space $X$ is high dimensional and it is desired to find the parameters which have small number of non zero entries.

2. $l_2$-regularization: $\Omega(\beta) = c \|\beta\|_2^2$. This is by far the most widely used regularization term as it has nice differentiability property.

3. $\Omega(\beta)$ can be chosen to be the indicator function of a closed and convex set $C$ as

$$\Omega(\beta) = \begin{cases} 0 & \text{if } \beta \in C \\ \infty & \text{otherwise} \end{cases}$$

A similar formulation of the learning problem can be done in the nonparametric setting. In the nonparametric setting we assume that our decision functions $g(x)$ lie in some functional space $\mathcal{H}$, with a suitably defined norm $\|\cdot\|_{\mathcal{H}}$. In this case similar to the parametric setting the objective of the learning problem can be written as

$$\min_{g \in \mathcal{H}} R(g) = \min_{g \in \mathcal{H}} \left[ \int L(y, g(x)) d\mathbb{P}(x, y) + c \|g\|_{\mathcal{H}}^2 \right]. \quad (7.1.2)$$

A widely used choice of the space $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS). A RKHS is characterized by a positive definite kernel $K : X \times X \to \mathbb{R}$, with the property that $K(\cdot, x) \in \mathcal{H}$ for every $x \in X$ and for every $f \in \mathcal{H}$, $\langle f, K(\cdot, x) \rangle = f(x)$. The latter property is known as the Reproducing property of the Hilbert space $\mathcal{H}$. A detailed analysis of the properties of RKHS can be found in [70] and the specific application to machine learning problems can be found in [71].

There are two primary approaches to solve the problems (7.1.1) and (7.1.2) namely Batch Learning and Online Learning. We now give a brief discussion on these approaches.

### 7.1.1 Batch Learning

The batch learning framework is characterized by the fact that the training data

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$$
is used in a batch to approximate the integral in equations (7.1.1) and (7.1.2) by an empirical sum. In this case the problems (7.1.1) and (7.1.2) reduce to

$$\min_{\beta \in \Lambda} R_n(\beta) = \min_{\beta \in \Lambda} \frac{1}{n} \left[ \sum_{i=1}^{n} L(y_i, g(x_i, \beta)) + \Omega(\beta) \right],$$  

(7.1.3)

and

$$\min_{g \in \mathcal{H}} R(g) = \min_{g \in \mathcal{H}} \frac{1}{n} \left[ \sum_{i=1}^{n} L(y_i, g(x_i)) + c \|g\|_{\mathcal{H}}^2 \right],$$  

(7.1.4)

respectively. The approximation of the integral with an empirical sum is known in the statistical learning literature as the principle of empirical risk minimization. Let us focus on the parametric learning problem of (7.1.3). Let us assume that the decision space is the span $\mathcal{H}$ generated by basis functions of the form $\{h_1(x), \ldots, h_k(x)\}$, where $h_j : \mathbb{R}^p \to \mathbb{R}$. Hence the decision space in this case is the set of functions with a representation $g(x, \beta) = \sum_{i=1}^{k} \beta_i h_i(x)$. Also, let us assume the case of $l_2$ regularization. The problem to be solved in this case is

$$\min_{\beta} \sum_{i=1}^{n} \|y_i - H(x_i)\|_2^2 + \lambda \|\beta\|_2^2. \tag{7.1.5}$$

Here, $H(x_i) = [h_1(x_i), \ldots, h_k(x_i)]$ and $\lambda \|\beta\|_2^2$ is the regularization term. This is a least squares problem the solution of which is given by

$$\beta^* = \left[ \lambda I + H'H \right]^{-1} H'y,$$

where

$$H = \begin{bmatrix} H(x_1) \\ \vdots \\ H(x_n) \end{bmatrix}, \quad y = [y_1, \ldots, y_n]^T.$$

An alternative view to this problem was presented in [20]. Let us define an $n + k$ dimensional vector space $\mathcal{H}$ equipped with the norm $\|(y_1, \ldots, y_n, \beta)\|_{\mathcal{H}}^2 := \|y\|_2^2 + \lambda \|\beta\|_2^2$. Further define a subspace $X \subset \mathcal{H} = \{(z, \beta) : z_i = H(x_i)\beta \text{ for all } i \in 1, \ldots, n\}$. Then the problem (7.1.5) can be rewritten
as

$$\min_{z, \beta} \| y - z \|^2 + \lambda \| \beta \|^2$$

such that \((z, \beta) \in X\) \hspace{1cm} (7.1.6)

Thus the optimal solution is given by the projection of the vector \((y, 0)\) on the subspace \(X\) under the norm \(\| \cdot \|_H\). It can be seen that the projection operator \(P_X\) has the matrix representation

$$P_X := \begin{bmatrix}
H(\lambda I + H' H)^{-1} H' & \lambda H(\lambda I + H' H)^{-1} \\
(\lambda I + H' H)^{-1} H' & \lambda(\lambda I + H' H)^{-1}
\end{bmatrix}.$$ 

Thus the optimal \(z^* = H(\lambda I + H' H)^{-1} H' y\) and the optimal \(\beta^* = (\lambda I + H' H)^{-1} H' y\).

For the case of nonparametric learning Eq. (7.1.4), it has been shown that the search for the minimizer can be restricted to a finite dimensional space parameterized as \(\sum_{i=1}^{n} \beta_i K(x_i, \cdot)\), where \(K(\cdot, \cdot)\) is the kernel and \(x_i\) are the training data points. This, very powerful result is known as the Representer Theorem, and was first arrived at in the field of approximation theory in [72]. This result essentially makes nonparametric learning a tractable problem.

7.1.2 Online Learning

The online learning problem is essentially the problem of stochastic optimization which directly tries to minimize the objective function (7.1.1). This is exactly the framework of stochastic approximation algorithms. Stochastic approximation algorithms are a well developed field in their own right and any attempt to provide a brief overview is futile. Instead we refer the readers to several good references [44, 73–75]. The essential feature of the algorithm is to consider the single sample estimate of the gradient of the objective function. This makes the stochastic approximation schemes ideal for online learning as they operate on a single data point at a time. The limitation of stochastic gradient descent algorithms is their slow rate of convergence, which is of the order \(O(\frac{1}{\sqrt{k}})\). Refer to [26] for more discussion regarding rate of convergence of stochastic approximation schemes.

Recently the online learning problem was extended to the nonparametric learning case in [76]
and [77]. In [76] the authors studied the case of nonparametric learning when the cost function and the regularization term are quadratic functions. The online learning algorithm in this case is given as

\[ f_{k+1} = f_k - \alpha_k((f_k(x_k) - y_k)K(x_k, \cdot) + \lambda f_k). \]  

(7.1.7)

In [77] the authors studied the online classification problem in a RKHS setting.

### 7.2 Distributed Batch Learning

The distributed learning problem refers to the case when the training data is not available at a central node. This situation can arise in two ways. In one case the data is separated artificially among a set of processors. This case is relevant in the case of massive data sets as it prohibits the use of a single processor. Another case arises when different entities can observe only part of the feature space. A formal description can be given as follows.

In the distributed learning framework we assume that we are provided a set of processors indexed by \( i = 1, \ldots, m \) which collaboratively operate on the training set \( S \). The entire data set can be thought of as consisting of a set of observations \( S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \). Let us define the data matrix

\[
X = \begin{bmatrix}
x_1^T \\
\vdots \\
x_n^T
\end{bmatrix}, \quad y = [y_1, \ldots, y_n]^T,
\]

where each feature vector \( x_i \in \mathbb{R}^p \) and \( y_i \in \mathbb{R} \). The data set is said to be horizontally partitioned if the rows of the data set \( X \), and \( y \) are partitioned among the agents. Similarly the data set is said to be vertically partitioned if the columns of the data matrix \( X \) are partitioned. In our discussion, we restrict ourself to horizontally partitioned data.

#### 7.2.1 Least-Squares Regression

In this section we show that we can use the result developed in Chapter 3, specifically Theorem 3.5.1 to develop a distributed supervised learning algorithm for the least-squares regression problem. The data subsets that the node \( i \) operates on is denoted by \( S_i \subseteq S \). Let the set \( \bar{S}_i \) denote the set
of indices of the data set which is available to node \(i\). A data partition \(\{S_1, \ldots, S_m\}\) induces an undirected graph \(E\) over the set of nodes. An edge \(e(i, j)\) exists between node \(i\) and node \(j\) if \(\bar{S}_i \cap \bar{S}_j \neq \emptyset\). Given an index \(k\) let us denote the set of nodes which have the data \((x_k, y_k)\) as \(N_k\). In addition, we assume that each node \(i\) has its own set of basis functions \(\mathcal{H}_i = \{h^i_1(x), \ldots, h^i_k(x)\}\). Let us denote \(H_i(x_j) = [h^i_1(x_j), \ldots, h^i_k(x_j)]\). The distributed learning problem can now be written as [20]:

\[
\begin{align*}
\minimize_{z, \beta_1, \ldots, \beta_m} \quad & \|y - z\|^2 + \sum_{j=1}^m \lambda_j \|\beta_j\|^2 \\
\quad \text{s.t.} \quad & z_i = H_j(x_i)\beta_j \quad \text{for all} \quad j \in N_i, i = 1, \ldots, n.
\end{align*}
\] (7.2.1)

An important point to note here is the fact that from the way constraints are introduced in the above problem, node \(k\) and neighboring node \(l\) need to agree on only the predicted values \(H_k(x_i)\beta_k\) and \(H_l(x_i)\beta_l\) of the data indices \(i\) which are common to \(\bar{S}_k\) and \(\bar{S}_l\). Let us define the vector space \(\mathcal{H}\) of dimension \(n + \sum_{j=1}^m k_j\) with the norm

\[
\|z, \beta_1, \ldots, \beta_m\|_{\mathcal{H}}^2 = \|z\|^2 + \sum_{j=1}^m \lambda_j \|\beta_j\|^2.
\]

The constraint subspace of node \(j\) is given as

\[
X_j := \{(z, \beta_1, \ldots, \beta_m) : z_i = H_j(x_i)\beta_j \quad \text{for all} \quad i \in \bar{S}_j\}.
\]

It can be seen that the set

\[
X = \cap_{j=1}^m X_j = \{(z, \beta_1, \ldots, \beta_m) : z_i = H_j(x_i)\beta_j \quad \text{for all} \quad i \in \bar{S}_j, j = 1, \ldots, m\}.
\]

Then, clearly the optimal solution of the problem is the orthogonal projection of vector \((y, 0, \ldots, 0)\) on the set \(X\). Though a solution to this problem is most desirable, our algorithm in its present form is more suitable to a special case of the above introduced general problem. In the next section we introduce the special case and adapt our consensus based algorithm for this problem.
Special Case:

A special case of the distributed learning problem is the case when we require all the agents to agree on the predicted value of all the data variables in the set $S$, even though each agent $j$ only operates on its local data set $S_j$. This can also be seen as a distributed scheme for the centralized regression problem (7.1.6), when the data is horizontally partitioned. Consider the special case when $H_1 = \ldots = H_m$. As in the consensus part, in our formulation each node maintains a copy of the decision variables. Let us denote the copy of the decision variable $(z, \beta)$ with node $j$ as $(z_j, \beta_j)$. The key feature of the centralized regression problem is the fact that it is desired that asymptotically all the nodes achieve consensus on the value of the vector $z$ and the regressor $\beta$. In this case as in the centralized regression part we define a $n + k$ dimensional vector space $\mathcal{H}$ equipped with the norm

$$
\|z_1, \ldots, z_n, \beta\|_{\mathcal{H}}^2 := \|z\|^2 + \lambda \|\beta\|^2.
$$

(7.2.2)

Since the basis functions are same in this case the set $X_j$ is defined as

$$
X_j := \{(z, \beta) : z_i = H_j(x_i) \beta \text{ for all } i \in \bar{S}_j\}.
$$

(7.2.3)

The convergent behavior of the algorithm is summarized in the following lemma.

**Lemma 7.2.1.** The distributed training algorithm of Table 7.2.1 converges asymptotically to the optimal solution of the problem (7.1.6)

**Proof.** It is clear that the optimal solution of problem (7.1.6) is given by $P_X((y, 0)^t)$. Now the result follows from Theorem 3.5.1.

122
Table 7.1: Consensus Based Distributed Training Algorithm

<table>
<thead>
<tr>
<th>Input:</th>
<th>Partitioned data ( {S_1, \ldots, S_m} ), basis functions ( \mathcal{H} = {h_1(x), \ldots, h_k(x)} ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize:</td>
<td>Initialize ((z_i(0), \beta_i(0)) = P_{X_i}[(y, 0)'] ) for all ( i = 1, \ldots, m ).</td>
</tr>
<tr>
<td>Train:</td>
<td>for ( t=1, \ldots, T ) \begin{itemize} \item <strong>Receive</strong>: For every node ( i ) receive noisy data ((z_{ij}(t), \beta_{ij}(t))) from neighboring nodes. \item <strong>Consensus Step</strong>: For every node ( i ) compute: ( v_i(t+1) = (z_i(t), \beta_i(t))' + \alpha(t+1) \sum_{j=1}^m R_{ij}(z_{ij}(t), \beta_{ij}(t))' ) \item <strong>Projection Step</strong>: For every node ( i ) compute: ( (z_i(t+1), \beta_i(t+1))' = P_{X_i}[v_i(t+1)] ) \end{itemize}</td>
</tr>
<tr>
<td>Send:</td>
<td>For every node ( i ) send ((z_i(t+1), \beta_i(t+1))) to its neighbors.</td>
</tr>
<tr>
<td>Output:</td>
<td>((z_i(T), \beta_i(T))) for all ( i = 1, \ldots, m )</td>
</tr>
</tbody>
</table>

7.2.2 General Convex Loss Functions

In the above lemma we saw that for the case of a quadratic loss function the optimal solution of problem (7.2.1) has the interpretation of being the projection on the constraint set \( X \). In this case we could use the result from Chapter 3 to derive a distributed algorithm. In this section we show that we can use the result from Chapter 4 and 5 to derive algorithms when the cost function is a general nondifferentiable convex function. This enables us to tackle distributed classification problems in this framework which typically employ the hinge-loss function which is a non-differentiable convex function. Consider the following centralized problem

\[
\min_{\beta \in \Lambda} \sum_{i=1}^n [L(y_i, g(x_i, \beta))] + \Omega(\beta).
\]

Here \( L(y_i, g(x_i, \beta)) \) is a convex function of \( \beta \) and \( \Lambda \) is a convex and compact set with nonempty interior. In the case of classification with Hinge-loss \( g(x, \beta) = \beta'x \) and \( L(y, g(x, \beta)) = \max\{0, 1 - y(\beta'x)\} \), and in the case of logistic regression \( g(x, \beta) = \exp(-y(\beta'x)) \), and \( L(y, g(x, \beta)) = \log(1 + \exp(-y(\beta'x))) \). Again, let us assume that the data is horizontally partitioned among the nodes \( S = S_1 \cup \ldots \cup S_m \). Moreover let us assume that there is no overlapping data, i.e \( S_i \cap S_j = \emptyset \). In this case the centralized problem can be naturally written in the form \( \min_{\beta} \sum_{i=1}^m f_i(\beta) \quad \beta \in \mathbb{R}^p \),
where the local objective function is

\[ f_i(\beta) = \sum_{j \in \bar{S}_i} [L(y_j, g(x_j, \beta))] + \frac{1}{m} \Omega(\beta), \]

and \( X_1 = \ldots, X_m = \Lambda \). The distributed algorithm in this case, if the noisy communication aspect is ignored is given by

\[ \beta_i^{k+1} = P_{\Lambda} \left[ \sum_{j=1}^{m} [W_k]_{ij} \beta_j^k - \gamma_{k+1} d_{k+1}^i \right], \quad (7.2.4) \]

where \( d_{k+1}^i \) is the subgradient of the local objective function \( f_i(\beta) \) evaluated at \( \sum_{j=1}^{m} [W_k]_{ij} \beta_j^k \).

Before we can apply the result of Theorem 4.6.1 to claim convergence of the algorithm to the unique minimizer of the centralized problem.

**Distributed Support Vector Machines:**

In this section we adapt our distributed optimization algorithm for the SVM problem. We first provide a little background for the problem of interest.

Let us assume that the feature vectors \( x \in \mathbb{R}^p \), and the output values \( y \in \{-1, 1\} \). This is a binary classification problem. Let us assume that the data sets are linearly separable in the two classes. Then, the maximum margin classification problem deals with finding the hyperplane which separates the data sets with the maximum margin. It can be shown [43] that the maximum margin classification problem can be written as the following optimization problem:

\[
\begin{align*}
\text{minimize} & \quad \beta^T \beta \\
\text{subject to} & \quad y_i (\beta^T x_i + b) \geq 1 \quad \forall i \in S.
\end{align*}
\]

\( (7.2.5) \)
Clearly this is a quadratic optimization problem, however typically the solution to the above problem is obtained by solving the dual problem

$$\text{maximize}_{\alpha \geq 0} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to \( \sum_{i=1}^{n} \alpha_i y_i = 0 \), \hspace{1cm} (7.2.6)

where \( \alpha_i \) are Lagrange multipliers. Both primal and dual problems are quadratic optimization problems and efficient algorithms exist for their solutions. It can be shown that once the solution to the dual optimization problem is known the optimal of the primal problem is given by \( \beta = \sum_{i=1}^{n} \alpha_i y_i x_i \), and the margin \( b \) can be computed as \( b = y_i - \beta^T x_i \) for any \( i \) such that \( \alpha_i > 0 \).

The vectors corresponding to which the Lagrange multiplier is nonzero are known as the support vectors. Several algorithms have been proposed for efficient computation of the optimal solution of the above problem. For the case when the data is not linearly separable the primal problem (7.2.5) is modified to be cast as

$$\text{minimize}_{\beta, b} \quad c \beta^T \beta + \sum_{i=1}^{n} \left[ 1 - y_i (\beta^T x_i + b) \right]^+,$$

where \([1-v]^+\) is the hinge loss function and \(c\) is a regularization penalty. A lot of attention has been cast on arriving at distributed solutions to the SVM problem both in the case of horizontal [78], vertical [79], and mixed partition [80] of data. Recently in [19] another approach to distributed SVM classification is presented. Most of the methods focus on a solution to the dual problem, since the dual optimization problem gives us additional flexibility of choosing nonlinear kernel functions.

The work which is most relevant to our setting is [81] and [82]. In both of these papers the authors use the alternating direction method of multipliers to develop distributed algorithms.

Our approach is more suitably applied to the primal problem. It can be seen that given a data partition \( \{S_1, \ldots, S_m\} \), the primal problem can be cast in our framework by rewriting it as

$$\text{minimize}_{\beta, b} \quad \sum_{j=1}^{m} \left( \frac{c}{m} \beta^T \beta + \sum_{i \in S_j} \left[ 1 - y_i (\beta^T x_i + b) \right]^+ \right).$$

(7.2.8)
Our algorithm when adapted to the SVM example is given as

\[
\begin{bmatrix}
\tilde{\beta}_i^k \\
\tilde{b}_i^k
\end{bmatrix} = \begin{bmatrix}
\beta_i^k - \alpha_k \sum_{j=1}^{m} [W_k]_{ij} \tilde{\beta}_j^k \\
\beta_i^k - \alpha_k \sum_{j=1}^{m} [W_k]_{ij} \tilde{b}_j^k
\end{bmatrix},
\]

\[
\begin{bmatrix}
\beta_{k+1}^i \\
b_{k+1}^i
\end{bmatrix} = \begin{bmatrix}
\bar{\beta}_i^k - \gamma_k \left( \frac{2}{m} \tilde{\beta}_i^k - \sum_{r \in S_i} y_r x_r 1_{\{y_r((\bar{\beta}_r^k)(x_r+\bar{b}_r^k) \leq 1)\}} \right) \\
\bar{b}_i^k + \gamma_k \left( \sum_{r \in S_i} y_r 1_{\{y_r((\bar{\beta}_r^k)(x_r+\bar{b}_r^k) \leq 1)\}} \right)
\end{bmatrix}.
\] (7.2.9)

**Simulation Result:**

We now provide our simulation result for a relatively small data set. The data set is generated from a mixture of gaussian distributions with the positive and negative samples generated from a gaussian distribution with different means. For this example we considered a simple network of four nodes with the static communication network shown in Figure 7.1. The centralized data set of 250 instances was distributed amongst the nodes with each node operating on a random exclusive set of 50 instances. For this example the regularization constant \(c\) was chosen to be 3. The simulation results are split into the noiseless communication case and noisy communication case. For the noiseless case the step size \(\alpha_k\) in (7.3.3) is chosen to be a constant \(\alpha_k = 0.8\) for all \(k\) and the step size \(\gamma_k = \frac{1}{k}\). Figure 7.2 denotes the node estimates after 20 iterations of the distributed algorithm for the noiseless communication case and Figure 7.3 denotes the node estimates after 500 iterations.

For the noisy communication case the step sizes were chosen as \(\alpha_k = \frac{1}{k^{0.5}}\) and \(\gamma_k = \frac{1}{k}\). Note that the step sizes chosen this way satisfy the assumption 5.3. The Figure 7.4 shows the node estimates after 1 iteration of the distributed algorithm and the Figure 7.5 shows the node estimates after 500 iterations of the distributed algorithm.
Figure 7.2: The final iterate values after 20 iterations of the distributed algorithm for the noiseless communication case. In the figure ◦ denotes positive samples and ⋆ the negative samples. The solid line indicates the optimal solution, the dotted lines indicate the different node estimates.

Figure 7.3: The final iterate values after 500 iterations of the distributed algorithm for the noiseless communication case. In the figure ◦ denotes positive samples and ⋆ the negative samples. The dotted lines indicate the different node estimates.
Figure 7.4: The iterate values after 1 iterations of the distributed algorithm for the noisy communication case. In the figure ◦ denotes positive samples and * the negative samples. The solid line indicates the optimal solution, the dotted lines indicate the different node estimates.

Figure 7.5: The iterate values after 500 iterations of the distributed algorithm for the noisy communication case. In the figure ◦ denotes positive samples and * the negative samples. The dotted lines indicate the different node estimates.
7.3 Distributed Online Learning in Parallel Data Streams

The above framework is readily extended to the online parametric learning framework. In this formulation the \( m \) processing nodes receive streaming data \((x_i(t), y_i(t))\) assumed to be drawn in an i.i.d fashion from an underlying distribution. The literature on this topic is relatively sparse. A recent work which tackles this setting is given in [83].

The objective function is to directly minimize the risk function (7.1.1). In this case the objective function of each agent is

\[
 f_i(\beta) = \frac{1}{m} \mathbb{E}[L(y, g(x, \beta))] + \frac{1}{m} \Omega(\beta).
\]

To recall, our distributed algorithm if noisy communication aspect is ignored is given by

\[
 \beta_{k+1} = P_A \left[ \sum_{j=1}^{m} [W_k]_{ij} \beta_j^k - \gamma_{k+1} \tilde{d}_k \right], \quad (7.3.1)
\]

where \( \tilde{d}_k \) is the noisy subgradient of the local objective function \( f_i(\beta) \) evaluated at \( \sum_{j=1}^{m} [W_k]_{ij} \beta_j^k \).

7.3.1 Distributed Online Support Vector Machines

Let us consider the online classification problem. The online classification problem deals with the case when the classifier is trained on a stream of labeled data \( \{(x_i, y_i)\} \), where \( x_i \in \mathbb{R}^n \) and \( y_i \in \mathcal{Y} \), where \( \mathcal{Y} \) is some finite set. A typical choice of \( \mathcal{Y} \) for the binary classification problem is \( \mathcal{Y} = \{+1, -1\} \). At each time step \( t \) during training the algorithm makes a label prediction \( h_t(x_t) \).

For the linear classification task the classifier is \( h_t(x_t) = \text{sign}(w'_t x_t) \), where \( w_t \) is the decision variable. After making the prediction, the actual label \( y_t \) is received. A loss function \( l(y_t, h_t(x_t)) \) evaluates the performance of the classifier and is used to update the classifier to \( h_{t+1} \). Even though the online classification problem is motivated from applications arising in networks very few results exist which utilize the power of distributed computation for generating powerful classification schemes. Typically the SVM training problem has been dealt with in the dual formulation. However recently subgradient descent based training methods have been suggested. These methods are particularly suitable for online training. In [77] the authors suggested a subgradient descent based strategy named NORMA for online training of the SVM. This gives them the flexibility to
consider nondifferentiable loss functions. For example the hinge loss function we considered in the
batch framework. In [77] the authors provided useful bounds on the training error for the online
subgradient descent scheme. More concretely they showed that the algorithm requires $O\left(\frac{1}{\epsilon^2}\right)$ itera-
tions to converge to a $\epsilon$ optimal solution. Later in [84] the authors developed another subgradient
based strategy termed Pegasos. Their algorithm operates on mini batches of $k$ examples sampled
from the training set, and alternates between a subgradient descent and a projection step. They
proved that Pegasos takes $\tilde{O}\left(\frac{1}{\epsilon}\right)$ iterations to get to a $\epsilon$ optimal solution. Various other subgradient
descent based online methods have been proposed, for example SGD-QN [85], FOBOS [86] and the
dual averaging scheme in [42]. In [42,86] the authors consider a composite objective function of the
form $F(x) = f(x) + g(x)$, where the function $g(x)$ is a non-smooth regularizer. The motivation for
this structure is the consideration of $l_1$ penalty in the objective function. Another approach for the
online classification problem was provided in [87] where the authors cast the problem of finding a
classifier as a feasibility problem.

Given the amount of research which has been carried out in the online learning framework the
attention to distributed algorithms for the online learning task has been severely limited. A recent
step in this direction is considered in [88].

Let us now consider the online classification problem with hinge loss function and an $l_2$ regular-
ization term. The objective is to compute

$$\min_{\beta, b} \ c\beta^T \beta + E \left[ 1 - y(\beta^T x + b) \right]_+. \quad (7.3.2)$$

The algorithms (7.3.1), with the additional consideration of communication noise when adapted to
the above problem can be written as

$$\begin{bmatrix}
\tilde{\beta}_{k+1}^i \\
\tilde{b}_{k+1}^i
\end{bmatrix} = \begin{bmatrix}
\beta_k^i - \alpha_k \sum_{j=1}^{m} [W_k]_{ij} \beta_j^i \\
b_k^i - \alpha_k \sum_{j=1}^{m} [W_k]_{ij} b_j^i
\end{bmatrix} - \frac{2c}{m} \left( 2c \tilde{\beta}_k^i - y_{i,k}x_{i,k} \mathbf{1}_{\{y_{i,k}(\tilde{\beta}_k^i)^T x_{i,k} + \tilde{b}_k^i) \leq 1\}} \right),$$

$$\begin{bmatrix}
\beta_{k+1}^i \\
b_{k+1}^i
\end{bmatrix} = \begin{bmatrix}
\tilde{\beta}_k^i + \frac{2c}{m} y_{i,k} \mathbf{1}_{\{y_{i,k}(\tilde{\beta}_k^i)^T x_{i,k} + \tilde{b}_k^i) \leq 1\}} \\
b_k^i + \frac{2c}{m} y_{i,k} \mathbf{1}_{\{y_{i,k}(\tilde{\beta}_k^i)^T x_{i,k} + \tilde{b}_k^i) \leq 1\}}
\end{bmatrix}, \quad (7.3.3)$$

130
where \((x_{i,k}, y_{i,k})\) is the data received by node \(i\) at time \(k\).

**Simulation Result:**

For the simulation purpose we considered the simple four node network shown in figure 7.1. The data set is randomly exclusively partitioned into the four nodes with each node operating on 50 data points. At each instance each node randomly selects a data point from its local data set and computes a gradient step based on just this point instead of the whole local data set as was the case in the batch setting. This way we simulate streaming data points. This mode of operation is similar to the PEGASOS algorithm of [84]. For this simulation we chose the regularization constant \(c\) as 3. step sizes were chosen as \(\alpha_k = \frac{1}{k^{0.45}}\) and \(\gamma_k = \frac{1}{k}\). Note that the step sizes chosen this way satisfy the Assumption 5.3 on the step sizes. The Figure 7.6 shows the node estimates after five iterations. The solid line shows the solution from an online algorithm which has access to the whole data set. The Figure 7.7 shows the node estimates after 1000 iterations.
Figure 7.7: The iterate values after 1000 iterations of the distributed online algorithm for the noisy communication case. In the figure ○ denotes positive samples and * the negative samples. The dotted lines indicate the different node estimates.

7.4 Conclusion

In this Chapter we developed a framework for distributed learning in the case when the data sets are distributed across nodes with computation and communication capability. We showed that our algorithms developed in earlier Chapters are well suited to this task. We considered the setting of both batch learning and the online learning framework. In both of these cases we provided simulation results showing the convergent behavior of the algorithms when applied to the problem of training of support vector machines.
Chapter 8

Conclusions

In this work we addressed the problem of distributed optimization in networks. We developed distributed algorithms for the case when the objective function has a special structure. This structure is motivated by various application areas which arise in sensor networks and machine learning. We incorporated the presence of various forms of uncertainties which arise in real world problems. Specifically the developed algorithms account for noisy communication links, stochastic subgradient errors and stochastic communication topology. The ability to handle stochastic subgradient errors allowed the algorithms to be applied in an online setting. The consideration of stochastic communication topology generalizes our algorithms to be applied in an asynchronous setting. In both the synchronous and asynchronous settings we identify sufficient conditions on the network topology for the convergence of our algorithms. We also established conditions on the step size sequences inherent in the algorithm which guarantee convergence. We showed the applicability of the developed algorithms in handling certain formulations of the distributed machine learning problem. We applied the distributed algorithm for a simple four node binary classification task in the support vector machine setting. We showed that the algorithm can be suitably modified to handle both the distributed batch and online learning framework for the classification problem.

We extended our distributed algorithms to the case of Bregman distance based algorithms and developed a distributed framework for computing min-max optimal problems arising in networks. We developed an exact penalty function based approach and a primal-dual iteration based approach for the min-max problem. Both of these algorithms allowed the presence of stochastic subgradient errors and are based on the framework of Bregman distances. We showed the applicability of the developed algorithms on a power allocation problem arising in cellular networks.
8.1 Future Work

In the current work we proposed distributed algorithms which employ first order gradient information. Though this approach has lot of benefits as ease of implementation and robustness to errors, it is well known to have slow convergence rates. On the other hand Newton’s method is known to have superior convergence rates. Though some progress has been made in the direction of distributed Newton’s methods [89], much work is left in developing an efficient distributed Newton’s scheme. Another aspect which is left relatively unexplored is the consideration of delays in networks and how it affects the performance of distributed algorithms. The delay aspect was discussed in [90,91] for consensus algorithms. The consideration of delays for distributed optimization algorithms is directly relevant in network applications.

There has been an interest in applying stochastic subgradient schemes in problems arising in computer vision and data mining in the framework of compressed sensing and matrix recovery. It is of direct interest to explore if the distributed algorithms developed in this thesis can be applied to such problems.
References


