New and Provable Results on Network Inference Problems and Multi-Agent Optimization Algorithms

by

Hoi To Wai

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2017 by the
Graduate Supervisory Committee:

Anna Scaglione, Chair
Visar Berisha
Angelia Nedić
Lei Ying

ARIZONA STATE UNIVERSITY

December 2017

ABSTRACT

Our ability to understand networks is important to many applications, from the analysis and modeling of biological networks to analyzing social networks. Unveiling network dynamics allows us to make predictions and decisions. Moreover, network dynamics models have inspired new ideas for computational methods involving multi-agent cooperation, offering effective solutions for optimization tasks. This dissertation presents new theoretical results on network inference and multi-agent optimization, split into two parts —

The first part deals with modeling and identification of network dynamics. I study two types of network dynamics arising from social and gene networks. Based on the network dynamics, the proposed network identification method works like a 'network RADAR', meaning that interaction strengths between agents are inferred by injecting 'signal' into the network and observing the resultant reverberation. In social networks, this is accomplished by stubborn agents whose opinions do not change throughout a discussion. In gene networks, genes are suppressed to create desired perturbations. The steady-states under these perturbations are characterized. In contrast to the common assumption of full rank input, I take a laxer assumption where low-rank input is used, to better model the empirical network data. Importantly, a network is proven to be identifiable from low rank data of rank that grows proportional to the network's sparsity. The proposed method is applied to synthetic and empirical data, and is shown to offer superior performance compared to prior work. The second part is concerned with algorithms on networks. I develop three consensus-based algorithms for multi-agent optimization. The first method is a decentralized Frank-Wolfe (DeFW) algorithm. The main advantage of DeFW lies on its projection-free nature, where we can replace the costly projection step in traditional algorithms by a low-cost linear optimization step. I prove the convergence rates of DeFW for convex and non-convex problems. I also develop two consensus-based alternating optimization algorithms — one for least square problems and one for non-convex problems. These algorithms exploit the problem structure for faster convergence and their efficacy is demonstrated by numerical simulations.

I conclude this dissertation by describing future research directions.

DEDICATION

To my family

ACKNOWLEDGMENTS

First and foremost, I am greatly indebted to my advisor, Prof. Anna Scaglione, for the guidance and support to me throughout my PhD. Even though we may have different 'taste' in research at times, she has given me the liberty to work on topics that I am interested in. This dissertation could never be completed without her wise advice. Also, thank you my thesis committee — Prof. Visar Berisha, Prof. Angelia Nedić and Prof. Lei Ying — for reading this dissertation and providing useful comments.

I am grateful to Prof. Eric Moulines, Prof. Asuman E. Ozdaglar for hosting me during two productive and wonderful summers in Paris and Boston. Their advices have helped me to diversify and explore different research topics while sharpening my mathematical skills. In addition, I am thankful to Prof. Amir Leshem and Prof. Baruch Barzel for bringing different aspects to my research as well as welcoming me to Tel Aviv for two short visits. I am fortunate to have interacted and collaborated with these mentors.

Besides, I would like to thank Prof. Mahnoosh Alizadeh, Prof. Tsung-Hui Chang, Dr. Jean Lafond, Dr. Shi Wei and Mr. César Uribe for being good collaborators and friends. Particularly with Jean, without him I couldn't have survived Paris with my virtually zero knowledge of French.

Thank you everyone in the SINELab — Reinhard Gentz, Lorenzo Ferrari, Mahdi Jamei, Kari Hreinsson, Eran Schweitzer, Raksha Ramakrishna, Prof. Xiaoxiao Wu, Dr. Bita Analui, Nikhil Ravi, Nurullah Karakoc — for being the best team to work and have fun with. Especially, I would like to thank Reinhard for being the most awesome lab-mate, room-mate and coffee-mate ever, let's not forget our promise and visit Mexico together.

I am thankful to all my friends and colleagues in Davis, San Francisco Bay area, Tempe/Phoenix, Paris and Boston, for giving me a great time and company. The special thanks go to the Infusion Coffee and Tea at Tempe, which makes good coffee that are turned into theorems. Lastly, I would like to thank my family — my mother, father and sisters — in Hong Kong, who have always been supporting me.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

1 Introduction

1.1 Overview and Background

We are living in the cusp of an era where human beings are surrounded by networked systems in their everyday life, from social-economic to biological networks, or from sensor to computer networks. Taking the social and economic networks as an example, we see that the network structure and dynamics determine how we make friends, how we vote in an election and how the government's policies are implemented. The ubiquitous presence of networks present a challenge for scientists from all disciplines to understand *networked systems* of real life and the network structure, dynamic system model underneath them, which will in turn help human beings to predict and determine the behavior of these systems.

The first focus of this dissertation is on the *network dynamics modeling and identification* issues related to the systems discussed above. For examples of the potential impacts, reverse engineering the gene regulatory networks can help discovering the cause of diseases and their prevention [De Smet and Marchal(2010),Buchanan(2010),Albert(2005),Ideker and Sharan(2008), Barabasi and Oltvai(2004)]; or understanding the social networks can help businesses to strategize advertisement spendings on their potential customers [Asch(1955), Ansari *et al.*(2000),Lam and Riedl(2004),Burke *et al.*(2005),Williams *et al.*(2007),Candogan *et al.*(2012)]. As we shall see later, we study the state-of-the-art models for opinion and gene dynamics under a unified notion of *perturbations* and characterize its equilibrium points. Moreover, we propose a network identification method that relies on introducing perturbations above in networked systems, which in the context of social-economical networks, it is related to modeling the *stubborn agents/zealots*; or in the context of gene regulatory networks, it is related to controlling the expressions of the targeted genes. As studied in [Das *et al.*(2014), Moussad *et al.*(2013), Ramos *et al.*(2015), Kuhlman *et al.*(2012), Kang *et al.*(2015), Barzel and Biham(2009)], these perturbations may be introduced intentionally on real networks. We would like to emphasize that we are particularly interested in the *low*

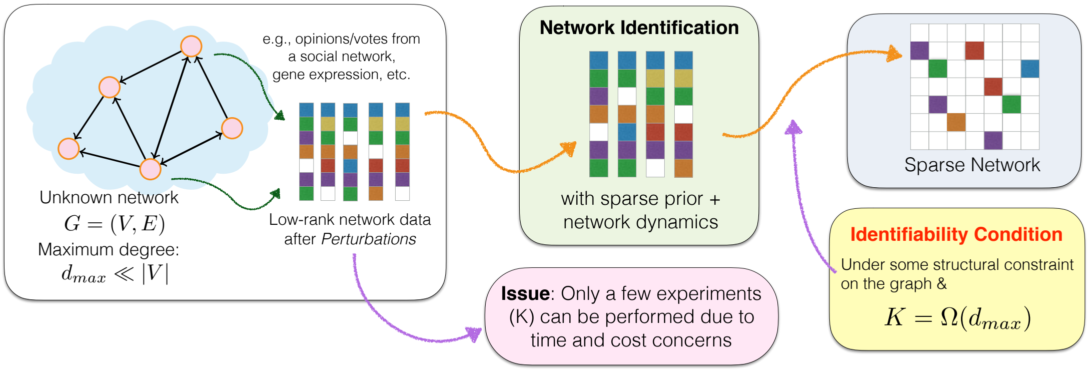
e.g., opinions/votes from a social network, gene expression, etc.
Unknown network
$G = (V, E)$
Maximum degree:
$d_{max} \ll |V|$
Low-rank network data after *Perturbations*
Network Identification
with sparse prior + network dynamics
Sparse Network
Issue: Only a few experiments (K) can be performed due to time and cost concerns
Identifiability Condition
Under some structural constraint on the graph &
$K = \Omega(d_{max})$

Figure 1.1: Overview of the proposed network identification approach. Our scheme makes use of *perturbation experiments* and knowledge of the *network dynamics* to infer the underlying sparse network.

rank observation regime, e.g., that can be induced by having a few perturbation experiments described above. This is a practical concern as well as a technical challenge since the empirical network data are typically low rank. For example, the opinions observed in social networks are often clustered or polarized, and the number of perturbation experiments conducted on gene networks is limited. Under these models, we demonstrate that the network topology and the weights on its edges can be perfectly identified under a mild condition on the number of measurements obtained or, equivalently, the rank of observed data.

The second focus of this dissertation is concerned with the *computation methods* on networked systems, *i.e.,* algorithms that run on the networks. Specifically, the networked systems discussed above can be viewed as systems with multi-agents that are tasked with jointly solving a certain optimization problem. Representative examples are the gossip-like mechanism for agents to reach consensus [DeGroot(1974), Dimakis *et al.*(2010)], or the pulse coupling mechanism that helps fireflies synchronize [Mirollo and Strogatz(1990)]. The above observations have inspired algorithm scientists to develop distributed algorithms to leverage the interconnected nature of computer and sensor networks [Tsitsiklis(1984), Sayed *et al.*(2013)]. These distributed algorithms have the advantages of being resilient to network failure and having the ability to work without central coordination. Such features made these algorithms especially attractive for scenarios when data is stored distributively

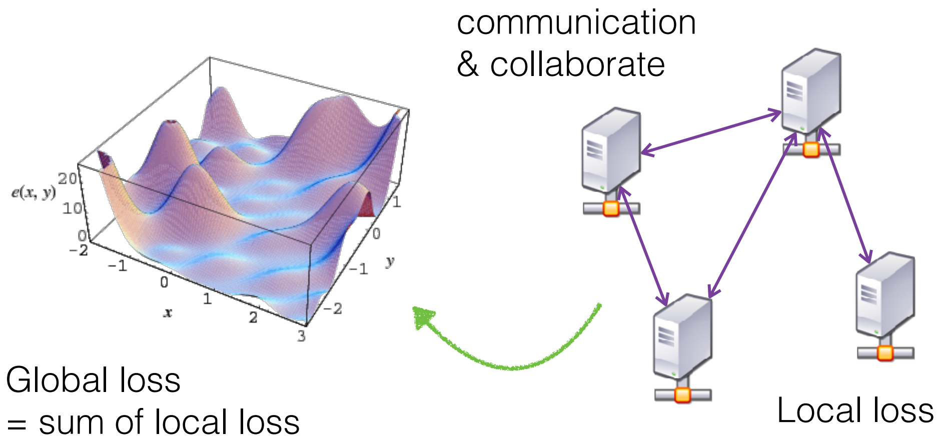

Figure 1.2: Overview of decentralized optimization algorithms. Tackling convex and non-convex optimization with multi-agent optimization is a natural way for harvesting the computational resource distributed over a large network with many agents.

across a bandwidth limited computer network, or when the individually stored data contains private information. This is important in the wake of ‘big data’ era and the need for high-dimensional optimization tool in machine learning applications. For example, the task of training hyper parameters in neural networks [LeCun *et al.*(2015)] with a large amount of data may be greatly accelerated with distributed optimization. This calls for the second goal of this dissertation, which is to develop fast optimization algorithms via distributed optimization and to analyze their performance.

Our current ability to understand the structure and dynamics of real life networks is limited by the lack of a disciplined analysis of the network identification method and the insufficient theories of decentralized optimizations adequate for algorithms that handle high dimensional or structured problems. Overall, this dissertation fills these gaps by proposing and studying network identification methods and novel multi-agents optimization algorithms. Importantly, we analyze the performance of these formulation and algorithm from a mathematical standpoint, providing verifiable conditions that can serve as important guidelines for the practitioners to apply these methods.

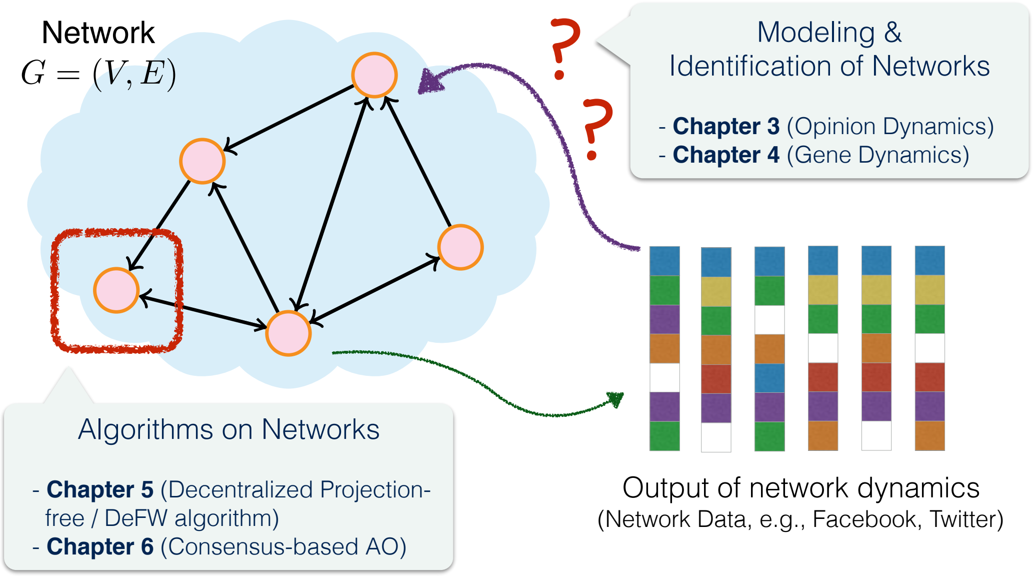
Network
$G = (V, E)$
Modeling &
Identification of Networks
- Chapter 3 (Opinion Dynamics)
- Chapter 4 (Gene Dynamics)
?
?
Algorithms on Networks
- Chapter 5 (Decentralized Projection-free / DeFW algorithm)
- Chapter 6 (Consensus-based AO)
Output of network dynamics
(Network Data, e.g., Facebook, Twitter)

Figure 1.3: Overview of the contributions in this dissertation and the corresponding chapters.

1.2 Contributions and Organization

This dissertation tackles two important problems pertaining to the study of *Network Science*, encompassing new results on *modeling and identification* as well as new *computation methods on networks*; see Figure 1.3 for an illustrative overview. The contributions and pointers to their respective chapters in this dissertation can be found below.

1.2.1 Modeling and Identification of Networks

The first part of this dissertation deals with the *modeling* and *identification* aspects of networks, with applications on social and biological networks. The relevant contents are found in Chapter 3 and 4. In a nutshell, our main idea is to present mathematical models on network dynamics and to show that the corresponding network identification problem can be studied as a similar problem as *RADAR*, *i.e.,* we observe the steady states of the system, treated as 'reverberation', resulted from injecting a 'signal' into the network. As we shall show, this allows us to characterize the causal relationship between the reverberation and the signal injected as a linear system of equations in terms of the network structure, even though the network states may follow a non-linear dynamic system. To demonstrate the said idea, two types of perturbations and dynamics types are studied in Chapter 3 and Chapter 4.

For the social networks studied in Chapter 3, we consider a general (non-linear) opinion dynamics in discrete time and perturb the system by changing the opinions of a selected group of S *stubborn agents* over a number of issues. For the molecular dynamics studied in Chapter 4, we consider a nonlinear, continuous time dynamics and introduce perturbations by performing S experiments such that a different gene is knock out in each experiment. As we will show in the two mentioned chapters, these perturbations lead to two similar algebraic conditions for their dependence on the graph structure. Under a few mild assumptions on the random graph statistics, the sparse recovery based formulation is shown to *perfectly* recover the network, including the weights on its edges, as long as we have $S = \Omega(d_{max})$ stubborn agents or perturbation experiments, where d_{max} is the maximum in-degree of the graph. To our best knowledge, this is the first analytical result on the recoverability of networks from a small number of stubborn agents or perturbation experiments. As the parameter S in the context above can also be interpreted as the *rank* of the observed data, our result also shows that the network identification problem can be solved with a much laxer assumption on the data than the prior work. For both social and gene regulatory networks, we apply our method on synthetic and empirical data to demonstrate its competitiveness.

1.2.2 Algorithms on Network

This dissertation is also concerned with optimization algorithms that runs on *network*, where we design dynamics equations on the networks to solve optimization problems. In this regard, we propose three new consensus-based algorithms. Chapter 5 presents a new decentralized optimization algorithm that is called the decentralized Frank-Wolfe (DeFW) algorithm for general optimization problems. The DeFW algorithm is built on the classical gossip-based average consensus protocol using a carefully designed message exchange mechanism and the recently popular Frank-Wolfe algorithm [Frank and Wolfe(1956),Jaggi(2013)]. With a focus on constrained optimization, a main feature of the DeFW algorithm lies on its projection-free nature, where we replace the costly Euclidean projection step in conventional optimization algorithm by a linear minimization oracle. Depending on the problem structure, such modification can lead to orders of savings in complexity per iteration. We further analyze the convergence rate of the DeFW algorithm. For convex and smooth objectives, let t be the iteration number, we show that the gap to optimal objective value for the iterates generated by DeFW decays as $\mathcal{O}(1/t)$; it can be accelerated to $\mathcal{O}(1/t^2)$ if the objective function is strongly convex and the optimal solution is in the interior of the constraint set. For non-convex but smooth objectives, the iterates generated by the DeFW algorithm are shown to converge to a stationary point asymptotically, and the convergence rate is as fast as $\mathcal{O}(\sqrt{1/t})$, depending on the step size rule chosen. We demonstrate applications of the DeFW algorithm on machine learning problems such as distributed matrix completion and communication efficient sparse learning. Our numerical results on real data suggests that the DeFW algorithm achieves a complexity saving of 20-30 times over the projection based distributed algorithms. We also show that when applied to low-rank regression problems, the DeFW algorithm can be tailor made with even lower computation complexity requirement on individual agents.

Chapter 6 presents two new consensus-based alternating optimization (AO) algorithms. These algorithms are suitable when the optimization problem exhibits additional structures that are exploitable for faster solution methods. Specifically, the first algorithm, called C-AOLS, is designed for least square problems with local nuisance parameters and rely on computing the closed form solution in a decentralized manner. The second algorithm is called the EXTRA-AO algorithm. The algorithm is developed from a recent gradient based distributed optimization algorithm called EXTRA [Shi *et al.*(2015)] and it is shown that the EXTRA-AO is applicable to generic smooth non-convex optimization problems. For the C-AOLS algorithm, we prove that it converges asymptotically to a stationary point even when the original problem is non-convex, once the algorithm is initialized at a point close enough to the stationary point. For the EXTRA-AO algorithm, we provide necessary conditions on the step sizes for the algorithm to converge to a stationary point. We demonstrate applications of these algorithms on signal processing problems such as signal estimation with asynchronous measurements and decentralized dictionary learning. Our results show that the proposed algorithms outperform the state-of-the-art while requiring similar levels of complexity.

1.3 List of Publications

Our results on network identification can be found in:

B1. H.-T. Wai, A. Scaglione and A. Leshem, "*Active Sensing of Social Networks: Network Identification from Low Rank Data*", chapter in Cooperative and Graph Signal Processing, edited by Petar Djuric and Cedric Richard, Elsevier, 2017, submitted.

J1. S.-X. Wu, H.-T. Wai and A. Scaglione, "*Estimating Social Opinion Dynamics Models from Voting Records*", submitted to IEEE Transactions on Signal Processing, Aug., 2017.

J2. H.-T. Wai, A. Scaglione and A. Leshem, "*Active Sensing of Social Networks*", IEEE Transactions on Signal and Information Processing over Networks, Sept., 2016.

C1. H.-T. Wai, S. Segarra, A. E. Ozdaglar and A. Scaglione, "*Community Detection from Low-Rank Excitations of Graph Filters*", *submitted to* ICASSP 2018, Oct., 2017.

C2. S.-X. Wu, H.-T. Wai and A. Scaglione, "*Data Mining the Underlying Trust in the US Congress*", in Proc. GlobalSIP 2016, Dec., 2016.

C3. H.-T. Wai, A. Scaglione and A. Leshem, "*Active Online Learning of Trusts in Social Networks*", in Proc. ICASSP 2016, March, 2016.

C4. H.-T. Wai, A. Scaglione and A. Leshem, "*Identifying trust in social networks with stubborn agents, with application to market decisions*", Invited paper, Allerton Conference, Oct. 2015.

C5. H.-T. Wai, A. Scaglione and A. Leshem, "*The Social System Identification Problem*", in Proc. IEEE CDC 2015, Dec., 2015.

X1. H.-T. Wai, A. Scaglione, U. Harush, B. Barzel and A. Leshem, "*RIDS: Robust Identification of Sparse Gene Regulatory Networks from Perturbation Experiments*", Tech. report, ArXiv/1612.06565, Dec., 2016.

Our results on multi-agents and large-scale optimization algorithms can be found in:

B2. H.-T. Wai, A. Scaglione and E. Moulines, contributed book chapter, "*Methods for decentralized signal processing with Big Data*", chapter in Cooperative and Graph Signal Processing, edited by Petar Djuric and Cedric Richard, Elsevier, 2017, submitted.

J3. H.-T. Wai, J. Lafond, A. Scaglione and E. Moulines, "*Decentralized Projection-free Optimization for Convex and Non-convex Constrained Problems*", IEEE Transactions on Automatic Control, Nov., 2017.

J4. H.-T. Wai and A. Scaglione, "*Consensus on State and Time: Decentralized Regression with Asynchronous Sampling*", IEEE Transactions on Signal Processing, June, 2015.

C6. H.-T. Wai, W. Shi, A. Nedić and A. Scaglione, "*Curvature-aided Incremental Aggregated Gradient Method*", Allerton 2017, Oct., 2017.

C7. H.-T. Wai, A. Scaglione, J. Lafond and E. Moulines, "*Fast and Privacy-preserving Distributed Low Rank Regression*", in Proc. ICASSP 2017, Mar., 2017.

C8. J. Lafond⋆, H.-T. Wai⋆ and E. Moulines, "*Non-convex Optimization with Frank-Wolfe Algorithm and Its Variants*", in *NIPS 2016 Workshop on Nonconvex Optimization for Machine Learning: Theory and Practice*, Dec., 2016. (⋆ equal contribution)

C9. H.-T. Wai, A. Scaglione, J. Lafond and E. Moulines, "*A Projection-free Decentralized Algorithm for Non-convex Optimization*", in Proc. GlobalSIP 2016, Dec., 2016.

C10. J. Lafond⋆, H.-T. Wai⋆ and E. Moulines, "*D-FW: Communication Efficient Dis-*

tributed Algorithms for High-dimensional Sparse Optimization", in Proc. ICASSP 2016, March, 2016. (* equal contribution)

C11. H.-T. Wai, T.-H. Chang and A. Scaglione, "*A Consensus-based Decentralized Algorithm for Non-convex Optimization with Application to Dictionary Learning*", in Proc. ICASSP 2015, Apr., 2015.

C12. H.-T. Wai and A. Scaglione, "*Decentralized Regression Under Asynchronous sub-Nyquist Sampling*", Invited paper, Asilomar Conf. 2014, Nov., 2014.

X2. J. Lafond, H.-T. Wai and E. Moulines, "*On the Online Frank-Wolfe Algorithms for Convex and Non-convex Optimizations*", Tech. report, ArXiv/1510.01171v2, Aug., 2016.

1.4 Prior Works

This section provides a high level overview of the prior works related to this dissertation. We shall, however, describe in detail the mathematical formulations of these prior work in the respective chapters.

1.4.1 On Network Identification

Understanding networks and inferring them using interaction data has been a long sought problem pursued by researchers of different disciplines, from physicists to bioinformatists, and from social scientists to machine learning scientists, etc. Prior methods for network inference can roughly be characterized into statistical learning and dynamics-based learning, whose prior arts are explored as follows.

In statistical learning methods for network topology inference, we model the interaction data as realizations of a random process parameterized by a statistical model describing the influence network. Among which the most popular approach is the celebrated graphical model [Wainwright *et al.*(2008)]. Here, the key idea is to rely on the *conditional independence* assumption pertaining to the causal relationship between the nodes — if a pair of nodes are *not connected*, then the random variables (r.v.s) defined on them are statistically independent when conditioned on a set of r.v.s defined on the graph cut between them. Importantly, this implies that the network topology can be captured by the support of the *inverse* of the covariance matrix. The observation above has leaded to the development of the famous Graphical LASSO formulation [Friedman *et al.*(2008)], which is essentially a penalized maximum likelihood method for the graphical model when the observations are Gaussian. As a matter of fact, the Graphical LASSO formulation results in a non-trivial convex optimization problem and its fast solution methods have been studied in [Friedman *et al.*(2008), Hsieh *et al.*(2014), d'Aspremont *et al.*(2008), Scheinberg *et al.*(2010), Yun *et al.*(2011)]; the theoretical guarantees of the formulation has also been studied in [Banerjee *et al.*(2008)]. Various models have been studied — [Segarra *et al.*(2016)] proposed to model the network data as graph signals observed as output of an unknown graph filter with white, full-rank excitation signals; [Tang *et al.*(2012b)] proposed a transfer factor graph model to infer links on heterogenous social networks, [Tang *et al.*(2012a)] proposed a trusts evolution model for product review data, [Bresler(2015)] considers Ising's model and proposed a greedy algorithm that is guaranteed to find the n-nodes graph using $\Omega(\log(n))$ samples, [Etesami *et al.*(2016), He *et al.*(2015)] studies the Hawkes model and model the actions of social agents as an arrival process, [Pouget-Abadie and Horel(2015)] considered the network inference problem from observing a cascading process. Another related approach utilizes the correlation scores between the nodes' values and the intuition that a pair of nodes are connected when they are correlated. They have largely been applied by practitioners on gene regulatory network (GRN) inference. These include methods such as random forest [Huynh-Thu *et al.*(2010)], ranking correlations [Küffner *et al.*(2012)], mutual information [Faith *et al.*(2007)], etc. In fact, the random forest based method (GENIE3) in [Huynh-Thu *et al.*(2010)] was the top performer in the DREAM5 network inference challenge [Marbach *et al.*(2012)]; also see the derivatives of GENIE3 reported in recent years [Petralia *et al.*(2015), Wu *et al.*(2016)]. Despite their success on GRN inference, there lacks a theoretical understanding on the limits of using such methods. It is worth mentioning that clustering algorithms such as Latent Dirichlet Allocation (LDA) or Probabilistic Latent Semantic Analysis (PLSA) have been adopted to infer networks such as citation graphs and social networks, see [Xiang *et al.*(2010), Dietz *et al.*(2007)].

Lastly, dynamics-based learning assumes that certain interaction data can be interpreted as the states of a dynamical system, where the latter is specified by a set of difference/differential equations. This fits into the hypothesis that the states of the agents/nodes in the networked system evolve with a fixed and known rule [Ronen *et al.*(2002)] and allows us to assign physical meaning to the network inference results. Like the statistical learning method described above, the general approach is to apply a linear regression to find the best fit network to the dynamical system. In the case of a sparse network, a sparsity enhancing regularizer, *i.e.,* the popular ℓ_1 norm, may also be adopted. These methods have been particularly successful in inferring networked systems with known physical properties, e.g., oscillator networks [Timme(2007)], epidemic networks [Shen *et al.*(2014)], social networks [Wang *et al.*(2011b), Han *et al.*(2015), Das *et al.*(2014)], gene regulatory networks [Yip *et al.*(2010), Bonneau *et al.*(2006)] and other networks [Ching *et al.*(2015), Shen *et al.*(2017)]. When the physical properties are not perfectly known, such method has shown success to certain degree when applied to GRN inference. Most of the above works focus on the network inference problem using *time series* data for improved identifiability condition, with the exception of [Sontag(2008), Timme(2007)]. Moreover, there lacks a rigorous mathematical analysis on the performance of the proposed method.

The gap filled in this thesis is to provide the theoretical underpinning of network inference problems in the case of *limited data.* For instance, the graphical model [Banerjee *et al.*(2008)] require the observed data to have rank that grows with the size of the network; and the dynamics-based learning approach [Timme(2007), Shen *et al.*(2017)] requires observing a large amount of the transient data. These assumptions can, however, be unrealistic since the actual network data are often of *low-rank*, as demonstrated in [DiMaggio *et al.*(1996), Brunet *et al.*(2004), Udell and Townsend(2017)]. Moreover, in social networks, the evolution of the opinions is latent, while the actions that result from the opinion matured by the agents are visible. As such, the network identification method proposed in this dissertation follows the dynamics-based learning approach with the following distinctions from the prior work: *(i)* our method requires only the *steady state* data, which are more robust with respect to an imperfect observation model; *(ii)* the observed network data can be of *low rank*, of which the rank is independent of the network size. Importantly, we proved that the network can still be perfectly recovered under such circumstances.

1.4.2 On Multi-agent Optimization

The study of multi-agent optimization has become popular starting from the 1980s, when it was motivated by performing distributed estimation on sensor network. Nowadays, with the wide spread adaptations of Internet-of-Things, the application of multi-agents optimization has evolved to handling the sophisticated problems emerging from machine learning applications using the now-powerful mobile devices (e.g., smart phones). In particular, the focus of recent researches has shifted from simply re-distributing computation load to more efficient handling of large dataset that is scattered within a network of computers.

Various authors have proposed decentralized optimization algorithms that are built on the average consensus (AC) protocol [Tsitsiklis(1984), Dimakis *et al.*(2010)]. Prior works include [Ram *et al.*(2012), Shi *et al.*(2015)] which studied the decentralized counterparts of projected gradient (PG) methods. These methods are simple to implement and are shown to converge for a large class of problems and network condition, e.g., on a time varying network. Moreover, [Yang *et al.*(2014), Lorenzo and Scutari(2016)] considered the successive convex approximation methods and a Jacobi-like scheme for decentralized optimization. The convergence properties and the performance of these algorithms were investigated extensively, especially when the objective is convex, see [Sayed *et al.*(2013), Li and Scaglione(2013), Ram *et al.*(2012), Wei and Ozdaglar(2013), Shi *et al.*(2015), Chang *et al.*(2014), Jakovetic *et al.*(2014), Duchi *et al.*(2012)]; for non-convex objectives, some recent results have been reported in [Hong(2016), Lorenzo and Scutari(2016), Bianchi and Jakubowicz(2013)]. A common trait found in the existing methods on the subject is that each iteration of these algorithms require at least one Euclidean *projection* operation. When the size of the problem is moderate, this projection step may be computed efficiently. When the problem involves a high dimensional parameter, the projection step may become computationally prohibitive, rendering most existing methods impractical.

The shortcomings with existing methods have called for a new paradigm known as the *projection-free* (a.k.a. Frank Wolfe, FW, conditional gradient) optimization. Compared to PG, the FW algorithm can often reduce the complexity by orders of magnitude for high dimensional problems. Owing to this, the FW algorithm has become increasingly popular in the machine learning community. However, the state-of-the-art works have only considered *centralized* and *convex* optimization problems, e.g., [Jaggi(2013),Lacoste-Julien and Jaggi(2015)]. The exceptions are [Bellet *et al.*(2015)] which considered a distributed FW algorithm for convex problems with a specific structure; or [Reddi *et al.*(2016),Lacoste-Julien(2016),Yu *et al.*(2014)] which analyzed the centralized FW algorithm for non-convex problems. In contrast, this dissertation studies a decentralized FW algorithm for the general joint optimization problem for convex and non-convex optimization.

Besides mimicking classical optimization tools such as gradient and Newton methods, applying decomposition techniques such as alternating optimization (AO) on networks is also attractive as they are often effective in handling complicated problem structures. To this end, decentralized primal-dual/ADMM algorithms are recently considered in [Chang *et al.*(2014),Aybat and Hamedani(2016),Hong(2016)], as well as the heuristics in [Chainais and Richard(2013)]. Compared to PG-based methods, these methods are able to efficiently handle more complicated constraints. However, except for [Hong(2016)], the convergence guarantees of these method are limited to the case of convex optimization, while many interesting applications for AO algorithms involve non-convex optimization. Examples are problems with dictionary learning or matrix factorization. To close this gap, this dissertation proposes new AO based algorithms for decentralized optimization and provides strong evidence of its convergence in non-convex problems.

2 Mathematical Preliminaries and Notations

This chapter presents standard notations and definitions for the mathematical objects used throughout this dissertation. We group these into three topics — networks and graphs, linear algebra and mathematical analysis. For easy reference, we state a few standard results that will be used later on, and provide proofs to them when appropriate.

2.1 Networks and Graph Theory

The network of interest is represented by a simple directed graph $G = (V, E)$, where V denotes the set of nodes and $E \subset V \times V$ denotes the set of edges. We orient the edges such that $(i, j) \in E$ denotes there is an edge pointing *from* node i to node j. Unless otherwise specified, we assume that $(i, i) \notin E$. The graph is strongly connected if for any pair $i, j \in V$, there exists a path p_{ij} that connects i to j. The path p_{ij} from i to j is defined as an ordered set of edges from E such that $p_{ij} = \{(i, v_1), (v_2, v_3), \ldots, (v_p, j)\}$, where $(i, v_1), (v_p, j) \in E$ and $(v_k, v_{k+1}) \in E$ for all $k = 1, ..., p-1$. A path that begins and ends at the same node is called a cycle. When the graph G is undirected, G is said to have a perfect matching if there exists $M \subseteq E$ such that no two edges in M have a common node and all nodes in V are incident to an edge in M.

The node set of a bipartite graph $G_{bi} = (V_{bi}, E_{bi})$ can be decomposed such that $V_{bi} = A \cup B$ and $A \cap B = \emptyset$. The bipartite graph *does not* contain any edge within node set A or B, or equivalently, it does not contain any cycle of odd length. We can determine if a bipartite graph with $|A| = |B|$ has a perfect matching by the following Hall's theorem:

Theorem 2.1 (Hall's Theorem [West(2000)]) *Let $W \subseteq A$, we denote $\mathcal{N}(W) \subseteq B$ as the neighborhood set of W,* i.e., $\mathcal{N}(W) := \cup_{w \in W}\{j : (w, j) \in E_{bi}\}$. *The bipartite graph G_{bi} has a perfect matching if and only if*

$$|W| \leq |\mathcal{N}(W)|, \ \forall \ W \subseteq A \ . \tag{2.1}$$

The concept of perfect matching is related to the expander theory that will be used in Chapter 3.

Lastly, the graph G is associated with a square weighted adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{|V|\times|V|}$ which measures the interaction strengths among nodes. We have $A_{ji} \neq 0$ if $(i,j) \in E$ such that the support of $\boldsymbol{A}$ encodes the network topology completely. Similarly, the bipartite graph G_{bi} is associated with a non-square weighted adjacency matrix $\boldsymbol{B} \in \mathbb{R}^{|B|\times|A|}$ defined in a similar way as $\boldsymbol{A}$. Further conditions on $V, E, \boldsymbol{A}, \boldsymbol{B}$ will be described in the context when we consider the specific applications.

2.2 Linear Algebra

For any natural number $n \in \mathbb{N}$, we denote $[n]$ as the set $\{1, 2, ..., n\}$. Vectors (*resp.* matrices) are denoted by boldfaced letters (*resp.* capital letters). We denote x_i as the ith element of the vector $\boldsymbol{x}$, $[\boldsymbol{E}]_{\mathcal{S},:}$ (*resp.* $[\boldsymbol{E}]_{:,\mathcal{S}}$) denotes the submatrix of $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ with only the rows (*resp.* columns) selected from $\mathcal{S} \subseteq [m]$ (*resp.* $\mathcal{S} \subseteq [n]$). Vector $\mathbf{e}_k \in \mathbb{R}^n$ is a unit vector with zeros everywhere except for the kth coordinate and $\mathbf{1}$ is an all-ones vector with suitable dimension. The superscript $(\cdot)^\top$ denotes matrix/vector transpose. $\|\cdot\|_2$ denotes the Euclidean norm, $\|\cdot\|_1$ is the ℓ_1-norm and $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^\top \boldsymbol{y}$ is the inner product. The binary operator $\odot$ denotes element wise product such that $[\boldsymbol{x} \odot \boldsymbol{y}]_i = x_i y_i$ for all i.

Consider the generic matrices $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{m_1 \times m_2}$, their inner product is defined as $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{Y})$ and $\|\boldsymbol{X}\|_F = \sqrt{\langle \boldsymbol{X}, \boldsymbol{X} \rangle}$ denotes its Frobenius norm. Moreover, $\boldsymbol{X}$ admits a singular value decomposition such that $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Here $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and $\boldsymbol{\Sigma}$ is a $\tilde{m} \times \tilde{m}$, non-negative diagonal matrix with $\tilde{m} = \min\{m_1, m_2\}$ and diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\tilde{m}} \geq 0$. The Schatten-$p$ norm of $\boldsymbol{X}$ is defined as $\|\boldsymbol{X}\|_{\sigma,p} = \big(\sum_{i=1}^{\tilde{m}} \sigma_i^p \big)^{1/p}$; and the spectral norm $\|\boldsymbol{X}\|_2$ is the maximum singular value of $\boldsymbol{X}$, *i.e.,* $\sigma_1(\boldsymbol{X})$. The vectorization of a matrix $\boldsymbol{X} \in \mathbb{R}^{m_1 \times m_2}$ is denoted by $\mathrm{vec}(\boldsymbol{X}) = [\boldsymbol{x}_1; \boldsymbol{x}_2; \ldots; \boldsymbol{x}_{m_2}] \in \mathbb{R}^{m_1 m_2}$ such that $\boldsymbol{x}_i$ is the ith column of $\boldsymbol{X}$. We use $\mathrm{Diag} : \mathbb{R}^n \rightarrow \mathbb{R}^{n\times n}$ (*resp.* $\mathrm{diag} : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^n$) to denote the diagonal operator that maps from a vector (*resp.* square matrix) to a diagonal square matrix (*resp.* vector) and $\geq$ to represent the element-wise inequality. The range space $\mathcal{R}(\boldsymbol{X})$ and the null space $\text{null}(\boldsymbol{X})$ of the matrix $\boldsymbol{X}$ are defined as:

$$\mathcal{R}(\boldsymbol{X}) := \{\boldsymbol{y} \in \mathbb{R}^{m_1} : \boldsymbol{y} = \boldsymbol{X}\boldsymbol{z} \text{ for some } \boldsymbol{z} \in \mathbb{R}^{m_2}\}, \quad \text{null}(\boldsymbol{X}) := \{\boldsymbol{z} \in \mathbb{R}^{m_2} : \boldsymbol{X}\boldsymbol{z} = \boldsymbol{0}\} .$$

When $\boldsymbol{X}$ is a square matrix, *i.e.,* when $m = m_1 = m_2$, then the matrix also admits an eigenvalue decomposition such that $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{-1}$ where $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_m]$ consists of the m eigenvectors and $\boldsymbol{\Lambda}$ is a diagonal matrix with elements $\lambda_1, \ldots, \lambda_m$ such that $\boldsymbol{X}\boldsymbol{q}_i = \lambda_i \boldsymbol{q}_i$. The spectral radius of $\boldsymbol{X}$ is defined as $\rho(\boldsymbol{X}) := \max\{|\lambda_1|, \ldots, |\lambda_m|\}$. If $\boldsymbol{X}$ is non-negative, then $\rho(\boldsymbol{X}) \leq \max_{1 \leq i \leq n} \sum_{j=1}^{n} X_{ij}$ [Horn and Johnson(1986), Theorem 8.1.22].

In this dissertation, we are particularly interested in the weighted adjacency matrix $\boldsymbol{A}$ defined with respect to a graph G and its spectrum when it is non-negative. The following result is a consequence of the Perron-Frobenius Theorem:

Theorem 2.2 (Theorem 6.2.24, 8.4.4 of [Horn and Johnson(1986)]) *If G is a strongly connected graph and its weighted adjacency matrix $\boldsymbol{A}$ is non-negative, then:*

1. *the spectral radius $\rho(\boldsymbol{A}) > 0$ and it is an eigenvalue of $\boldsymbol{A}$.*
2. *there is a positive vector $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} = \rho(\boldsymbol{A})\boldsymbol{x}$.*
3. *$\rho(\boldsymbol{A})$ is an algebraically simple eigenvalue of $\boldsymbol{A}$,* i.e., *the positive vector $\boldsymbol{x}$ above is the unique vector satisfying $\boldsymbol{A}\boldsymbol{x} = \rho(\boldsymbol{A})\boldsymbol{x}$.*

In addition, if $\boldsymbol{A}$ is a stochastic matrix, such that $\boldsymbol{A}\boldsymbol{1} = \boldsymbol{1}$, and it satisfies the conditions in Theorem 2.2, then $\rho(\boldsymbol{A}) = 1$ and the largest eigenvalue of $\boldsymbol{A}$ is 1, with the vector $\boldsymbol{1}$ being the corresponding eigenvector.

Now, if the conditions in Theorem 2.2 are satisfied, and the weighted adjacency matrix $\boldsymbol{A}$ is symmetric and doubly stochastic, then the following fact holds:

Fact 2.1 *Let $\boldsymbol{x}_1, ..., \boldsymbol{x}_N \in \mathbb{R}^d$ be a set of N vectors and $\boldsymbol{x}_{avg} := N^{-1} \sum_{i=1}^{N} \boldsymbol{x}_i$ be their average. Suppose $\boldsymbol{A}$ is a doubly stochastic, non-negative matrix and the conditions in The-*

orem 2.2 are satisfied. Then the output after performing one round of average consensus (AC) update:

$$\overline{\boldsymbol{x}}_i = \sum_{j=1}^{N} A_{ij} \cdot \boldsymbol{x}_j \tag{2.2}$$

must satisfy

$$\sqrt{\sum_{i=1}^{N} \|\overline{\boldsymbol{x}}_i - \boldsymbol{x}_{avg}\|^2} \leq \sigma_2(\boldsymbol{A}) \cdot \sqrt{\sum_{i=1}^{N} \|\boldsymbol{x}_i - \boldsymbol{x}_{avg}\|^2} \ , \tag{2.3}$$

where $\sigma_2(\boldsymbol{A}) < 1$.

Notice that $\boldsymbol{A}$ corresponds to the *mixing matrix* required in the average consensus (AC) protocol [Tsitsiklis(1984)]. Repeatedly applying (2.3) shows the well known fact that the AC protocol computes the network average at a geometric rate.

2.3 Mathematical Analysis

We describe some basic concepts of mathematical analysis below. Let G, L, μ be some non-negative constants and $\| \cdot \|$ be a norm defined on $\mathbb{R}^d$. Consider a function $f : \mathbb{R}^d \to \mathbb{R}$,

- the function f is G-Lipschitz if for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$

  $$|f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}')| \leq G\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\star \ , \tag{2.4}$$

  where $\| \cdot \|_\star$ is the dual norm of $\| \cdot \|$;

- the function f is L-smooth if for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$

  $$f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}') \leq \langle \nabla f(\boldsymbol{\theta}'), \boldsymbol{\theta} - \boldsymbol{\theta}' \rangle + \frac{L}{2}\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2 \ , \tag{2.5}$$

  where $\nabla f(\boldsymbol{\theta})$ is the gradient of f evaluated at $\boldsymbol{\theta}$, note that the above is equivalent to $\|\nabla f(\boldsymbol{\theta}') - \nabla f(\boldsymbol{\theta})\|_2 \leq L\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|_2$;

- the function f is μ-strongly convex if for all $\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$,

$$f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}') \leq \langle \nabla f(\boldsymbol{\theta}), \boldsymbol{\theta} - \boldsymbol{\theta}' \rangle - \frac{\mu}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2 \, , \tag{2.6}$$

notice that if $\mu = 0$, the above definition reduces to that of stating that f is convex.

We also state the following descent Lemma pertaining to the proximal operator:

Lemma 2.1 *Suppose that f is L'-Lipschitz and the step size β satisfies $\beta \leq 1/L'$, then for any potentially non-smooth g and $\boldsymbol{x} \in \mathrm{dom}(f+g)$, it holds that*

$$f(\boldsymbol{y}) + g(\boldsymbol{y}) \leq f(\boldsymbol{x}) + g(\boldsymbol{x}) - \frac{1}{2\beta} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 \, , \tag{2.7}$$

where

$$\boldsymbol{y} = \mathrm{prox}_{\beta g(\cdot)} \Big(\boldsymbol{x} - \beta \nabla f(\boldsymbol{x}) \Big) := \arg\min_{\boldsymbol{z}} \; \Big(g(\boldsymbol{z}) + \frac{1}{2\beta} \|\boldsymbol{z} - (\boldsymbol{x} - \beta \nabla f(\boldsymbol{x}))\|_2^2 \Big) \, . \tag{2.8}$$

Proof: Consider [Beck and Teboulle(2009), Lemma 2.3]. We assign $L = 1/\beta$, $\mathbf{x} = \boldsymbol{x}$, $F(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x})$ and observe that our variable $\boldsymbol{y}$ can be calculated by $p_L(\mathbf{x})$ in the cited lemma. Since f is L'-Lipschitz, we observe that:

$$F(p_L(\mathbf{x})) \leq g(p_L(\mathbf{x})) + f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), p_L(\mathbf{x}) - \mathbf{x} \rangle + \frac{L'}{2} \|p_L(\mathbf{x}) - \mathbf{x}\|_2^2 \leq Q(p_L(\mathbf{x}), \mathbf{x}) \, , \tag{2.9}$$

where $Q(\cdot)$ is also defined in the cited lemma. Therefore, the required condition is satisfied and applying the conclusion in the cited lemma yields (2.7). **Q.E.D.**

Lastly, for some positive constants C_1, C_2, C_3, $C_2 \leq C_3$ and non-negative functions $f(t), g(t)$, the notations $f(t) = \mathcal{O}(g(t))$, $f(t) = \Theta(g(t))$ indicate $f(t) \leq C_1 g(t)$, $C_2 g(t) \leq f(t) \leq C_3 g(t)$ for sufficiently large t, respectively.

— PART I —

Modeling and Identification of Networks

3 Network RADAR for Opinion Dynamics

This chapter is the first part of our expositions on the *modeling and identification* of network dynamics. Specifically, we focus on modeling the opinion dynamics on social networks and discuss the methodology for identifying the network underneath it.

3.1 Context and Background

Modeling opinion dynamics on social networks is a century old problem, with the original research dating back to the beginning 20th century, focusing on explaining the phenomena of *crowd wisdom* [Galton(1907)]. However, the *network structure* wasn't considered in opinion dynamics model until the 1970s when the DeGroot model [DeGroot(1974)] was introduced. The DeGroot model postulates that agents are *influenced* by their immediate neighbors and *updates* their opinions by calculating a *convex and weighted* combination of the *difference* between the neighbors' opinions and own's opinion. The intuition behind the model is that the agent belief is a random mixture model of its neighbors belief he/she had before interacting. From this, several opinion dynamics models with *nonlinear* interactions are considered, e.g., the Hegselmann-Krause model [Hegselmann and Krause(2002)], the voter model [Holley and Liggett(1975)], etc. Another interesting direction is to consider the existence of *stubborn agents* in the network, *i.e.,* sociopaths who only trust their own opinion [Mobilia(2003), Mobilia *et al.*(2007), Acemoglu *et al.*(2013), Yildiz *et al.*(2013)]; the influences of these stubborn agents are also studied under the context of DeGroot model in [Acemoglu *et al.*(2010), Yildiz and Scaglione(2010), Acemoglu *et al.*(2013)]. Such models give a plausible explanation for the *non-consensual* behaviors exhibited in the actual social networks. In this context, this chapter provides a unified view on the opinion dynamics models and the steady state opinions resulted from it. Our model encompasses the DeGroot model and a few nonlinear opinion dynamics model derived from it. In particular, we analyzed the steady-state opinions resulted from these models when the social networks contains a set of *stubborn agents*.

The identification problem of social network has attracted much attention as motivated by recent studies on how the decision making process of an individual is affected by the network structure [Kempe *et al.*(2003), Candogan *et al.*(2012)], as well as the availability of network data due to the digitalized online social network platforms (e.g., Facebook and Twitter). Recent advances in this direction can be found in [Moussaïd *et al.*(2013), Das *et al.*(2014), De *et al.*(2014)], where the previous work assumed that the evolutions of opinions are observed at different time instances, giving a rich set of observations on the network. To relax the assumptions on *transient* observations, the celebrated graphical model also lead to a sound heuristics following the conditional independence assumptions. However, most of the above work rely on identifying the network from a set of *full rank* data which can only be achieved when the network is excited at all nodes *independently* and a sufficient amount of data is accrued. To this end, this chapter deals with a laxer assumption employing *low-rank* network data that is induced by a small number of stubborn agents in the social network. Importantly, we provide a provable guarantee on the recoverability of the network in terms of the observation rank in the steady state data.

In the following, Section 3.2 sets up the system model as well as a mathematical description of the perturbation experiments. Then, Section 3.3 outlines the network identification method employed; Section 3.4 summarizes the proven recoverability conditions. Section 3.5 describes a related community detection method based on the low rank opinion data observed. Lastly, we present results from our numerical experiments in Section 3.6.

3.2 Opinion Dynamics Model

Following the notations defined in Section 2.1 about networks, the social network of interest G has n agents such that $V = [n]$ and the weights of $\boldsymbol{A}$ represents the strength of *trust* between agents. We allow self-edge in this chapter such that $(i, i) \in E$ for all i. The matrix is non-negative and it is normalized such that it is stochastic, *i.e.,* $\boldsymbol{A}\mathbf{1} = \mathbf{1}$. From now on, we shall refer to this matrix as the *trust matrix*. We consider K different discussions among the n agents. Each discussion is indexed by $k \in [K]$. The opinion of the ith agent at discrete time t is denoted by a scalar $x_i(t;k)$, for example, the ith agent's opinion $x_i(t;k)$ may represent a probability distribution function of his/her stance on the discussion[1], at time $t \in \mathbb{N}$ during the kth discussion. As the individuals' opinions are constantly influenced by opinions of the others, we model the opinion dynamics with an iterative process:

$$x_i(t+1;k) = x_i(t;k) + \sum_{j=1}^{n} A_{ij} h_i\big(x_j(t;k) - x_i(t;k)\big) \; , \tag{3.1}$$

where $h_i(\cdot)$ is a response function taking the following form:

$$h_i(x-y) = \mu_i(|x-y|) \cdot (x-y) \; , \tag{3.2}$$

and $\mu_i : \mathbb{R}_+ \to \mathbb{R}_+$ measures the opinion distance and it satisfies the following:

$$\textit{(a).}\ \mu_i(0) = 1, \quad \textit{(b).}\ \mu_i(x) \geq 0, \quad \textit{(c).}\ \mu_i(x) \textit{ is non-increasing in } x \; .$$

The model in (3.1) says that the opinion update for the ith agent is equal to a weighted sum of the difference between the ith agent's opinion and his/her friends, scaled by a non-linear function μ. The non-increasing property of the function μ represents the fact that the influence from one agent to another shall not increase if the two agents' opinion are further apart. A number of classical opinion dynamics model can be described by (3.1). For example, the celebrated DeGroot model can be recovered from (3.1) by setting $\mu(x) = 1$ for all x see [DeGroot(1974)] and [Friedkin and Johnsen(2011), Chapter 1] for a detailed description of the application of the aforementioned model in social networks; the bounded confidence model [Hegselmann and Krause(2002)] can also be recovered by setting $\mu(x) = 1$ for $x < \tau$ and $\mu(x) = 0$ for $x \geq \tau$, where τ is some threshold value. To simplify the analysis,

[1] While our discussion is focused on the case when $x_i(t;k)$ is a scalar, it should be noted that the techniques developed can be easily extended to the vector case.

we can rewrite Eq. (3.1) as

$$\boldsymbol{x}(t+1;k) = \Big(\mathrm{Diag}(\mathbf{1} - \big(\boldsymbol{A} \odot \boldsymbol{\mu}(\boldsymbol{x}(t;k))\big)\mathbf{1}) + \big(\boldsymbol{A} \odot \boldsymbol{\mu}(\boldsymbol{x}(t;k))\big)\Big)\ \boldsymbol{x}(t;k)\ , \tag{3.3}$$

where we stacked the vectors as $\boldsymbol{x}(t;k) = (x_1(t;k), \ldots, x_n(t;k))^\top$, denoted $[\boldsymbol{A}]_{ij} = A_{ij}$ and $\boldsymbol{\mu}(\boldsymbol{x}(t;k))$ is a symmetric, non-negative matrix given as:

$$\big[\boldsymbol{\mu}(\boldsymbol{x}(t;k))\big]_{ij} := \begin{cases} \mu_i\big(|x_i(t;k) - x_j(t;k)|\big), & \text{if } i \neq j \text{ and } (j,i) \in E\ , \\ 0, & \text{if } i = j \text{ or } (j,i) \notin E\ . \end{cases} \tag{3.4}$$

It can be proven that:

Observation 3.1 *Suppose that the graph G is strongly connected and $\mu_i(x) > 0$ for all i, x, then with some $c(k)$ that depends on μ_i and $\boldsymbol{x}(0;k)$, the steady state opinions satisfies*

$$\lim_{t\to\infty} \boldsymbol{x}(t;k) = c(k)\mathbf{1}\ . \tag{3.5}$$

Note that the above is a well known fact in the distributed control literature [Blondel *et al.*(2005)] for the case of linear DeGroot dynamics, for the nonlinear case, similar result has also been shown in [Li *et al.*(2011b)]. For completeness, a simple proof is provided in Appendix 3.A.

We note that the results described in Observation 3.1 may be unrealistic as *consensus* is seldom reached in actual social network. Furthermore, for the purpose of retrieving $\boldsymbol{A}$ from the observed opinions, the observation above clearly indicates that the *steady-state* opinions cannot help reveal any information about the network structure. In fact, most prior works on the topic have required complete/partial knowledge of $\mathcal{T}$ such that the opinion dynamics are trackable. In addition, [De *et al.*(2014)] infers $\boldsymbol{A}$ by solving a simple least square problem using a linear dynamics model; [Timme(2007),Wang *et al.*(2011b)] deal with a time-varying, non-linear dynamics model and apply sparse recovery to infer $\boldsymbol{A}$. The drawback of these methods is that they require knowing the discrete time stamps for the observations made. This knowledge may be difficult to obtain in general since the actual system states are updated with an *unknown* interaction rate and the interaction timing between agents is not observable in most practical scenarios.

To model the *non-consensual* behavior in social networks, we consider the effects of *stubborn agents*, *i.e.,* individuals whose opinions cannot be influenced by the others, to the steady state of the opinion dynamics described. As we shall see next, the resulting steady states depend on the initial opinions of the stubborn agents. In this way, the stubborn agents can be seen as introducing *perturbations* into the social network. The steady states characterized will then be used for the network identification.

Before we move on, we remark that it is possible to consider a setting with a time varying trust matrix of random connectivity. In fact, our analysis can be extended as one replaces $\boldsymbol{A}$ in the previous equation with a random matrix $\boldsymbol{A}(t;k)$, which satisfies the following:

H3.1 *The matrix* $\boldsymbol{A}(t;k)$ *is an independently and identically distributed (i.i.d.) random matrix drawn from a distribution satisfying* $\mathbb{E}[\boldsymbol{A}(t;k)] = \boldsymbol{A}$ *for all* $t \in \mathbb{N}, k \in [K]$, *where the expectation is taken* w.r.t. *the distribution of* $\boldsymbol{A}(t;k)$.

However, for simplicity, we focus on the static case from now on. A random opinion dynamics setting will be revisited later.

3.2.1 Effects of Stubborn Agents

We consider extending the social network G by appending S *stubborn agents* into the social network. Formally, stubborn agents (a.k.a. zealots) are members of a social network whose opinions can not be swayed by others. If agent i is stubborn, then $x_i(t;k) = x_i(0;k)$ for all t. Adapting to the DeGroot opinion dynamics, these agents can be characterized by the structure of their respective rows in the trust matrix:

Definition 3.1 *An agent* i *is stubborn if and only if its corresponding row in the trust*

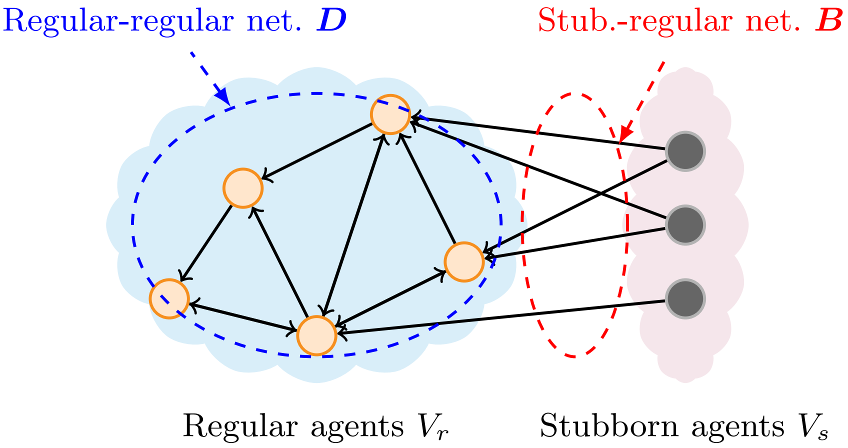
Regular-regular net. $\boldsymbol{D}$
Stub.-regular net. $\boldsymbol{B}$
Regular agents V_r
Stubborn agents V_s

Figure 3.1: Illustration of the social network with stubborn agents and the sub-networks therein.

matrix $\boldsymbol{A}$ *is the canonical basis vector;* i.e., *for all* j*,*

$$A_{ij} = \begin{cases} 1, & \text{if } j = i, \\ 0, & \text{otherwise .} \end{cases} \tag{3.6}$$

The extended social network G' consists of $n + S$ agents, denoted by $V' = [n + S]$ and the edge set is denoted by $E' \subseteq V' \times V'$. Without loss of generality, we let $V_s := [S]$ be the set of stubborn agents and $V_r := V' \setminus V_s = \{1 + S, ..., n + S\}$ be the set of regular agents. The trust matrix $\boldsymbol{A}$ can thus be partitioned as follows:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{D} \end{pmatrix} , \tag{3.7}$$

where $\boldsymbol{B} \in \mathbb{R}^{n \times S}$ and $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ are the sub matrices of $\boldsymbol{A} \in \mathbb{R}^{(n+S) \times (n+S)}$. The matrix $\boldsymbol{B}$ captures the sub-network between stubborn and regular agents, and $\boldsymbol{D}$ captures the sub-network among the regular agents themselves. See Figure 3.1 for an illustration of the sub-networks involved. We further impose the following assumptions:

H3.2 *Each agent in* V_r *has non-zero trust in at least one agent in* V_s*.*

H3.3 *The induced subgraph $G'[V_r] = (V_r, E'(V_r))$ is strongly connected.*

It can be shown that the two assumptions above imply that the principal submatrix $\boldsymbol{D}$ satisfies $\|\boldsymbol{D}\|_2 < 1$. We are interested in the steady state opinion resulting from (3.1) at $t \to \infty$, which can be characterized using the observation below. Define the shorthand notation $\boldsymbol{x}^k := \boldsymbol{x}(\infty; k)$ and the partition that $\boldsymbol{x}^k = (\boldsymbol{z}^k; \boldsymbol{y}^k)$ such that $\boldsymbol{z}^k$ (*resp.* $\boldsymbol{y}^k$) corresponds to the steady-state opinions of the stubborn agents (*resp.* regular agents), we have:

Lemma 3.1 *Letting $t \to \infty$ in (3.1), we have:*

$$\Big(\mathrm{Diag}(\boldsymbol{f}^k) - \boldsymbol{D} \odot \boldsymbol{\mu}(\boldsymbol{y}^k)\Big)\boldsymbol{y}^k = \Big(\boldsymbol{B} \odot \tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)\Big)\boldsymbol{z}^k \quad \textit{and} \quad \boldsymbol{z}^k = \boldsymbol{z}(0; k) \ , \tag{3.8}$$

where

$$\boldsymbol{f}^k := \big(\boldsymbol{D} \odot \boldsymbol{\mu}(\boldsymbol{y}^k)\big)\mathbf{1} + \big(\boldsymbol{B} \odot \tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)\big)\mathbf{1} \ , \tag{3.9}$$

and $\boldsymbol{\mu}(\boldsymbol{y}^k)$ is defined in (3.4)*. The matrix $\tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)$ is defined as*

$$\big[\tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)\big]_{ij} = \mu(|y_i^k - z_j^k|) \ . \tag{3.10}$$

The proof can be found in Appendix 3.B. Notice that the above characterization leads to a nonlinear system of equations in $\boldsymbol{x}(\infty; k)$ which could be solved using a fixed point iteration. However, it does not lead to a closed form solution in general. Nevertheless, the stubborn agents introduce *perturbations* to the steady states of the opinion dynamics systems and reveal structures of the social network in the steady states. Next, we discuss two interesting observations on Lemma 3.1 for the special case with linear DeGroot opinion dynamics, *i.e.,* when $\mu(x) = 1$ for all x.

Low Rank Steady-State Data. When restricted to the special case when $\mu(x) = 1$ for all x, *i.e.,* the linear DeGroot dynamics, Lemma 3.1 instead provides a closed form expression for the steady state opinions. Specifically, the steady state opinions of regular agents at

$t \to \infty$ can be written as:

$$\boldsymbol{y}^k = (\boldsymbol{I} - \boldsymbol{D})^{-1} \boldsymbol{B} \boldsymbol{z}^k \ . \tag{3.11}$$

We observe:

- The steady state opinions depend on the stubborn agents and the structure of the network. Unlike the case *without* stubborn agents, the information about the network structure $\boldsymbol{D}, \boldsymbol{B}$ is retained in (3.11).
- The range space of $\boldsymbol{A}^\infty$, where $\boldsymbol{A}^\infty = \lim_{t\to\infty} A^t$, has a dimension of at most S only. Since the number of stubborn agents S is usually much less than the number of regular agents, this implies that the steady-state opinion also lies in a low dimensional space. Importantly, we see that the existence of stubborn agents in the DeGroot opinion dynamics naturally gives rise to a *low rank* structure in the steady-state opinion data.

Diagonal Ambiguity. Under the linear DeGroot opinion dynamics setting, we notice that there is an intrinsic diagonal ambiguity for the linear equation in $\boldsymbol{B}, \boldsymbol{D}$ of Lemma 3.1, as characterized by the following:

Lemma 3.2 *Let $\boldsymbol{\ell} \in \mathbb{R}^n$ be such that $\boldsymbol{0} \leq \boldsymbol{\ell} < \boldsymbol{1}$ and define the matrix pair $(\boldsymbol{B_\ell}, \boldsymbol{D_\ell})$ such that:*

$$\boldsymbol{D_\ell} = \mathrm{Diag}(\boldsymbol{\ell}) + \mathrm{Diag}\left(\frac{\boldsymbol{1} - \boldsymbol{\ell}}{\boldsymbol{1} - \mathrm{diag}(\boldsymbol{D})}\right) \mathrm{off}(\boldsymbol{D}), \quad \boldsymbol{B_\ell} = \mathrm{Diag}\left(\frac{\boldsymbol{1} - \boldsymbol{\ell}}{\boldsymbol{1} - \mathrm{diag}(\boldsymbol{D})}\right) \boldsymbol{B} \ , \tag{3.12}$$

where $\mathrm{off}(\boldsymbol{D}) = \boldsymbol{D} - \mathrm{Diag}(\mathrm{diag}(\boldsymbol{D}))$ *and the fraction inside the bracket is an element-wise division. Then it holds that $(\boldsymbol{I} - \boldsymbol{D})^{-1}\boldsymbol{B} = (\boldsymbol{I} - \boldsymbol{D_\ell})^{-1}\boldsymbol{B_\ell}$ for all $\boldsymbol{\ell}$.*

Once again, we relegate the proof to Appendix 3.C. In fact, the ambiguity stated above can be understood as the loss of information on the *rate of convergence* in the opinion dynamics, which is natural as the data $\boldsymbol{y}^k, \boldsymbol{z}^k$ recorded are steady state opinions.

Lemma 3.2 implies that the equality (3.8) can be satisfied by infinitely many pairs of $(\boldsymbol{B_\ell}, \boldsymbol{D_\ell})$, each with a different diagonal entries $\boldsymbol{\ell}$ in the square matrix $\boldsymbol{D_\ell}$. As a remedy,

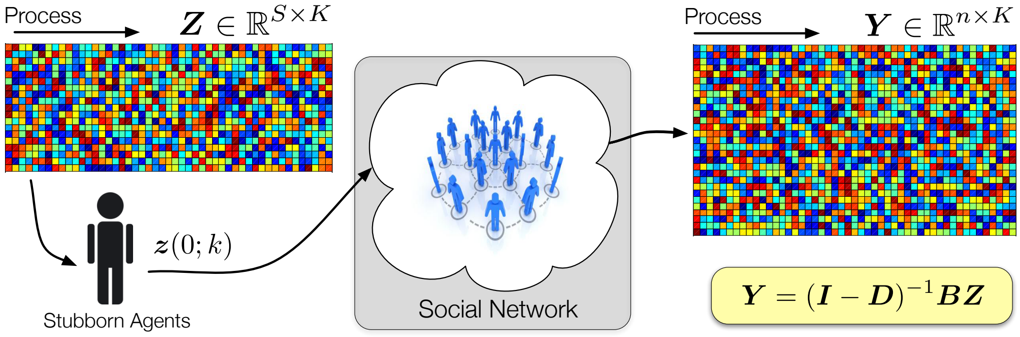

Figure 3.2: Data acquisition for opinion dynamics identification.

we choose to emphasize the degree of 'openness' of the agents. We consider the normalized network $(\boldsymbol{B}_{\boldsymbol{\ell}}, \boldsymbol{D}_{\boldsymbol{\ell}})$ with $\boldsymbol{\ell} = \boldsymbol{0}$, written as:

$$\boldsymbol{D}_0 := \text{Diag}\left(\frac{\boldsymbol{1}}{\boldsymbol{1} - \text{diag}(\boldsymbol{D})}\right) \text{off}(\boldsymbol{D}), \quad \boldsymbol{B}_0 := \text{Diag}\left(\frac{\boldsymbol{1}}{\boldsymbol{1} - \text{diag}(\boldsymbol{D})}\right) \boldsymbol{B}\ . \tag{3.13}$$

Notice that both the network topology and relative strengths of interaction between agents are preserved, regardless of the chosen $\boldsymbol{\ell}$. From now on, we shall use $(\boldsymbol{B}, \boldsymbol{D})$ to denote the normalized network $(\boldsymbol{B}_0, \boldsymbol{D}_0)$ to keep the presentation simple to follow.

Nevertheless, Lemma 3.1 motivates us to formulate a regression problem that estimates $\boldsymbol{D}, \boldsymbol{B}$ from the observed steady-state opinions, as described in the next section.

3.3 Opinion Dynamics Identification

We now study the problem of network identification using stubborn agents. Instead of tracking the state evolution in the network similar to [De *et al.*(2014), Timme(2007), Wang *et al.*(2011b), Shen *et al.*(2017)], our method is based on collecting the *steady-state opinions* from $K \geq S$ discussions and fitting them into the equilibrium equations governed by the dynamics. In particular, our input data can be described as the collection $\{\boldsymbol{y}^k, \boldsymbol{z}^k\}_{k=1}^K$ of steady state opinions.

Graphical LASSO. Before describing the proposed method, let us take a short detour by describing a popular method for network identification with steady-state data. The graphical LASSO (gLASSO) is a method introduced in [Friedman *et al.*(2008)] for inferring the latent structure of the random variables (r.v.s) generated from a Gaussian Markov random field (a.k.a. undirected graphical model) [Wainwright *et al.*(2008)]. The method relies on the following observation — consider a random vector $\boldsymbol{X} \in \mathbb{R}^n$ generated from a graphical model with $G = (V, E)$ as the underlying graph and X_i is an r.v. associated with node $i \in V$. Assume that the covariance matrix $\boldsymbol{C}_X$ of $\boldsymbol{X}$ is full rank. If $(i, j) \notin E$ and $S \subseteq V$ is a graph cut between them, then X_i, X_j are independent when conditioned on $(X_s)_{s \in S}$. Furthermore, the conditional independence property can be captured by the support of the inverse of covariance matrix such that $[\boldsymbol{C}_X^{-1}]_{ij} = 0$. In light of this, [Friedman *et al.*(2008)] proposed the following Graph LASSO optimization:

$$\min_{\boldsymbol{A} \in \mathbb{R}^{n \times n}} \quad - \log \det \boldsymbol{A} + \operatorname{Tr}(\hat{\boldsymbol{C}}_X \boldsymbol{A}) + \rho \| \operatorname{vec}(\boldsymbol{A}) \|_1 \ \text{ s.t. } \ \boldsymbol{A} = \boldsymbol{A}^\top \,, \tag{3.14}$$

where $\rho > 0$ is a regularization parameter, $\hat{\boldsymbol{C}}_X$ is the empirical covariance matrix of $\boldsymbol{X}$ that approximates the inverse of $\boldsymbol{C}_X$ and therefore the connectivity of the graph G. The gLASSO problem (3.14) is essentially a penalized maximum likelihood method for the graphical model. In particular, if we set the regularization parameter as $\rho = \Theta(1/\sqrt{k})$ where k is the number of samples obtained for estimating $\boldsymbol{C}_X$, then the latent graph structure can be recovered in high probability [Banerjee *et al.*(2008)].

The Graph LASSO might be applied as a heuristic to estimate the latent network structure of the social network by operating on the covariance of the regular agents' opinions. To obtain insights on its performance, let us assume the linear opinion dynamics setting and that the initial opinions of the stubborn agents are white, *i.e.,* $\mathbb{E}[\boldsymbol{z}^k (\boldsymbol{z}^k)^\top] = \boldsymbol{I}$. The covariance matrix of the regular agents' opinions is:

$$\boldsymbol{C}_y = \mathbb{E}[\boldsymbol{y}^k (\boldsymbol{y}^k)^\top] = (\boldsymbol{I} - \boldsymbol{D})^{-1} \boldsymbol{B} \boldsymbol{B}^\top (\boldsymbol{I} - \boldsymbol{D})^{-\top} \,. \tag{3.15}$$

Note that $\boldsymbol{C}_y$ is rank deficient, and its pseudo inverse (denoted by $(\cdot)^\dagger$) is given as:

$$\boldsymbol{C}_y^\dagger = (\boldsymbol{I} - \boldsymbol{D})(\boldsymbol{B}\boldsymbol{B}^\top)^\dagger(\boldsymbol{I} - \boldsymbol{D})^\top \ . \tag{3.16}$$

Effectively, solving the Graph LASSO problem (3.14) by setting $\hat{\boldsymbol{C}}_x = (1/K)\sum_{k=1}^K \boldsymbol{y}^k(\boldsymbol{y}^k)^\top$ as the empirical covariance matrix of the regular agents' opinions finds the sparsest positive semidefinite matrix that approximates (3.16). As $\boldsymbol{I} - \boldsymbol{D}$ is sparse, it is anticipated that the gLASSO method may be able to recover the support of $\boldsymbol{D}$ partially. However, there is no theoretical guarantee to its identifiability condition, even when the covariance $\boldsymbol{C}_y$ is estimated perfectly.

Proposed Method. Departing from the common graphical model based formulation, we propose to utilize a model/dynamics-based learning approach for identifying the latent network structure. To handle the rank deficiency issue, we exploit the sparsity of the network and demonstrate that the network can be *provably* identified under such weak assumptions on the observed data.

To begin with, consider splitting the linear equation in a row-by-row fashion, we observe that (3.8) implies:

$$f_i^k y_i^k - \Big(\boldsymbol{d}_i^\top \odot \big[\boldsymbol{\mu}(\boldsymbol{y}^k)\big]_{i,:}\Big)\boldsymbol{y}^k = \Big(\boldsymbol{b}_i^\top \odot \big[\tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)\big]_{i,:}\Big)\boldsymbol{z}^k, \ \forall \ i \in [n] \ , \tag{3.17}$$

where $\boldsymbol{b}_i, \boldsymbol{d}_i$ denote the ith *row* of the matrix $\boldsymbol{B}, \boldsymbol{D}$, respectively. Importantly, the above is a linear equation in the network parameters $(\boldsymbol{b}_i, \boldsymbol{d}_i)$. This motivates to identify the network via solving the below problem:

$$\min_{\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i \geq \boldsymbol{0}} \ \sum_{k=1}^K J_k(\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i) + \rho \cdot g(\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i) \ \text{ s.t. } \ \hat{\boldsymbol{b}}_i^\top \boldsymbol{1} + \hat{\boldsymbol{d}}_i^\top \boldsymbol{1} = 1 \ , \tag{3.18}$$

where

$$J_k(\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i) := \Big| f_i^k y_i^k - \Big(\hat{\boldsymbol{d}}_i^\top \odot \big[\boldsymbol{\mu}(\boldsymbol{y}^k)\big]_{i,:}\Big)\boldsymbol{y}^k - \Big(\hat{\boldsymbol{b}}_i^\top \odot \big[\tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)\big]_{i,:}\Big)\boldsymbol{z}^k \Big|^2 \ , \tag{3.19}$$

$\rho > 0$ is a regularization parameter and $g(\cdot,\cdot)$ is the regularization function. Furthermore, if the dynamics is linear (with $\mu(x) = 1$), we shall include the constraint $[\hat{\boldsymbol{d}}_i]_i = 0$ in (3.18) to avoid the issue with diagonal ambiguity [cf. Lemma 3.2].

In this work, we examine two types of regularizations which focus on enforcing sparsity of the identified network, they are chosen depending on the type of opinion dynamics and available prior knowledge on the network:

$$\begin{aligned} & g_{\mathsf{stub}}(\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i) := \|\hat{\boldsymbol{b}}_i\|_1, \quad g_{\mathsf{active}}(\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i) := \|\hat{\boldsymbol{d}}_i\|_1 + \mathcal{I}_{\Omega_B}(\hat{\boldsymbol{b}}_i) \ , \\ & \text{such that } \ \mathcal{I}_{\Omega_B}(\hat{\boldsymbol{b}}_i) = \begin{cases} 0, & \text{if } [\hat{\boldsymbol{b}}_i]_j = 0, \ \forall \ (i,j) \notin \Omega_B \ , \\ \infty, & \text{otherwise} \ , \end{cases} \end{aligned} \tag{3.20}$$

where Ω_B is defined as the support set of the true stubborn-regular network, *i.e.,* $\Omega_B := \{(i,j) : B_{ij} \neq 0\}$. For the regularization functions above, $g_{\mathsf{stub}}(\cdot)$ regularizes on the sparsity of the stubborn-regular network; while $g_{\mathsf{active}}(\cdot)$ assumes knowledge on the network topology between stubborn and regular agents (only). We refer to the setting when Ω_B is available as the *active sensing* scenario. In the *active sensing* scenario, the stubborn agents are regarded as *active* agents inserted intentionally in order to reveal the network structure for the network identifier, as such, the set of agents to be influenced by them can be controlled artificially and thus Ω_B is known to us. Typically, we can achieve better performance in the active sensing setting and in fact, certain provable identifiability guarantees can be provided in this setting, as we shall expatiate in the next section. Notice that the support set of $\boldsymbol{D}$, *i.e.,* the regular-regular agents' network topology, is assumed to be completely *unknown.*

Lastly, we comment on the computational complexity of solving (3.18) using the proposed regularizers. Notice that (3.18) is a convex problem with $n + S$ variables. The above can be solved in polynomial time using off-the-shelf package like `cvx` or with specific softwares like `GPSR` [Figueiredo *et al.*(2007)]. To recover the entire network involving $(\boldsymbol{b}_i, \boldsymbol{d}_i)_{i=1}^n$, we can solve the n instances of (3.18) in parallel.

3.4 Guarantees for DeGroot Dynamics Identification

Our next endeavor is to study the *network identifiability* condition such that the social network can be *identified* perfectly. Notice that we seek to provide conditions such that both the network topology and the weights on the edges are recovered.

To conduct a tractable analysis, we focus on the setting with *linear* DeGroot dynamics, *i.e.,* when $\mu(x) = 1$ for all x, and using *active sensing*, *i.e.,* where the support of $\boldsymbol{B}$, Ω_B, is known. Our goal to derive the smallest possible number of stubborn agents and the corresponding configuration that guarantees perfect identifiability. The provided condition in turn represents also the lowest possible opinion data rank required. We assume:

H3.4 *Each row of the matrix $\boldsymbol{D}$, $\boldsymbol{d}_i$, satisfies $\|\boldsymbol{d}_i\|_0 \leq d_{\sf max}$ for all $i \in [n]$.*

H3.5 *The observation model* (3.11) *is exact such that opinions are observed without noise and we observe opinions from $K \geq S$ topics.*

H3.6 *The support of the matrix $\boldsymbol{B}$, $\Omega_B := \{(i,j) : [\boldsymbol{B}]_{ij} = 0\}$, is known.*

In light of H3.5 and H3.6, we shall study the the following network identification problem: for all $i \in [n]$,

$$\begin{aligned} \min_{\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i} \;\; \|\hat{\boldsymbol{d}}_i\|_0 \;\; \text{s.t.} \;\; & \hat{\boldsymbol{b}}_i \geq \boldsymbol{0},\; \hat{\boldsymbol{d}}_i \geq \boldsymbol{0},\; \hat{\boldsymbol{b}}_i^\top \boldsymbol{1} + \hat{\boldsymbol{d}}_i^\top \boldsymbol{1} = 1,\; [\hat{\boldsymbol{d}}_i]_i = 0, \\ & \hat{\boldsymbol{b}}_i^\top \boldsymbol{z}^k + \hat{\boldsymbol{d}}_i^\top \boldsymbol{y}^k = y_i^k,\; \forall\, k,\; [\hat{\boldsymbol{b}}_i]_j = 0,\; \forall\, j \in \Omega_{b_i}\;, \end{aligned} \tag{3.21}$$

the above problem is similar to problem (3.18) with the regularizer $g^1(\cdot)$. Analyzing the set of feasible solution to (3.21) leads us to study the linear system:

$$\hat{\boldsymbol{b}}_i^\top \boldsymbol{z}^k + \hat{\boldsymbol{d}}_i^\top \boldsymbol{y}^k = y_i^k,\; \forall\, k \quad \Longrightarrow \boldsymbol{Z}^\top \hat{\boldsymbol{b}}_i + \boldsymbol{Y}^\top \hat{\boldsymbol{d}}_i = \boldsymbol{y}_i\;, \tag{3.22}$$

where $\boldsymbol{Z} := (\boldsymbol{z}^1, \ldots, \boldsymbol{z}^K) \in \mathbb{R}^{S \times K}$, $\boldsymbol{Y} := (\boldsymbol{y}^1, \ldots, \boldsymbol{y}^K) \in \mathbb{R}^{n \times K}$ and $\boldsymbol{y}_i = (y_i^1, \ldots, y_i^K)$ is the

ith row of $\boldsymbol{Y}$. From (3.11), it holds that

$$\boldsymbol{Y}\boldsymbol{Z}^{\dagger} = (\boldsymbol{I}-\boldsymbol{D})^{-1}\boldsymbol{B} \; , \tag{3.23}$$

where $\boldsymbol{Z}^{\dagger}$ denotes the pseudo inverse of $\boldsymbol{Z}$. Traditionally, analyzing the identifiability of (3.22) requires characterizing the *spark* of the resulting 'sensing matrix' (see [Eldar(2014)]). However, determining the spark of a matrix is non-trivial.

In fact, the system (3.22) is closely related to the problem of *compressed sensing*, as we consider the following alternative representation:

$$\begin{aligned} \boldsymbol{Z}^\top \hat{\boldsymbol{b}}_i + \boldsymbol{Y}^\top \hat{\boldsymbol{d}}_i = \boldsymbol{y}_i &\Longleftrightarrow \boldsymbol{Z}^\top \hat{\boldsymbol{b}}_i + \boldsymbol{Y}^\top (\hat{\boldsymbol{d}}_i - \boldsymbol{e}_i) = \boldsymbol{0} \\ &\Longleftrightarrow \hat{\boldsymbol{b}}_i + (\boldsymbol{Y}\boldsymbol{Z}^\dagger)^\top (\hat{\boldsymbol{d}}_i - \boldsymbol{e}_i) = \boldsymbol{0} \\ &\Longleftrightarrow \hat{\boldsymbol{b}}_i + \boldsymbol{B}^\top (\boldsymbol{I}-\boldsymbol{D})^{-\top} (\hat{\boldsymbol{d}}_i - \boldsymbol{e}_i) = \boldsymbol{0} \\ &\Longleftrightarrow \boldsymbol{B}^\top \big((\boldsymbol{I}-\boldsymbol{D})^{-\top} (\hat{\boldsymbol{d}}_i - \boldsymbol{e}_i) + \boldsymbol{e}_i \big) = \boldsymbol{b}_i - \hat{\boldsymbol{b}}_i \\ &\Longleftrightarrow \boldsymbol{B}^\top (\boldsymbol{I}-\boldsymbol{D})^{-\top} (\hat{\boldsymbol{d}}_i - \boldsymbol{d}_i) = \boldsymbol{b}_i - \hat{\boldsymbol{b}}_i \; . \end{aligned} \tag{3.24}$$

On the left hand side of (3.24), we note that due to the self-trust constraint $[\hat{\boldsymbol{d}}_i]_i = 0$, the number of unknowns in $\boldsymbol{d}_i$ is $n-1$. On the right hand side of (3.24), the difference $\boldsymbol{b}_i - \hat{\boldsymbol{b}}_i$ is zero on indices j whenever $[\boldsymbol{b}_i]_j = 0$, as the support of $\boldsymbol{b}_i$ is known to (3.21); otherwise, the terms on the right hand side are in general unknown.

In light of the above, a sufficient condition for network identification can be obtained by ignoring the rows in the linear system whenever $[\boldsymbol{b}_i]_j \neq 0$. In particular, we require that the matrix obtained by deleting such rows from $\boldsymbol{B}^\top(\boldsymbol{I}-\boldsymbol{D})^{-\top} \in \mathbb{R}^{S\times n}$ have a null space that consists only of *dense* vector. It follows that one could study the so-called restricted isometry property of such matrix. Before giving our identifiability condition, we provide two further remarks:

- Observe that $\boldsymbol{B}^\top(\boldsymbol{I}-\boldsymbol{D})^{-\top} = \boldsymbol{B}^\top(\boldsymbol{I}+\boldsymbol{D}+\boldsymbol{D}^2+\ldots)^\top$; *i.e.,* the sensing matrix (before row-deletion) is a perturbed version of $\boldsymbol{B}^\top$. When the perturbation induced by $\boldsymbol{D}$ is small, we could study $\boldsymbol{B}$ alone as the sensing matrix.

- There exists a trade-off between $|\Omega_{b_i}|$ and the identifiability performance. Notice that a sensing matrix's performance (e.g., as measured by the so called restricted isometry property constant) is typically better if the matrix is dense. However, as indicated in (3.24), we need to ensure that there is a sufficient number of known 'observations' (or zeros) in the right hand side of the underdetermined system (3.24), which is determined by $|\Omega_{b_i}|$.

The second remark prompts us to consider an optimized placement of stubborn agents when the matrix $\boldsymbol{B}^\top$ is required to be sparse while maintaining a good sensing performance. As suggested in [Khajehnejad *et al.*(2011)], a good choice is to construct $\boldsymbol{B}$ such that each row in $\boldsymbol{B}$ has a constant number ℓ of non-zero elements. Our proposed construction is summarized by the following assumption:

H3.7 *The support of $\boldsymbol{B} \in \mathbb{R}^{n \times S}$, i.e., $\Omega_{\boldsymbol{B}}$, is constructed such that each row of it has exactly ℓ non-zero elements, selected randomly and independently.*

Notice that the above corresponds to setting the stubborn-regular network as a random, constant *left*-degree bipartite graph.

The theorem below provides the main result of this chapter. It gives the condition on $d_{\sf max}$ and S such that the social network can be identified through (3.21). Let $H(x)$ be the binary entropy function, we have the following sufficient condition:

Theorem 3.1 *Define $\alpha := 2d_{\sf max}/n$, $b_{min} := \min_{ij \in \Omega_{\boldsymbol{B}}} B_{ij}$, $b_{max} := \max_{ij \in \Omega_{\boldsymbol{B}}} B_{ij}$, $\beta := S/n$ and $\beta' := \beta - \ell/n$. Assume that conditions H3.4 and H3.7 hold, and that the following conditions are also satisfied*

$$\ell > \max\left\{4, 1 + \frac{H(\alpha) + \beta' H(\alpha/\beta')}{\alpha \log(\beta'/\alpha)}\right\},\ b_{min}(2d-3) - 1 - 2b_{max} > 0\ . \tag{3.25}$$

Then, as $n \to \infty$, there is a unique optimal solution to (3.21) *that* $(\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i) = (\boldsymbol{b}_i, \boldsymbol{d}_i)$ *with*

	$\alpha = 0.08$	$\alpha = 0.16$	$\alpha = 0.24$	$\alpha = 0.32$	$\alpha = 0.40$
$\ell = 5$	0.3420	0.5280	0.6730	0.7940	0.8950
$\ell = 6$	0.2340	0.3850	0.5100	0.6190	0.7160
$\ell = 7$	0.1870	0.3190	0.4330	0.5360	0.6290

Table 3.1: Evaluating the minimum β' required by (3.25) for the sufficient condition of perfect network identification with different combinations of ℓ, α. Note that $\beta' > \alpha$ and the number of stubborn agents required can be evaluated as $S \approx \beta' n + \ell$ and the maximum in-degree required is $\alpha n/2$.

probability one. Moreover, the failure probability is bounded as:

$$\Pr\Big((\hat{\boldsymbol{b}}_i, \hat{\boldsymbol{d}}_i) \neq (\boldsymbol{b}_i, \boldsymbol{d}_i), \ \forall \ i \in [n] \Big) \leq \left(\frac{\ell}{\beta} \right)^4 \frac{\ell - 1}{n^2} + \mathcal{O}(n^{2-(\ell-1)(\ell-3)}) \ . \tag{3.26}$$

The proof of Theorem 3.1 is in Appendix 3.D where the claim is proven by treating the unknown entries of $\boldsymbol{B}$ as *erasure bits*, and showing that the sensing matrix with erasure corresponds to a high quality expander graph in high probability. To the best of our knowledge, Theorem 3.1 is a new recoverability result proven for blind compressed sensing problems.

The first condition in (3.25) provides a guideline for determining the number of stubborn agents S needed and the role played by the sparsity parameter ℓ for $\boldsymbol{B}$. To gain some intuition, consider the situation where $n \to \infty$, $\beta', \alpha \to 0$ while the ratio β'/α is constant; then, the second condition in (3.25) can be approximated by

$$\ell > \max \left\{ 4, 1 + \frac{\beta'}{\alpha} \frac{H(\alpha/\beta')}{\log(\beta'/\alpha)} \right\} , \tag{3.27}$$

where the right hand side is minimized by $\beta'/\alpha \approx 1.27$ and requiring $\ell > 4.362$. Hence, setting $\ell = 5$ so this condition holds, the number of the stubborn agents needed is:

$$S \geq \beta n = 5 + \beta' n \geq 5 + 1.27 \alpha n \geq 5 + 2.54 d_{\sf max} = \Omega(d_{\sf max}) \ . \tag{3.28}$$

On the other hand, the second condition in (3.25) indicates that the amount of *relative trust* on the stubborn agents in the network cannot be too small. This is reasonable since the network identification performance should depend on the degree of influence of the stubborn agents relative to everyone else. Table 3.1 gives a list of the values required for β' and subsequently the required number of stubborn agents can be derived. Note that the number of stubborn agents required is still large. However, as this number only corresponds to a sufficient condition for perfect network identification, in practice the model based method also provides good performance when this condition is significantly relaxed.

We notice that Theorem 3.1 is proven for the case when an ℓ_0 norm minimization problem (3.21) is considered. Even though (3.21) is non-convex, it can be well approximated by its ℓ_1 approximation. In particular, problem (3.18) solved with $g_{\sf active}(\cdot)$ gives a good approximation that it, as we observe from the numerical experiments. We also remark that the probability bound in (3.26) is associated to the random construction of Ω_B in H3.7. In particular, when n is finite, this failure probability grows with the size of $\Omega_{\boldsymbol{B}}$, *i.e.,* $\mathcal{O}(\ell^5)$. This indicates a possible tradeoff between the size of $\Omega_{\boldsymbol{B}}$ and the identification accuracy. We conclude this section by showing how to deal with randomized interactions. Lastly, we remark that the results above are proven only for DeGroot dynamics. An interesting but challenging extension is to generalize the identifiability conditions for nonlinear dynamics.

3.4.1 Random Opinion Dynamics

So far, our network identification method requires collecting the steady state opinions $(\boldsymbol{z}^k, \boldsymbol{y}^k)$ resulted from a static opinion dynamics model. A more realistic setting is to consider a randomized opinion dynamics. Here we focus on the DeGroot opinion dynamics as in the previous section. Importantly, we recall H3.1 and the following randomized linear opinion dynamics:

$$\boldsymbol{x}(t+1;k) = \boldsymbol{A}(t;k)\boldsymbol{x}(t;k), \quad \text{where} \quad \boldsymbol{A}(t;k) := \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B}(t;k) & \boldsymbol{D}(t;k) \end{pmatrix}. \qquad (3.29)$$

In the same spirit, we also define $\mathbb{E}[\boldsymbol{B}(t;k)] = \boldsymbol{B}$ and $\mathbb{E}[\boldsymbol{D}(t;k)] = \boldsymbol{D}$. Now, let us re-examine the requirements on the opinion data. From (3.11), as one wishes that the collected data $(\boldsymbol{y}^k, \boldsymbol{z}^k)$ from the kth discussion to satisfy $\boldsymbol{y}^k = (\boldsymbol{I} - \boldsymbol{D})^{-1}\boldsymbol{B}\boldsymbol{z}^k$. Naturally, this can be obtained by taking the following expectations:

$$\boldsymbol{y}^k = \mathbb{E}[\boldsymbol{y}(\infty;k)|\boldsymbol{z}^k] \;\text{ and }\; \boldsymbol{z}^k = \mathbb{E}[\boldsymbol{z}(t;k)] \; . \tag{3.30}$$

However, in practice, this may be difficult to realize as computing the expectation requires taking an average over the ensemble of the sample paths of $\{\boldsymbol{A}(t;k)\}_{\forall t, \forall k}$.

Instead of proceeding with (3.30), we prove that the randomized opinion dynamics is an ergodic process and replace (3.30) with a *time average*. To fix idea, let us consider a noisy observation model on the opinions:

$$\hat{\boldsymbol{x}}(t;k) = \boldsymbol{x}(t;k) + \boldsymbol{n}(t;k) \; . \tag{3.31}$$

Now, suppose that we accrue a time series of the opinions $\{\hat{\boldsymbol{x}}(t;k)\}_{t \in \mathcal{T}_k}$, where $\mathcal{T}_k \subseteq \mathbb{N}$ is an arbitrary sampling set. We define:

$$\hat{\boldsymbol{x}}(\mathcal{T}_k;k) \triangleq \frac{1}{|\mathcal{T}_k|} \sum_{t \in \mathcal{T}_k} \hat{\boldsymbol{x}}(t;k) \approx \mathbb{E}[\boldsymbol{x}(\infty;k)|\boldsymbol{x}(0;k)] \; . \tag{3.32}$$

Specifically, the temporal opinions samples are collected through random (and possibly noisy) sampling at time instances on the opinions. The following theorem characterizes the performance of (3.32):

Theorem 3.2 *Consider the estimator in* (3.32) *and we denote the steady state opinion as* $\overline{\boldsymbol{x}}(\infty;k) \triangleq \lim_{t\to\infty} \mathbb{E}\{\boldsymbol{x}(t;k)|\boldsymbol{x}(0;k)\} = \boldsymbol{A}^{\infty}\boldsymbol{x}(0;k)$. *Assume that* $\mathbb{E}\{\|\boldsymbol{D}(t;k)\|_2\} < 1$. *Let* $T_o := \min\{t : t \in \mathcal{T}_k\}$, *if* $T_o \to \infty$, *1. then the estimator* (3.32) *is unbiased:*

$$\mathbb{E}[\hat{\boldsymbol{x}}(\mathcal{T}_k;k)|\boldsymbol{z}(0;k)] = \overline{\boldsymbol{x}}(\infty;k) \ . \tag{3.33}$$

2. then the estimator (3.32) *is asymptotically consistent:*

$$\lim_{|\mathcal{T}_k|\to\infty} \mathbb{E}[\|\hat{\boldsymbol{x}}(\mathcal{T}_k;k) - \overline{\boldsymbol{x}}(\infty;k)\|_2^2|\boldsymbol{x}(0;k)] = 0 \ . \tag{3.34}$$

For the latter case, we have

$$\mathbb{E}\Big[\|\hat{\boldsymbol{x}}(\mathcal{T}_k;k) - \overline{\boldsymbol{x}}(\infty;k)\|_2^2 \mid \boldsymbol{x}(0;k)\Big] \leq \frac{C'}{|\mathcal{T}_k|}\Big(\sum_{i=0}^{|\mathcal{T}_k|-1} \lambda^{\min_\ell |t_{\ell+i}-t_\ell|}\Big) \ , \tag{3.35}$$

where $C' < \infty$ is a constant and $\lambda = \lambda_{max}(\boldsymbol{D}) < 1$, i.e., *the latter term is a geometric series with bounded sum.*

Note that a similar result to Theorem 3.2 was reported in [Ravazzi *et al.*(2015)]. Our result is specific to the case with stubborn agents, which allows us to find a precise characterization of the mean square error. The proof of Theorem 3.2 can be found in Appendix 3.E.

Remark 3.1 *From* (3.35)*, we observe that the upper bound on the mean square error can be optimized by maximizing $\min_{i,j,i\neq j}|t_i - t_j|$. Suppose that the samples $\hat{\boldsymbol{x}}(\mathcal{T}_k;k)$ have to be taken from a finite interval $[T_{max}] \setminus [T_o]$, $T_{max} < \infty$ and $|\mathcal{T}_k| < \infty$; here, the best estimate can be obtained by using sampling instances that are drawn uniformly from $[T_{max}] \setminus [T_o]$.*

3.5 Community Detection from Low-Rank Data

The *community detection* problem is a closely related to network identification described in the above. Here, the aim is to partition the set of agents into *communities* which form close knit with the other [Fortunato(2010)]. Knowing the community structure is crucial to understanding a social network from a *macroscopic* view, e.g., it reveals the *membership* property of specific agent such as the political party that an agent leans towards. To this

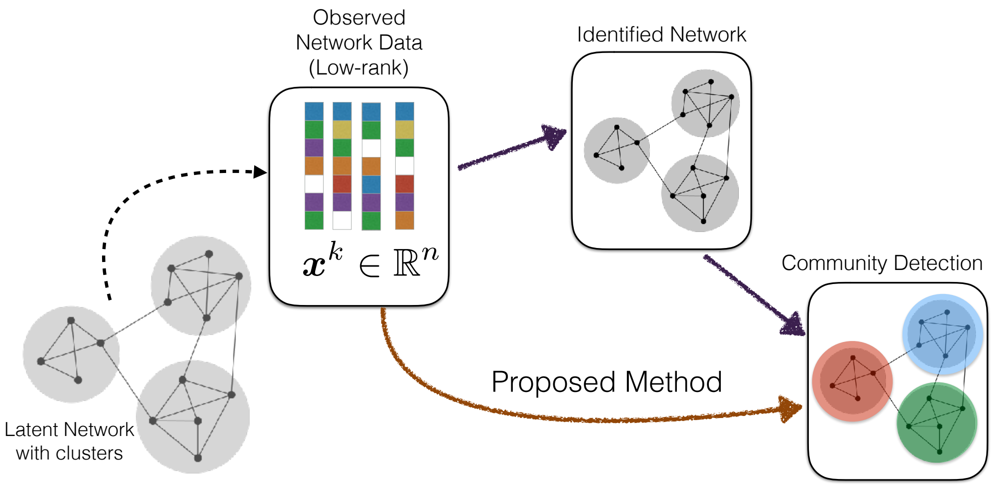

Figure 3.3: Proposed approach for the *sketched* community detection method from low-rank observed opinion data.

end, computational methods, which are often put into the same context as data clustering, have been studied in the prior work [Boutsidis *et al.*(2015),Tremblay *et al.*(2016)] with focus on handling large graphs efficiently. However, almost all of the prior work require knowing the network topology structure as the input parameter. When the network topology is unknown, a naive approach is to adopt a *two-stage* procedure — we first apply the network identification method described in Section 3.3 to infer the entire network, then we apply the off-the-shelf methods in [Fortunato(2010)] to perform community detection. Clearly, the *two-stage* approach is not desirable as it involves a relatively high computation and storage complexity in identifying the entire network, moreover in the rank deficient case (with limited number of stubborn agents) the network identification result may be erroneous, leading to further errors in the community detection.

In this section, we present a new method for community detection on the low-rank *steady-state* data collected under the DeGroot opinion dynamics model with stubborn agents. Specifically, we propose a *direct* approach which finds the community structure without the need of identifying the entire network, as overviewed in Figure 3.3.

Community Detection. It is instructive to describe the relationship between the com-

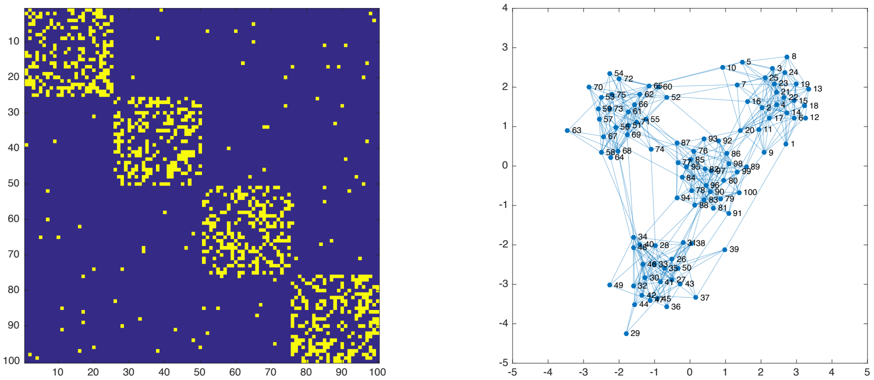

Figure 3.4: Illustrating a network with $C = 4$ communities and its adjacency matrix. (Left) Adjacency matrix with the yellow dots indicating the existence of an edge. (Right) Visualization of the corresponding graph using a 'force' layout.

munity structure and the rank of adjacency matrix. In particular, a set of nodes is grouped into a *community* when the edge density among these nodes is high compared to those nodes not in the same community. An instance of such network with $C = 4$ communities is illustrated in Figure 3.4. As seen, the adjacency matrix is roughly block diagonal with $C = 4$ blocks, and each block is approximately equal to an all-ones matrix, *i.e.,* a rank-one matrix, implying that the rank of the adjacency matrix is roughly $C = 4$.

The above observation suggests that for a network with C communities ($C \ll n$), its adjacency matrix should also have an approximate rank of C, *i.e.,* a low rank matrix. In fact, the classical *spectral clustering* method [Ng *et al.*(2002)] can be motivated from this observation — we first find the top C left singular vectors, $\boldsymbol{V}_C \in \mathbb{R}^{n \times C}$, corresponding to the largest C singular values, of the (weighted) adjacency matrix $\boldsymbol{D}$, then we apply the K-means algorithm [Hartigan and Wong(1979)] to cluster the n row vectors in $\boldsymbol{V}_C$ into C clusters. In this method, the first step can be seen as a 'denoising' procedure as it retains the principal components necessary for clustering.

Proposed Method. Now we begin the development of our community detection method, let us recall the relationship between the pair of observed tuple $(\boldsymbol{y}^k, \boldsymbol{z}^k)$. Under the DeGroot opinion dynamics setting, we have $\boldsymbol{y}^k = (\boldsymbol{I} - \boldsymbol{D})^{-1} \boldsymbol{B} \boldsymbol{z}^k$ from (3.23). Through stacking the

K pairs of observed opinion data and taking pseudo inverse of the stubborn agents' opinions, the following linear transformation can be computed:

$$\boldsymbol{\Gamma} := \boldsymbol{Y}\boldsymbol{Z}^{\dagger} = (\boldsymbol{I} - \boldsymbol{D})^{-1}\boldsymbol{B} = \big(\boldsymbol{I} + \boldsymbol{D} + \boldsymbol{D}^2 + \cdots\big)\boldsymbol{B} \ . \tag{3.36}$$

We observe that $\boldsymbol{\Gamma}$ can be decomposed into:

$$\boldsymbol{\Gamma} = \boldsymbol{B} + \boldsymbol{D}(\boldsymbol{I} - \boldsymbol{D})^{-1}\boldsymbol{B} = \boldsymbol{B} + \big(\boldsymbol{D} + \boldsymbol{D}^2 + \cdots\big)\boldsymbol{B} \ , \tag{3.37}$$

Let us first focus on the last term. As $n \gg S$, the matrix above has a dimension of $n \times S$ and it can be viewed as a *sketch* of the matrix series $\boldsymbol{D} + \boldsymbol{D}^2 + \cdots$. Moreover, the matrix series $\tilde{\boldsymbol{D}} := \boldsymbol{D}(\boldsymbol{I} - \boldsymbol{D})^{-1} = \boldsymbol{D} + \boldsymbol{D}^2 + \cdots$ is a superposition of the regular sub-network (or regular-to-regular network) and its high order multiplications, note that:

$$[\boldsymbol{D}^k]_{ij} > 0 \;\text{ if there exists a path from } j \text{ to } i \text{ in } G[V_r] \text{ with length } \leq k \ .$$

On the other hand, as $\|\boldsymbol{D}\|_2 < 1$ under H3.3, the edge weights of the higher-order links in $\boldsymbol{D}^k$ decays exponentially. Therefore, the *community structure* in the original regular-regular network is retained in the series $\tilde{\boldsymbol{D}}$ since the latter is dominated by $\boldsymbol{D}$. It follows that if there are C communities in the original regular-regular network $\boldsymbol{D}$, the matrix series $\tilde{\boldsymbol{D}}$ will be approximately rank C. We remark that the matrix series $\tilde{\boldsymbol{D}}$ is also related to the notion of Bonacich centrality [Bonacich(1987)].

Consider the second term in the decomposition (3.37), as $n \gg S$, the matrix product $\tilde{\boldsymbol{D}}\boldsymbol{B}$ can be seen as a *sketched* version of $\tilde{\boldsymbol{D}}$. However, if $S \geq C$ and $\tilde{\boldsymbol{D}}$ contains C communities, under some mild conditions, it can be shown that the range space of $[\tilde{\boldsymbol{D}}\boldsymbol{B}]_C$, the rank C approximation of $\tilde{\boldsymbol{D}}\boldsymbol{B}$, is the same as the range space of $\tilde{\boldsymbol{D}}$. This implies the spectral clustering result on $\tilde{\boldsymbol{D}}$ should match with that on $\tilde{\boldsymbol{D}}\boldsymbol{B}$. The observation above can be quantified rigorously as stated in the following proposition:

Proposition 3.1 *Define the following SVDs of the matrices:*

$$\tilde{\boldsymbol{D}} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{U}^\top, \quad \tilde{\boldsymbol{D}}\boldsymbol{B} = \boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{Q}^\top \; , \tag{3.38}$$

and we denote the partitions $\boldsymbol{V} = [\boldsymbol{V}_C \; \boldsymbol{V}_{N-C}]$*,* $\boldsymbol{U} = [\boldsymbol{U}_C \; \boldsymbol{U}_{N-C}]$*,* $\boldsymbol{P} = [\boldsymbol{P}_C \; \boldsymbol{P}_{N-C}]$ *and* $\boldsymbol{Q} = [\boldsymbol{Q}_C \; \boldsymbol{Q}_{N-C}]$*, such that* $\boldsymbol{V}_C$ *(*resp. $\boldsymbol{V}_{N-C}$*) denotes the 'left-most' (*resp. *'right-most') C (*resp. $N-C$*) column vectors of* $\boldsymbol{V}$*. Let* $[\tilde{\boldsymbol{D}}]_C$ *be the rank-C approximation of* $\tilde{\boldsymbol{D}}$*. If the matrices* $[\tilde{\boldsymbol{D}}]_C\boldsymbol{B}$ *and* $\tilde{\boldsymbol{D}}\boldsymbol{B}$ *are at least rank-C, then*

$$\|\boldsymbol{P}_C\boldsymbol{P}_C^\top - \boldsymbol{V}_C\boldsymbol{V}_C^\top\|_2^2 = \frac{\gamma^2}{1+\gamma^2}, \quad \gamma \leq \frac{\lambda_{C+1}}{\lambda_C}\|\boldsymbol{U}_{N-C}^\top\boldsymbol{B}\boldsymbol{Q}_C\|_2\|(\boldsymbol{U}_C^\top\boldsymbol{B}\boldsymbol{Q}_C)^{-1}\|_2 \; . \tag{3.39}$$

The proof is inspired by [Boutsidis *et al.*(2015)] and can be found in Appendix 3.F.

The squared spectral norm error in (3.39) quantifies the difference between the range space spanned by the top C left singular vectors of $\tilde{\boldsymbol{D}}$ and its sketched version $\tilde{\boldsymbol{D}}\boldsymbol{B}$. Essentially, the proposition shows that this error depends on the *spectral gap* of the 'adjacency' matrix $\tilde{\boldsymbol{D}}$. Notice that the constant $\eta := \lambda_{C+1}/\lambda_C \ll 1$ if $\tilde{\boldsymbol{D}}$ is approximately rank C. This confirms with the intuition that if $\tilde{\boldsymbol{D}}$ contains C communities, then the two range spaces match and the spectral clustering applied on $\tilde{\boldsymbol{D}}\boldsymbol{B}$ can recover these communities.

Now suppose that *(i)* $S > C$, *i.e.,* there are more stubborn agents than the number of communities in $\boldsymbol{D}$, and *(ii)* the stubborn-regular network $\boldsymbol{B}$ is sparse, then the linear transformation $\boldsymbol{\Gamma}$ follows a 'sparse + low-rank' decomposition [cf. (3.37)]. We are interested in estimating the 'low-rank' component in it, this motivates us to consider the following problem:

$$\boldsymbol{\mathcal{Y}}^\star = \arg \min_{\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{n \times S}} \; \|\text{vec}(\boldsymbol{\Gamma} - \boldsymbol{\mathcal{Y}})\|_1 + \rho\|\boldsymbol{\mathcal{Y}}\|_{\sigma,1} \; , \tag{3.40}$$

where $\|\cdot\|_{\sigma,1}$ denotes the nuclear norm and $\rho > 0$ is a predefined parameter. In the above problem, the first term exploits the sparseness of $\boldsymbol{B}$ and seeks to remove it from the observation $\boldsymbol{\Gamma}$; while the second term exploits the low-rankness of $\boldsymbol{\Gamma} - \boldsymbol{B}$.

Algorithm 3.1 Community detection from low-rank excitation.

```
Input: Collected opinion tuples $\{\boldsymbol{y}^k, \boldsymbol{z}^k\}_{k=1}^K$; desired number of communities $C$.
Calculate $\boldsymbol{\Gamma} = \boldsymbol{Y}\boldsymbol{Z}^\dagger$ from the collected opinion tuples $\{\boldsymbol{y}^k, \boldsymbol{z}^k\}_{k=1}^K$.
Estimate $\tilde{\boldsymbol{D}}\boldsymbol{B}$ by solving the convex program (3.40) and denote $\boldsymbol{\mathcal{Y}}^\star$ as the optimal solution.
Find the top $C$ *left* singular vectors to $\boldsymbol{\mathcal{Y}}^\star$ associated with the largest $C$ singular values. Denote the set of singular vectors as $\hat{\boldsymbol{P}}_C \in \mathbb{R}^{N \times C}$.
Perform $K$-means clustering (e.g., [Hartigan and Wong(1979)]), which optimizes:
    $\min_{\mathcal{C}_1,\ldots,\mathcal{C}_C} \sum_{i=1}^{C} \sum_{j \in \mathcal{C}_i} \left\| \hat{\boldsymbol{p}}_j - \frac{1}{|\mathcal{C}_i|} \sum_{q \in \mathcal{C}_i} \hat{\boldsymbol{p}}_q \right\|_2^2 \ \text{s.t.} \ \mathcal{C}_i \subseteq V \ ,$    (3.41)
    where $\hat{\boldsymbol{p}}_i := [\hat{\boldsymbol{P}}_C]_{i,:} \in \mathbb{R}^C$. Let the solution be $\hat{\mathcal{C}}_1, \ldots, \hat{\mathcal{C}}_C$.
Output: Partition of $V_r$ into $C$ communities, $\hat{\mathcal{C}}_1, \ldots, \hat{\mathcal{C}}_C$.
```

As shown in [Chandrasekaran *et al.*(2011)], when $\boldsymbol{B}$ is sufficiently sparse and the ratio C/S is sufficiently small, then the optimal solution to (3.40) will be the desired sketch $\tilde{\boldsymbol{D}}$, *i.e.,* $\boldsymbol{\mathcal{Y}}^\star \approx \tilde{\boldsymbol{D}}\boldsymbol{B}$. As argued by Proposition 3.1, applying spectral clustering on $\tilde{\boldsymbol{D}}\boldsymbol{B}$ yields a similar result as applying spectral clustering on $\tilde{\boldsymbol{D}}$ (and thus $\boldsymbol{D}$) given $\eta \ll 1$. Finally, we summarize the proposed community detection method in the pseudo code Algorithm 3.1.

3.6 Numerical Results

This section validates the efficacies of our methods with numerical experiments. Specifically, we focus on cases when the network dynamics model (3.3) is exact while the measurements may be noisy. In the following, we first focus on the linear DeGroot opinion dynamics case with $\mu(x) = 1$ for all x, then we study the network recovery performance under nonlinear opinion dynamics. We mainly focus on synthetic opinion data where both the network topology and the opinions observed on it are generated randomly. To emphasize the crucial importance of considering data collected from real networks, e.g., the online social networks (e.g., Facebook, Twitter), we also consider the case when the network topology is taken from real world social networks.

For topology identification, we compare the area under precision-recall (AUPR) and area under receiver operating characteristic curves (AUROC), see [Davis and Goadrich(2006)] for the respective definitions; for the network weights identification, we compare the normalized mean square error:

$$\mathsf{NMSE} = \frac{\|[\hat{\boldsymbol{B}}^\star, \hat{\boldsymbol{D}}^\star] - [\boldsymbol{B}_0, \boldsymbol{D}_0]\|_F^2}{\|[\boldsymbol{B}_0, \boldsymbol{D}_0]\|_F^2} \ , \tag{3.42}$$

where $\boldsymbol{B}_0, \boldsymbol{D}_0$ are the normalized ground truth trust matrices as defined in (3.13), and $\hat{\boldsymbol{B}}^\star, \hat{\boldsymbol{D}}^\star$ are the optimal solution obtained from solving (3.18). Notice that the topology identification is perfect when AUROC and AUPR approach 1. If the NMSE is zero, then the network topology is perfectly identified, *i.e.,* it is a stronger condition to satisfy.

3.6.1 Linear Opinion Dynamics

The first series of numerical experiments focuses on the linear DeGroot opinion dynamics. In particular, our aim is to present numerical evidences to verify our theoretical findings on perfect network identification [cf. Theorem 3.1]. We set $K = 2S$ and $\rho = 0.1$ in Problem (3.18) for all experiments below. The initial opinions are all generated as $\mathcal{U}[-1, 1]$.

We first focus on the case with *perfect/noiseless* opinion observation, *i.e.,* the steady state opinions $\boldsymbol{y}^k, \boldsymbol{z}^k$ observed follow the relationship (3.11) exactly. Now let us consider a case with a relatively small-scale network with $n = 100$ regular agents and compare the performance with different number of stubborn agents, and the regular-regular network (corr. to $\boldsymbol{D}$) is generated as an Erdos-Renyi (ER) graph with connectivity probability of $\alpha = 0.08$. Moreover, following the analysis results in Theorem 3.1, we construct the stubborn-regular network (corr. to $\boldsymbol{B}$) by randomly assigning $\ell = 5$ stubborn agents to each regular agent. The network weights are first generated as $\mathcal{U}[0, 1]$ and then normalized such that $\boldsymbol{A}$ forms a stochastic matrix.

Under the setting described, Figure 3.5 shows the topology identification performance against the number of stubborn agents S. From the figure, we observe that the AUROC /

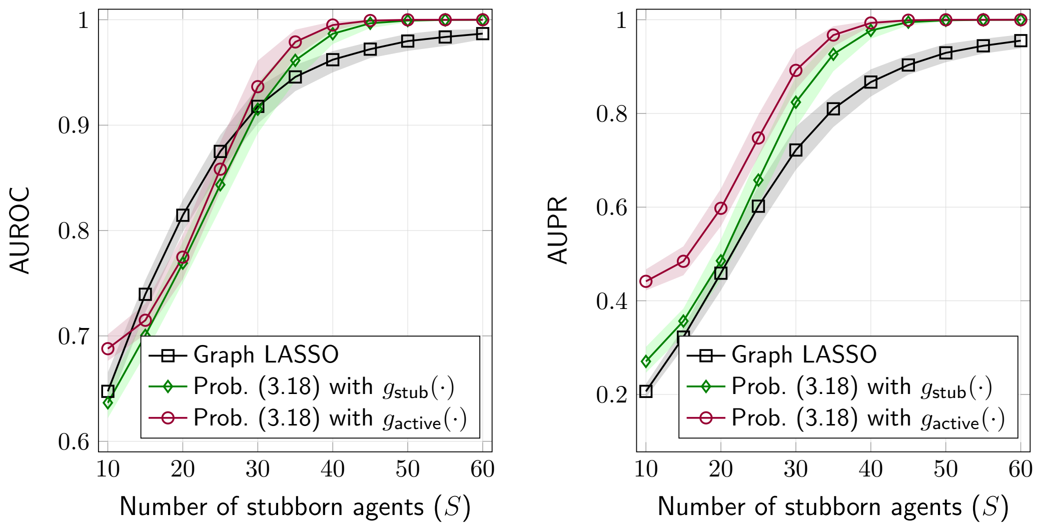

Figure 3.5: Comparing the topology recovery performance against the number of stubborn agents S. The regular-regular network is a 100-nodes ER graph with connectivity $\alpha = 0.08$. The shaded area shows the 5% / 95% percentile interval for the AUROC/AUPR performances. Problem (3.18) is tested with two regularizers — $g_{\mathsf{stub}}(\cdot)$ and $g_{\mathsf{active}}(\cdot)$.

AUPR performance improve as the number of stubborn agents increases. For the proposed method via solving (3.18), a nearly perfect topology identification is achieved when $S \approx 45 < n = 100$. Moreover, we see that the proposed method has outperformed Graph LASSO. Lastly, we observe that the proposed method with 'active sensing' (with $g_{\mathsf{active}}(\cdot)$ in problem (3.18)) has a slightly better topology identification performance than the case without 'active sensing' (with $g_{\mathsf{stub}}(\cdot)$ in problem (3.18)).

We also compare the network weights identification performance in Figure 3.6 using the same setting as before. Notice that Theorem 3.1 predicts that the number of stubborn agents required is roughly $S \approx 57$. This is confirmed by the left plot which shows that the NMSE performance under the setting with 'Reg. $\boldsymbol{B}$' and 'active sensing'. In particular, we observe that the actual performance is slightly better than the predicted one as the latter is only a sufficient condition. In the same figure, we also examined the effect on the performance with different construction of the stubborn-regular topology (corr. to $\boldsymbol{B}$). Specifically, the ER-like random construction yields a worse performance in terms of the

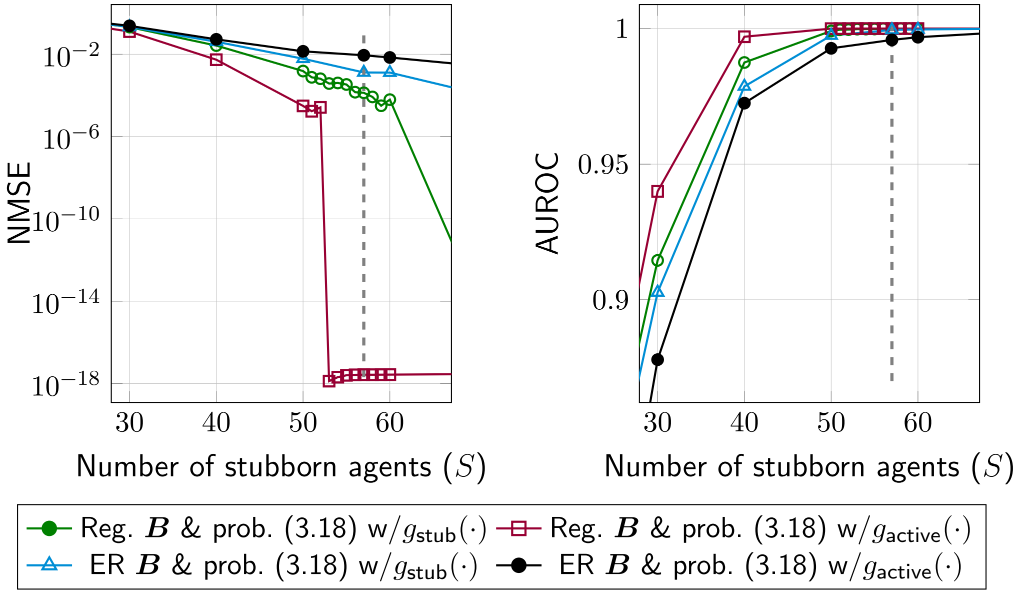

Figure 3.6: Comparing the network identification performance using different regularizers for (3.18) and constructions for the stubborn-normal network (corr. to $\boldsymbol{B}$). 'Reg. $\boldsymbol{B}$' refers to the setting when $\boldsymbol{B}$ is constructed according to H3.7 with $\ell = 5$; 'ER $\boldsymbol{B}$' corresponds to the construction with random edge selection with connectivity $\alpha = 0.08$. The dashed line indicates the number of stubborn agents required by Theorem 3.1.

NMSE performance, *i.e.,* it does not exhibit the phase-transition behavior as the case with the regular bipartite construction in H3.7. Nevertheless, we observe that the topology identification performance is less affected by this change, *i.e.,* the AUROC are close to 1 as we increase the number of stubborn agents in all cases.

We further verify the claim in Theorem 3.1 with the effect of ℓ in the construction of the stubborn-regular network $\boldsymbol{B}$. Specifically, we evaluate the NMSE and AUROC performance against the number of regular agents n in Figure 3.7, where the regular-regular network is constructed as ER graph with connectivity fixed at $\alpha = 0.08$. We focus only on the 'active sensing' setting with problem (3.18) using $g_{\mathsf{active}}(\cdot)$ as the regularizer. The number S of stubborn agents is set as the minimum number required by Theorem 3.1 [cf. see Table 3.1], which is a constant fraction of the number of regular agents n. Note that the theorem states that S can be reduced if we increase ℓ, yet the probability of failure may increase polynomially with ℓ (while still approaching zero as n increases). The results in the figure

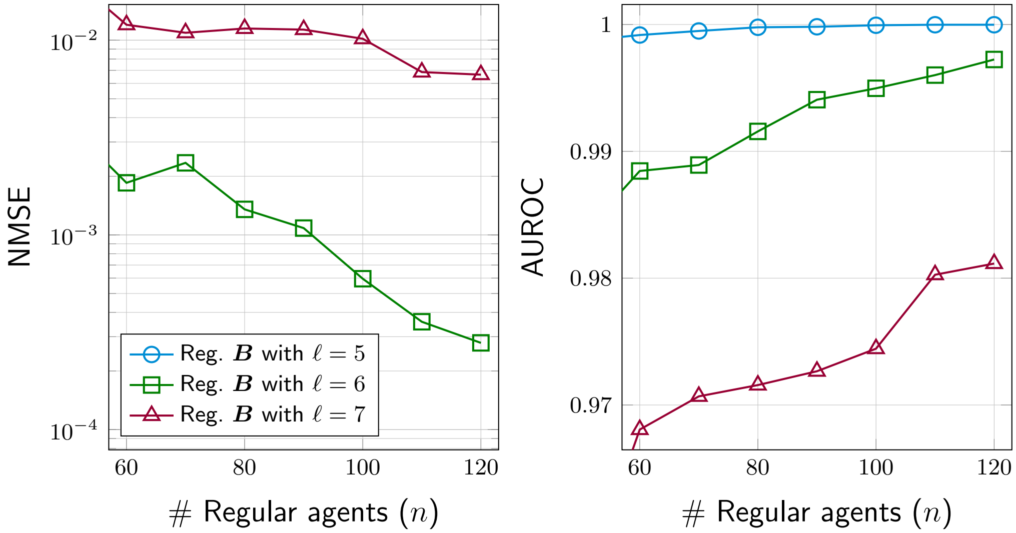

Figure 3.7: Comparing the network identification performance against the number of regular agents n, but using different ℓ for the constructions of the stubborn-regular network. The network connectivity is fixed at $\alpha = 0.08$ and the number of stubborn agents considered is predicted using Table 3.1.

corroborate with the statement above, as the identification performance (in terms of NMSE and AUROC) improve as n increases, for all values of ℓ. From the results above, we also observe that setting $\ell = 5, 6$ gives a better performance in practice for finite n.

The next experiment considers the network identification performance on a large-scale network. Here, we look at a similar setting as in Figure 3.6, yet the number of regular agents is $n = 1000$ the regular-regular network is an ER graph with the connectivity set to $\alpha = 1.01 \log n/n \approx 0.007$. We also remark that as the observed opinion data is low-rank, the graphical LASSO formulation is not numerically stable and is therefore skipped. As seen from the figure, the result from Theorem 3.1 continues to hold as it predicts the perfect identification when $S \geq 115$. The result also demonstrates that the proposed method can be scaled to handle large networks.

We now examine the performance of the proposed method applied to real network topology. Specifically, we consider the `facebook100` dataset [Traud *et al.*(2012)] and focus on the medium-sized network example `ReedCollege`. The randomized opinion exchange

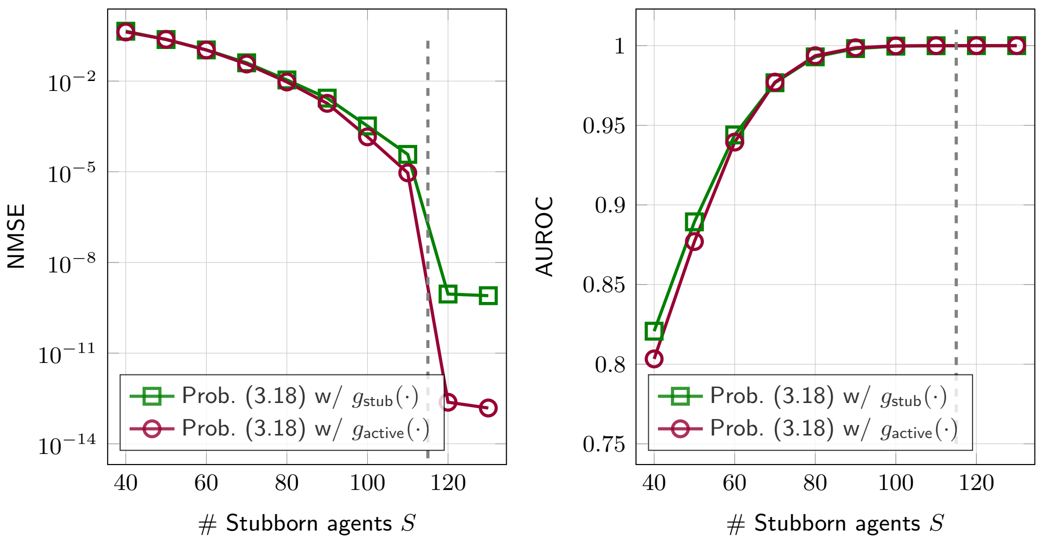

Figure 3.8: Comparing the network identification performance on a large-scale network with $n = 1000$. The regular-regular network is an ER graph with connectivity $\alpha \approx 0.007$; and the stubborn-regular network is constructed according to H3.7 with $\ell = 5$. The dashed line indicates the number of stubborn agents required by Theorem 3.1.

model is based on the randomized broadcast gossip protocol in [Aysal *et al.*(2009)] with uniformly assigned trust weights. Out of the available agents, we picked $S = 180$ agents with degrees closest to the median degree as the stubborn agents and removed the agents that are not adjacent to *any* of the stubborn agents. The selection of the stubborn agents is motivated by Theorem 1 as we require a moderate average degree for the resultant stubborn-to-regular agent network with better recovery guarantees. Our aim is to estimate the trust matrix $\boldsymbol{D}$, which corresponds to the subgraph with $n = 666$ regular agents, $|E| = 13,269$ edges and mean degree 19.92. Note that the bipartite graph from stubborn agents to regular agents has a mean degree of 25.07. The opinion dynamics data $\{\boldsymbol{y}^k, \boldsymbol{z}^k\}_{k=1}^K$ (with $K = 2S$) is collected using the estimator in Section 3.4.1, where we set $|\mathcal{T}_k| = 5 \times 10^5$ and the sampling instances are uniformly taken from the interval $[10^5, 5 \times 10^7]$. We apply the FISTA algorithm to approximately solve the network reconstruction problem (3.18). The NMSE of the reconstructed $\boldsymbol{D}$ is 0.1035 after 4×10^4 iterations. The program has terminated in about 30 minutes on a Xeon server running MATLAB$^{\text{TM}}$ 2014b.

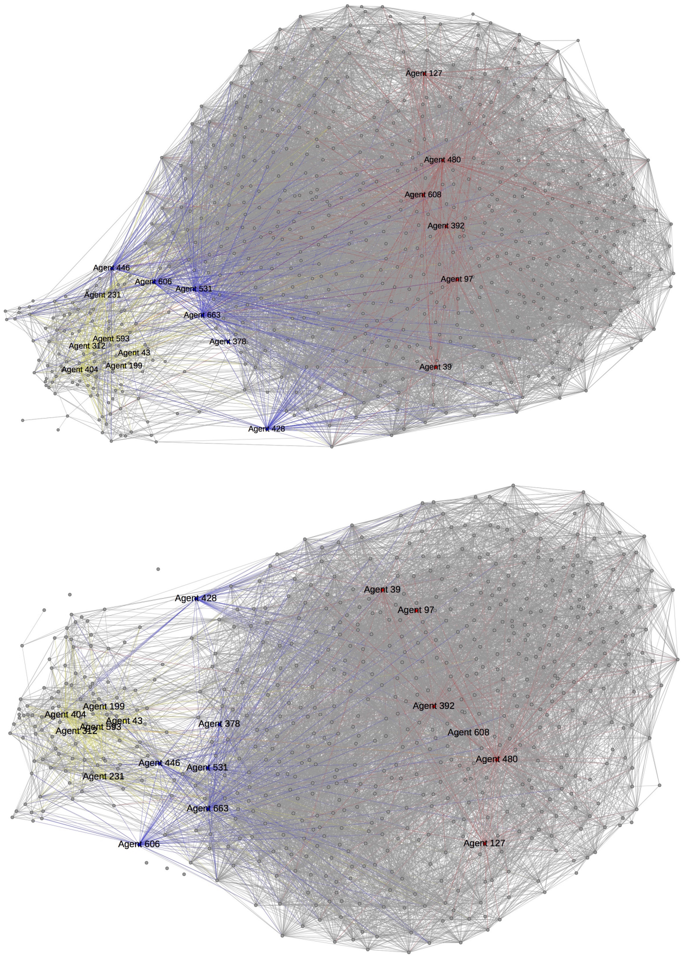

Figure 3.9: Comparing the social network of `ReedCollege` from `facebook100` dataset: (Left) the original network; (Right) the estimated network.

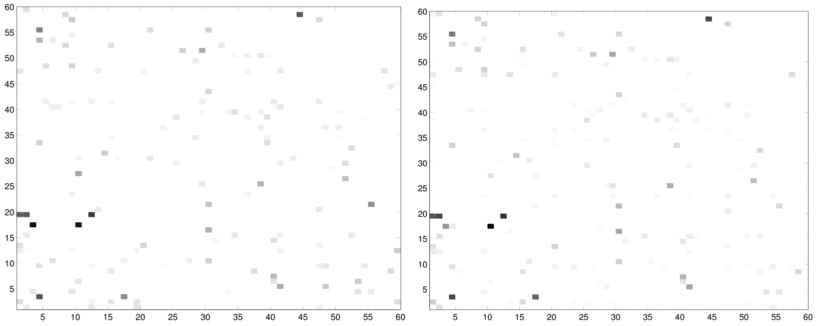

Figure 3.10: Comparing the reconstructed network for the `ReedCollege` social network in `facebook100` dataset — a closer look. (Left) Original network. (Right) Reconstructed network.

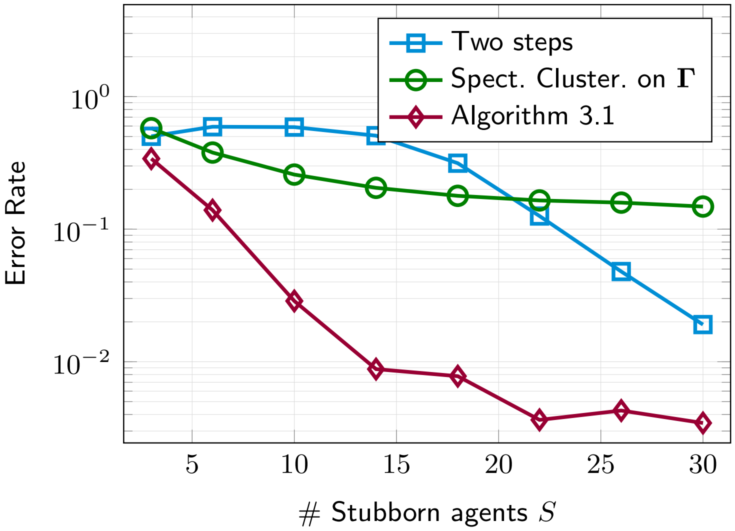

Figure 3.11: Community detection performance against the number of stubborn agents S. We consider a regular-to-regular network with $C = 3$ communities. The error rate is evaluated by comparing the community detection result to the ground truth used for generating the SBM networks.

We compared the estimated social network in both macroscopic and microscopic levels. Figure 3.9 shows the true/estimated network plotted in `gephi` [Bastian *et al.*(2009)] using the 'Force Atlas 2' layout with the edge weights taken into account. While it is impossible to compare every edges in the network, the figure gives a macroscopic view of the efficacy of the network reconstruction method. In particular, the estimated network follows a similar topology as the original one. For instance, there are clearly two clusters in both the estimated and original network. Moreover, the relative roles for individual agents are matched in both networks. For example, agents $\{39, ..., 608\}$ are found in the larger cluster, agents $\{378, ..., 663\}$ are found at the boundary between the clusters and agents $\{43, ..., 404\}$ are found in the smaller cluster, in both networks. Finally, in Figure 3.10 we compare the estimated principal sub-matrix of $\boldsymbol{D}$ taken from the first 60 rows/columns, *i.e.,* this corresponds to the social network between 60 agents. As seen, the original and estimated matrices are similar to each other, both in terms of the support set and the weights on individual edges between the agents.

Lastly, we present numerical results for the low-rank community detection method in Section 3.5. In the following, we generate the regular-regular network according to a stochastic block model (SBM) [Fortunato(2010)] with $n = 120$ agents, $C = 3$ communities, intra-community connectivity of $a = 16 \log n/n$ and inter-community connectivity of $b = \log n/n$. Moreover, the stubborn-regular network is generated as a random bipartite graph with connection probability of $6 \log n/n$. As a benchmark, we consider a 'two steps' approach, where we first infer the regular-regular network $\hat{\boldsymbol{D}}^{\star}$ through solving (3.18), then we detect the communities in $\hat{\boldsymbol{D}}^{\star}$ using the spectral clustering method; as well as a direct application of the spectral clustering method on the linear transformation $\boldsymbol{\Gamma}$ obtained from opinion data [cf. (3.36)]. In Figure 3.11, we compare the error rate made by the community detection method against the number of stubborn agents S included, *i.e.,* the observation rank of the opinion data. The figure shows that the error rate decreases as S increases for both the 'two steps' method and the direct method proposed in Section 3.5. However, the direct method has significantly outperformed the 'two steps' method over all ranges of S. This discrepancy in the performance is potentially caused by the errors made in the first stage of network identification with (3.18), as the regular-regular network is not sufficiently sparse with respect to the considered range of S.

3.6.2 Nonlinear Opinion Dynamics

We conclude this section by presenting the numerical results on network identification when the underlying opinion dynamics is nonlinear. Similar to the last subsection, the following numerical experiments are done when the steady state opinions are observed for $K = 2S$ discussions and the opinions are observed noiselessly. The nonlinear response function is set as $\mu(|x|) = e^{-\sigma \cdot x^2}$ to model the decay of interaction strengths when the opinion difference between a pair of agents is large. We assume that the nonlinear function together with its parameter σ is known. Notice that the theoretical analysis conducted in Theorem 3.1 does not apply in this case as the response function is nonlinear.

Again, we begin by comparing the network identification performance on a small net-

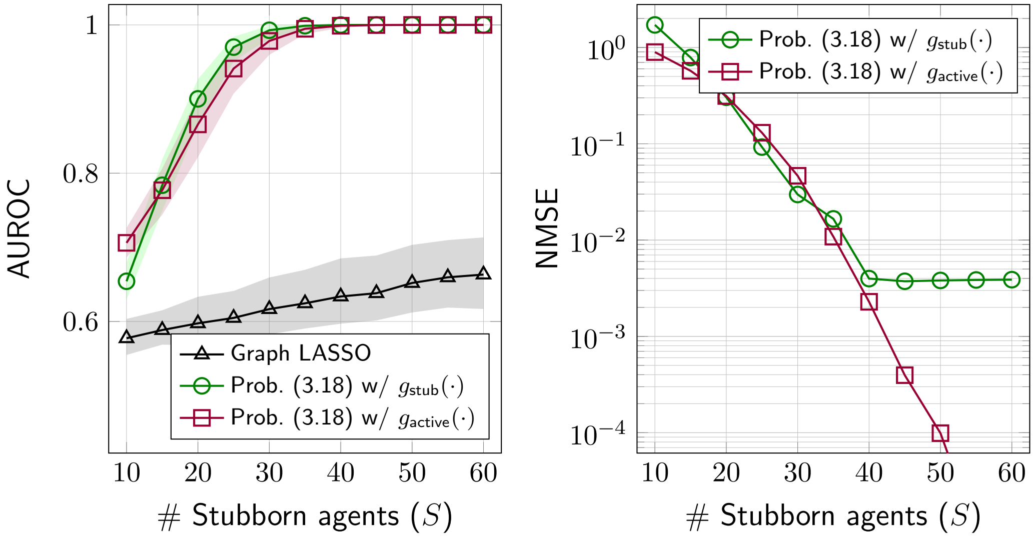

Figure 3.12: Comparing the network identification performance with nonlinear opinion dynamics, where the nonlinear function is set as $\mu(x) = e^{-5x^2}$. The regular-regular network has $n = 100$ nodes and is an ER graph with connectivity $\alpha = 0.08$. The shaded area shows the 5% / 95% percentile interval for the AUROC/AUPR performances.

work with $n = 100$ regular agents which are connected by a graph with the ER model of connectivity $\alpha = 0.08$. The stubborn-regular network follows the same construction as in H3.7 with $\ell = 5$, and we set the nonlinear response function as $\mu(x) = e^{-5x^2}$. The numerical results are shown in Figure 3.12. From the figure, we observe that the proposed method with (3.18) performs well, *i.e.,* it recovers the network topology and weights when the number of stubborn agents is $S \approx 35$. We observe that the graph LASSO heuristic has a much worse performance than in the similar case with DeGroot opinion dynamics [cf. Figure 3.5]. The advantage of using 'active sensing', *i.e.,* solving (3.18) with $g_{\mathsf{active}}(\cdot)$, compared to the case without 'active sensing', is demonstrated when we compare the NMSE performance. The 'active sensing' setting has a better NMSE when $S \geq 40$.

Lastly, we consider the case of identifying a large-scale network with $n = 1000$ regular agents and regular-regular connectivity of $\alpha = 2 \log n/n$, the stubborn-regular network follows the same construction as in H3.7 with $\ell = 5$. Moreover, we use a different nonlinear response function for the n regular agents, *i.e.,* we set $\mu_i(|x|) = e^{-\sigma_i x^2}$ where $\sigma_i \sim \mathcal{U}[0.2, 0.3]$ for $i \in \{1, ..., 500\}$ and $\sigma_i \sim \mathcal{U}[0.7, 0.8]$ for $i \in \{501, ..., 1000\}$. The simulation result is

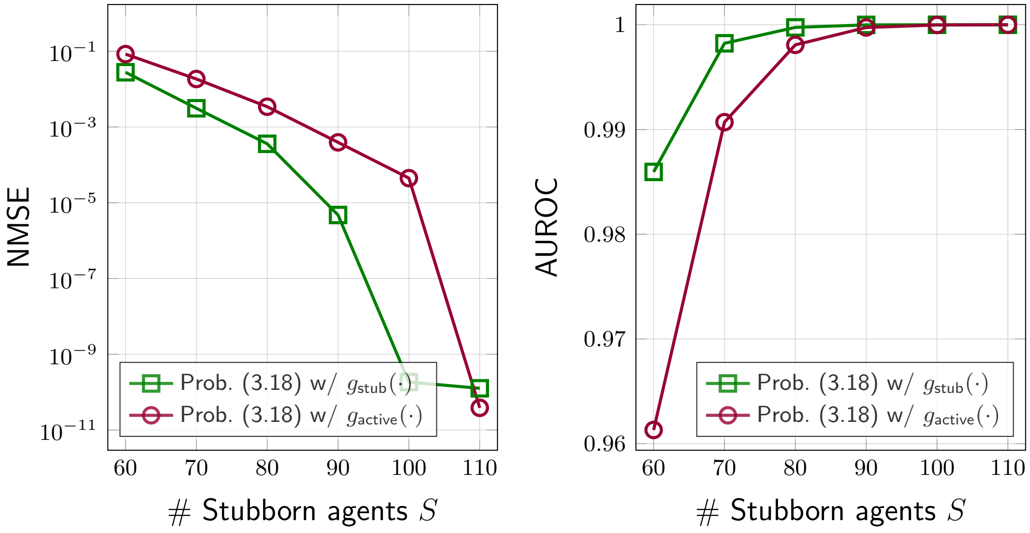

Figure 3.13: Comparing the network identification performance on large-scale network with $n = 1000$ using *heterogeneous* and nonlinear opinion dynamics.

shown in Figure 3.13. We observe similar behavior to the case with small networks, where the proposed methods are shown to identify the network using a number of stubborn agents that is only a fraction of n (in this case, it is $S \approx 100$ vs. $n = 1000$).

3.7 Chapter Summary

In this chapter, we have considered the modeling and identification of social networks based on opinion dynamics models. We first model the opinion dynamics as a discrete time non-linear dynamics with pairwise interactions. The considered model encompasses the linear DeGroot model as well as other models such as bounded confidence. We show that these models can be led into *consensus* when the underlying graph is strongly connected. This does not agree with the common observation on actual social networks, where consensus is often not reached. To remedy this, we introduce *stubborn agents* such that the steady state opinions are no longer in consensus. The steady-state opinions are *low-rank* with the rank dependent on the number of stubborn agents.

Upon describing and analyzing the opinion dynamics model, we propose a network identification method based on the sparse recovery techniques. We provide a sufficient condition for perfect network identification, which depends on the *sparseness* of the social network as well as the number of stubborn agents in it. In addition, a consistent estimator was derived to handle the case where the network dynamics is random. Simulation results on synthetic and real networks indicate that the networks can be identified with high accuracy.

Appendix

3.A Proof of Observation 3.1

Using the matrix-form of the opinion dynamics model (3.3), the steady state $\boldsymbol{x}^k := \boldsymbol{x}(\infty; k)$ satisfies the equilibrium condition:

$$\boldsymbol{x}^k = \Big(\mathrm{Diag}(\mathbf{1} - \big(\boldsymbol{A} \odot \boldsymbol{\mu}(\boldsymbol{x}^k)\big)\mathbf{1}) + \big(\boldsymbol{A} \odot \boldsymbol{\mu}(\boldsymbol{x}^k)\big)\Big)\ \boldsymbol{x}^k = \tilde{\boldsymbol{A}}^k \boldsymbol{x}^k\ , \tag{3.43}$$

where

$$\tilde{\boldsymbol{A}}^k := \mathrm{Diag}(\mathbf{1} - \big(\boldsymbol{A} \odot \boldsymbol{\mu}(\boldsymbol{x}^k)\big)\mathbf{1}) + \big(\boldsymbol{A} \odot \boldsymbol{\mu}(\boldsymbol{x}^k)\big)\ . \tag{3.44}$$

We can verify that $\tilde{\boldsymbol{A}}^k$ is a non-negative, stochastic matrix with an right eigenvector $\mathbf{1}$ and spectral radius of 1. Furthermore, under the assumption that $\mu(x) > 0$ for all $x \geq 0$, the matrix has the same support as $\boldsymbol{A}$, *i.e.,* it corresponds to the adjacency matrix of a strongly connected graph. As such, the matrix $\tilde{\boldsymbol{A}}^k$ is an *irreducible* matrix. Applying the Perron-Frobenius theorem [cf. Theorem 2.2] shows that $\mathbf{1}$ is the *only* right eigenvector of $\tilde{\boldsymbol{A}}^k$ with eigenvalue 1. This implies that for some $c(k) \in \mathbb{R}$, we have

$$\boldsymbol{x}^k = c(k)\mathbf{1}\ . \tag{3.45}$$

3.B Proof of Lemma 3.1

We prove the lemma by rewriting the equilibrium condition of the opinion dynamics (3.1) with *stubborn agents*. In particular, for any regular agent $i \in V_r$, we have:

$$0 = \sum_{j=1}^{S} B_{ij}\mu(|y_i^k - z_j^k|)(z_j^k - y_i^k) + \sum_{j=1}^{n} D_{ij}\mu(|y_j^k - y_i^k|)(y_j^k - y_i^k)\ . \tag{3.46}$$

In matrix form and using the definitions of $\boldsymbol{\mu}(\boldsymbol{y}^k)$ and $\tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)$, Eq. (3.46) can be written as:

$$f_i^k y_i^k - \big(\boldsymbol{d}_i^\top \odot \big[\boldsymbol{\mu}(\boldsymbol{y}^k)\big]_{i,:}\big)\boldsymbol{y}^k = \big(\boldsymbol{b}_i^\top \odot \big[\tilde{\boldsymbol{\mu}}(\boldsymbol{y}^k, \boldsymbol{z}^k)\big]_{i,:}\big)\boldsymbol{z}^k \, . \tag{3.47}$$

Stacking up the above for all $i \in V_r$ yields the desired result.

3.C Proof of Lemma 3.2

From the construction of $(\boldsymbol{B_\ell}, \boldsymbol{D_\ell})$, we observe the following chain

$$\begin{aligned}
(\boldsymbol{I} - \boldsymbol{D_\ell})^{-1}\boldsymbol{B_\ell} &= \Big(\boldsymbol{I} - \mathrm{Diag}(\boldsymbol{\ell}) - \mathrm{Diag}\Big(\frac{\boldsymbol{1} - \boldsymbol{\ell}}{\boldsymbol{1} - \mathrm{diag}(\boldsymbol{D})}\Big)\mathrm{off}(\boldsymbol{D})\Big)^{-1}\boldsymbol{B_\ell} \\
&= \Big(\mathrm{Diag}\Big(\frac{\boldsymbol{1} - \boldsymbol{\ell}}{\boldsymbol{1} - \mathrm{diag}(\boldsymbol{D})}\Big)\Big(\boldsymbol{I} - \mathrm{Diag}(\boldsymbol{D}) - \mathrm{off}(\boldsymbol{D})\Big)\Big)^{-1}\boldsymbol{B_\ell} \\
&= \Big(\mathrm{Diag}\Big(\frac{\boldsymbol{1} - \boldsymbol{\ell}}{\boldsymbol{1} - \mathrm{diag}(\boldsymbol{D})}\Big)\Big(\boldsymbol{I} - \boldsymbol{D}\Big)\Big)^{-1}\boldsymbol{B_\ell} \\
&= (\boldsymbol{I} - \tilde{\boldsymbol{D}})^{-1}\mathrm{Diag}\Big(\frac{\boldsymbol{1} - \boldsymbol{\ell}}{\boldsymbol{1} - \mathrm{diag}(\boldsymbol{D})}\Big)^{-1}\boldsymbol{B_\ell} = (\boldsymbol{I} - \boldsymbol{D})^{-1}\boldsymbol{B} \, .
\end{aligned} \tag{3.48}$$

This concludes the proof.

3.D Proof of Theorem 3.1

The proof of Theorem 3.1 is divided into two parts. The first part shows a sufficient condition for recovering $(\boldsymbol{B}, \boldsymbol{D})$ using (3.21); and the second part shows that the sufficient condition holds with high probability as $n \to \infty$.

Let $d(v)$ denote the degree of a vertex v. Our proof relies on the following definition of an unbalanced expander graph:

Definition 3.2 *An (α, δ)-unbalanced expander graph is an A, B-bipartite graph (bigraph) with $|A| = n, |B| = m$ with bounded left degrees*[2] *in $[d_l, d_u]$ which satisfies the following: (i) for all $v_i \in A$, we have $d(v_i) \in [d_l, d_u]$; (ii) for any $S \subseteq A$ with $|S| \leq \alpha n$, we have $\delta|E(S, B)| \leq |N(S)|$, where $E(S, B)$ is the set of edges connected from S to B and $N(S) =$*

[2] We follow the convention by calling A as the 'left' vertices such that the left degrees refer to the degrees of vertices in A.

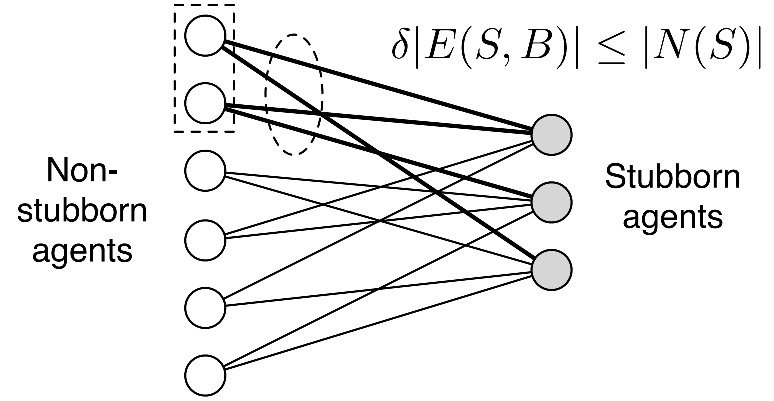

Figure 3.14: Illustrating the properties of an expander graph. In the above example bipartite graph, if $\alpha = 1/3$, δ is at most $3/4$ since $|E(S,B)| = 4$ and $|N(S)| = 3$ when S is the first two vertices in the set of ordinary agents.

$\{v_j \in B : \exists\ v_i \in S\ s.t.\ (v_j, v_i) \in E\}$ *is the neighbor set of* S *in* B.

We can imagine that A (*resp.* B) is the set of regular (*resp.* stubborn) agents and $E(A,B)$ represents the connection between stubborn and regular agents; see the illustration in Figure 3.14. We denote the collection of (α,δ)-unbalanced expander graphs by $\mathcal{G}(\alpha,\delta)$. Previous works [Berinde *et al.*(2008), Wang *et al.*(2011a), Khajehnejad *et al.*(2011), Gilbert and Indyk(2010)] have shown that the expander graph structure allows for the construction of measurement matrices with good sparse recovery performance.

We now proceed by showing the sufficient condition. Consider problem (3.21), we denote the support of $\hat{\boldsymbol{b}}_i - \boldsymbol{b}_i$ as $\Omega^i_{\boldsymbol{B}}$, where $|\Omega^i_{\boldsymbol{B}}| = \ell$ since the support information is incorporated when solving (3.21) and $\hat{\boldsymbol{b}}_i - \boldsymbol{b}_i$ is a sparse vector supported on $\Omega^i_{\boldsymbol{B}}$. We can thus treat the rows where $\hat{\boldsymbol{b}}_i - \boldsymbol{b}_i$ is supported on as some 'erasure bits'. In particular, the following rows-deleted linear system can be deduced from the last line in (3.24):

$$\overline{\boldsymbol{B}}^{\top}_{(\Omega^i_{\boldsymbol{B}})^c}(\boldsymbol{I}-\boldsymbol{D})^{-\top}(\boldsymbol{d}_i - \hat{\boldsymbol{d}}_i) = \boldsymbol{0}\ , \tag{3.49}$$

where $\boldsymbol{B}^{\top}_{(\Omega^i_{\boldsymbol{B}})^c}$ is an ℓ-rows-deleted matrix obtained from $\boldsymbol{B}^{\top}$.

We prove the sufficient condition by deriving a Restricted Isometry Property-1 (RIP-

1) condition for $\boldsymbol{A} = \boldsymbol{B}^{\top}_{(\Omega^i_{\boldsymbol{B}})^c}$ and its perturbation $\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{D})^{-\top}$. We define $a_{min} = \min_{ij \in \text{supp}(\boldsymbol{A})} A_{ij}$ and $a_{max} = \max_{ij \in \text{supp}(\boldsymbol{A})} A_{ij}$ and prove the following proposition:

Proposition 3.2 *Let $n > m$ and $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be a non-negative matrix that has the same support as the adjacency matrix of an (α, δ)-unbalanced bipartite expander graph with bounded left degrees $[d_l, d_u]$. Then $\boldsymbol{A}$ satisfies the RIP-1 property:*

$$\big(a_{min}\delta d_l - a_{max}(d_u - \delta d_l)\big)\|\boldsymbol{x}\|_1 \leq \|\boldsymbol{A}\boldsymbol{x}\|_1 \leq d_u a_{max}\|\boldsymbol{x}\|_1 \ , \tag{3.50}$$

for all k-sparse $\boldsymbol{x}$ such that $k \leq \alpha n$. Furthermore, we have

$$\upsilon^\star \|\boldsymbol{x}\|_1 \leq \|\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x}\|_1 \ , \tag{3.51}$$

where $\upsilon^\star = a_{min}\delta d_l - a_{max}(d_u - \delta d_l) - (1 - d_l a_{min})$.

Proof. The following proof is a generalization of [Khajehnejad *et al.*(2011), Appendix D]. First of all, the upper bound in (3.50) follows from $\|\boldsymbol{A}\boldsymbol{x}\|_1 \leq \|\boldsymbol{A}\|_{1,1}\|\boldsymbol{x}\|_1$, where $\|\boldsymbol{A}\|_{1,1}$ is the matrix norm induced by $\|\cdot\|_1$ on $\boldsymbol{A}$ [Horn and Johnson(1986)], *i.e.,*

$$\|\boldsymbol{A}\|_{1,1} = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |A_{ij}| \ . \tag{3.52}$$

Obviously we have $\|\boldsymbol{A}\|_{1,1} \leq d_u a_{max}$.

To prove the lower bound in (3.50), using the expander property, we observe that

$$\delta d_l |S| \leq \delta |E(S, B)| \leq |N(S)| \ , \tag{3.53}$$

for all $S \subseteq \text{supp}(\boldsymbol{x}) = \{i : x_i \neq 0\}$ and $|S| \leq \alpha n$. As a consequence of Hall's theorem [West(2000)], the bigraph induced by $\boldsymbol{A}$ contains δd_l disjoint matchings for $\text{supp}(\boldsymbol{x})$. We can thus decompose $\boldsymbol{A}$ as:

$$\boldsymbol{A} = \boldsymbol{A}_M + \boldsymbol{A}_C \ , \tag{3.54}$$

where the decomposition is based on dividing the support such that $\text{supp}(\boldsymbol{A}_M) \cap \text{supp}(\boldsymbol{A}_C) = \emptyset$. In particular, $\boldsymbol{A}_M$ is supported on the δd_l matchings for $\text{supp}(\boldsymbol{x})$; *i.e.,* by the matching property, each row of $\boldsymbol{A}_M$ has at most one non-zero, and each column of $\boldsymbol{A}_M$ has δd_l non-zeros, and the remainder $\boldsymbol{A}_C$ has at most $d_u - \delta d_l$ non-zeros per column. Applying the triangular inequality gives:

$$\|\boldsymbol{A}\boldsymbol{x}\|_1 \geq \|\boldsymbol{A}_M \boldsymbol{x}\|_1 - \|\boldsymbol{A}_C \boldsymbol{x}\|_1 \ , \tag{3.55}$$

since $\|\boldsymbol{A}_M \boldsymbol{x}\|_1 \geq a_{min} \delta d_l \|\boldsymbol{x}\|_1$ and $\|\boldsymbol{A}_C \boldsymbol{x}\|_1 \leq a_{max}(d_u - \delta d_l)\|\boldsymbol{x}\|_1$, this implies:

$$\|\boldsymbol{A}\boldsymbol{x}\|_1 \geq \big(a_{min}\delta d_l - a_{max}(d_u - \delta d_l)\big)\|\boldsymbol{x}\|_1 \ . \tag{3.56}$$

For the second part in the lemma, *i.e.,* (3.51), note that:

$$\|\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x}\|_1 \geq \|\boldsymbol{A}\boldsymbol{x}\|_1 - \|\boldsymbol{A}\boldsymbol{D}^\top(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x}\|_1 \ , \tag{3.57}$$

since $\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{A}\boldsymbol{D}^\top(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x}$. The latter quantity can be upper bounded by

$$\begin{aligned} \|\boldsymbol{A}\boldsymbol{D}^\top(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x}\|_1 &\leq \|\boldsymbol{A}\|_{1,1}\|\boldsymbol{D}^\top\|_{1,1}\|(\boldsymbol{I} - \boldsymbol{D})^{-\top}\|_{1,1}\|\boldsymbol{x}\|_1 \\ &\leq d_u a_{max} \frac{\|\boldsymbol{D}^\top\|_{1,1}}{1 - \|\boldsymbol{D}^\top\|_{1,1}}\|\boldsymbol{x}\|_1 \leq (1 - d_l a_{min})\|\boldsymbol{x}\|_1 \ , \end{aligned} \tag{3.58}$$

where in the second to last inequality, we used the property $\|(\boldsymbol{I} - \boldsymbol{C})^{-1}\| \leq 1/(1 - \|\boldsymbol{C}\|)$ for any $\|\boldsymbol{C}\| < 1$ [Horn and Johnson(1986)]; and in the last inequality, we used the fact that $1 - d_u a_{max} \leq \|\boldsymbol{D}^\top\|_{1,1} \leq 1 - d_l a_{min}$ (note that each row in $\overline{\boldsymbol{D}}$ sums to at most $1 - d_l a_{min}$ and at least $1 - d_u a_{max}$). Combining (3.56), (3.57) and (3.58) yields the desired inequality. **Q.E.D.**

A sufficient condition for the desired ℓ_0 recovery result can be obtained by proving the following corollary:

Corollary 3.1 *Let the conditions from Proposition 3.2 on $\boldsymbol{A}$ holds. Suppose that both $\boldsymbol{x}_1, \boldsymbol{x}_2$ are $(k/2)$-sparse such that $k \leq \alpha n$ and:*

$$\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x}_1 = \boldsymbol{A}(\boldsymbol{I} - \boldsymbol{D})^{-\top}\boldsymbol{x}_2 \ , \tag{3.59}$$

then $\boldsymbol{x}_1 = \boldsymbol{x}_2$ if

$$\upsilon^\star = a_{min}\delta d_l - a_{max}(d_u - \delta d_l) - (1 - d_l a_{min}) > 0 \ . \tag{3.60}$$

Proof. Observe that $\boldsymbol{x}_1 - \boldsymbol{x}_2$ is at most k-sparse, using Proposition 3.2, we have

$$\upsilon^\star \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_1 \leq \|\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{D})^{-\top}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\|_1 = 0 \ . \tag{3.61}$$

This implies that $\boldsymbol{x}_1 = \boldsymbol{x}_2$. **Q.E.D.**

As $\boldsymbol{d}_i$ is $k/2$-sparse, $b_{min} \leq a_{min}$ and $b_{max} \geq a_{max}$, Eq. (3.25) and Corollary 3.1 guarantee that $\boldsymbol{d}_i$ is the *unique* solution out of all $k/2$-sparse vectors that $\boldsymbol{d}_i$ satisfies (3.49). This means that any $\hat{\boldsymbol{d}}_i$ that satisfies (3.49) must be either $\boldsymbol{d}_i$ or have $\|\hat{\boldsymbol{d}}_i\|_0 > (k/2)$. Since the optimization problem (3.21) finds the sparsest solution satisfying (3.49), we must have $\hat{\boldsymbol{d}}_i^\star = \boldsymbol{d}_i$ for all i. Furthermore, this implies $\hat{\boldsymbol{b}}_i^\star = \boldsymbol{b}_i$ in (3.24) and we have the desired result as $(\hat{\boldsymbol{B}}^\star, \hat{\boldsymbol{D}}^\star) = (\boldsymbol{B}, \boldsymbol{D})$.

The second part of our proof shows that for all i, the support set of the ℓ-rows-deleted matrix $\boldsymbol{B}^\top_{(\Omega^i_{\boldsymbol{B}})^c}$ corresponds to an (α, δ)-expander graph with high probability. Our plan is to first prove that the corresponding bipartite graph has a bounded degree $r \in [\ell - 1, \ell]$ with high probability (w.h.p.), and then show that a randomly constructed bipartite with bounded degree $r \in [\ell - 1, \ell]$ is also an expander graph w.h.p..

Let us prove the following proposition:

Proposition 3.3 *Let G be a random A, B-bigraph with $|A| = n$, $|B| = n_s = \beta n$, constructed by randomly connecting d vertices from A to each vertex of B. All of the subgraphs*

$G_1, ..., G_n$ have left degree $r \in [\ell-1, \ell]$ with high probability (as $n \to \infty$) if each of these subgraphs is formed by randomly deleting ℓ vertices from B in G.

Proof. We lower bound the desired probability as follows:

$$\begin{aligned}
&\Pr\Big(G_1, ..., G_n = \text{bipartite with (left) deg. } r \in [\ell-1, \ell]\Big) \\
&= 1 - \Pr\Big(\cup_{i=1}^{n} \left(G_i = \text{bipartite with min. deg. } r < \ell - 1\right)\Big) \\
&\geq 1 - n \cdot \Pr\Big(\cup_{k=1}^{n} \left(d(v_k) < \ell - 1,\ v_k \in A_i,\ A_i \subseteq V(G_i)\right)\Big) \\
&\geq 1 - n^2 \cdot \Pr\Big(d(v_k) < \ell - 1,\ v_k \in A_i,\ A_i \subseteq V(G_i)\Big)\ .
\end{aligned} \tag{3.62}$$

Note that the event described in the last term is equivalent to deleting at least 2 neighbors of $v_k \in A_i$ from B. As the neighbors of A are also randomly selected, the latter probability can be upper bounded by:

$$\begin{aligned}
\Pr\Big(d(v_k) < \ell - 1,\ v_k \in A_i,\ A_i \subseteq V(G_i)\Big) &= \Pr\big(d(v_k) = 0 \cup \cdots \cup d(v_k) = \ell - 2\big) \\
&\leq (\ell - 1) \cdot \left(\frac{\ell^2}{(\beta n)^2}\right)^2 = (\ell - 1) \cdot \left(\frac{\ell}{\beta n}\right)^4\ .
\end{aligned} \tag{3.63}$$

Plugging this back into (3.62) yields the desired result. **Q.E.D.**

The proof of Theorem 3.1 is completed by the proposition:

Proposition 3.4 *Let G be a random A, B-bipartite graph with $|A| = n$, $|B| = m = \beta' n = S - \ell$, constructed by randomly connecting $r \in [\ell-1, \ell]$ vertices from A to each vertex of B. Then G is an $(\alpha, 1 - 1/(\ell-1))$-expander graph with high probability if $\ell \geq 4$, $\alpha < \beta'$ and $\ell - 1 > (H(\alpha) + \beta' H(\alpha/\beta'))/\alpha \log(\beta'/\alpha)$.*

Proof. The following proof is similar in flavor to the proof of [Khajehnejad *et al.*(2011), Proposition 1], with the additional complexity that the left degree is variable. For simplicity, we denote $\boldsymbol{A}$ as the adjacency matrix of G and let $E_{i_1,...,i_r}$ be the event such that $\boldsymbol{A}_{:,i_1,...,i_r}$ contains at least $m - r + 1$ zero rows, where $\boldsymbol{A}_{:,i_1,...,i_r}$ is the submatrix formed by choosing the $\{i_1, ..., i_r\}$ columns. Note that if $r \leq \alpha n$ and $E_{i_1,...,i_r}$ occurs, $G \notin \mathcal{G}(\alpha, 1 - 1/(\ell-1))$

since $(1-1/(\ell-1))|E(\{i_1,...,i_r\})| \geq r > r-1 = |N(\{\{i_1,...,i_r\})|$. The failure probability can thus be upper bounded as:

$$\begin{aligned}
&\Pr\Big(G \notin \mathcal{G}(\alpha, 1-1/(\ell-1))\Big) \\
&\leq \Pr\Big(\bigcup_{\ell-1 \leq r \leq \alpha n, 1 \leq i_1 < i_2 < \cdots < i_r} E_{i_1,\ldots,i_r}\Big) \leq \sum_{r=\ell-1}^{\alpha n} \binom{n}{r} \Pr(E_{1,\ldots,r}) \ .
\end{aligned} \tag{3.64}$$

Suppose that there are $r-s$ columns with $\ell-1$ non-zero entries and s columns with ℓ non-zero entries; hence we have $\binom{\beta n}{\ell-1}^{r-s}\binom{\beta n}{\ell}^{s}$ possible sub-matrices to choose from. Now, a necessary condition for $E_{1,\ldots,r}$ is such that all the non-zero entries are contained in a sub-sub-matrix of size $r \times r$. There are at most $\binom{r}{\ell-1}^{r-s}\binom{r}{\ell}^{s}$ possible configurations and $\binom{\beta n}{r}$ such sub-sub-matrices. For this case, we obtain the upper bound:

$$\begin{aligned}
\Pr(E_{1,\ldots,r},\ \text{fix } s) &\leq \frac{\binom{\beta' n}{r}\binom{r}{\ell-1}^{r-s}\binom{r}{\ell}^{s}}{\binom{\beta' n}{\ell-1}^{r-s}\binom{\beta' n}{\ell}^{s}} \\
&\leq \binom{\beta' n}{r} \cdot \left(\frac{r}{\beta' n}\right)^{(r-s)(\ell-1)} \cdot \left(\frac{r}{\beta' n}\right)^{s\ell} \\
&= \binom{\beta' n}{r} \cdot \left(\frac{r}{\beta' n}\right)^{(r-s)(\ell-1)+s\ell} ,
\end{aligned} \tag{3.65}$$

where we used the fact that $\binom{r}{\ell}/\binom{m}{\ell} \leq (r/m)^{\ell}$ if $r < m$ in the first inequality. Taking the union bound for all configurations $s \in [0, r]$ gives:

$$\begin{aligned}
&\Pr(E_{1,\ldots,r}) \leq \textstyle\sum_{s=0}^{r} \Pr(E_{1,\ldots,r},\ \text{fix } s) \\
&\leq \binom{\beta' n}{r} \cdot \left(\left(\frac{r}{\beta' n}\right)^{r(\ell-1)} + \left(\frac{r}{\beta' n}\right)^{r(\ell-1)+1} + \cdots + \left(\frac{r}{\beta' n}\right)^{r\ell}\right) \\
&= \binom{\beta' n}{r} \cdot \frac{1}{1-r/(\beta' n)} \left(\left(\frac{r}{\beta' n}\right)^{r(\ell-1)} - \left(\frac{r}{\beta' n}\right)^{r\ell+1}\right) \\
&< \frac{1}{1-\alpha/\beta'} \binom{\beta' n}{r} \cdot \left(\frac{r}{\beta' n}\right)^{r(\ell-1)} .
\end{aligned} \tag{3.66}$$

The second equality is due to the geometric series and the last inequality is due to $r \leq \alpha n$.

We thus have:

$$\Pr\Big(G \notin \mathcal{G}(\alpha, 1 - 1/(\ell-1))\Big) \leq \frac{1}{1 - \frac{\alpha}{\beta'}} \sum_{r=\ell-1}^{\alpha n} \binom{n}{r} \binom{\beta' n}{r} \left(\frac{r}{\beta' n}\right)^{r(\ell-1)} . \tag{3.67}$$

The remainder of the proof follows from that of [Khajehnejad *et al.*(2011)]; *i.e.,* Lemma A.1 and A.2, through replacing ℓ by $\ell-1$. In particular, if $\ell-1 > (H(\alpha)+\beta' H(\alpha/\beta'))/\alpha \log(\beta'/\alpha)$, we can show that

$$\Pr\Big(G \notin \mathcal{G}(\alpha, 1 - 1/(\ell-1))\Big) = \mathcal{O}(n^{1-(\ell-1)(d-3)}) . \tag{3.68}$$

This completes the proof. **Q.E.D.**

Combining Proposition 3.3 and 3.4 implies that the ℓ-rows-deleted sensing matrix $\boldsymbol{B}^\top_{(\Omega^i_{\boldsymbol{B}})^c}$ corresponds to an $(\alpha, 1-1/(\ell-1))$-expander graph with high probability. Therefore, the conclusion in Corollary 3.1 follows by setting $\delta = 1 - 1/(\ell-1)$. Moreover, by applying the union bound, the probability of failure is upper bounded as:

$$\Pr(Fail) \leq \left(\frac{\ell}{\beta}\right)^4 \frac{\ell-1}{n^2} + \mathcal{O}(n^{2-(\ell-1)(\ell-3)}) , \tag{3.69}$$

which vanishes as $n \to \infty$.

3.E Proof of Theorem 3.2

To simplify the notations, we drop the dependence on the discussion index k for the opinion vectors $\boldsymbol{x}(t;k)$ and the trust matrices $\boldsymbol{A}(t;k)$. We first prove that the estimator is unbiased. Consider the following chain:

$$\mathbb{E}[\hat{\boldsymbol{x}}(\mathcal{T}_k)|\boldsymbol{x}(0)] = \frac{1}{|\mathcal{T}_k|} \sum_{t_i \in \mathcal{T}_k} \mathbb{E}[\hat{\boldsymbol{x}}(t_i)|\boldsymbol{x}(0)] = \frac{1}{|\mathcal{T}_k|} \sum_{t_i \in \mathcal{T}_k} \boldsymbol{A}^{t_i} \boldsymbol{x}(0) = \boldsymbol{A}^{\infty} \boldsymbol{x}(0) , \tag{3.70}$$

where we used the fact that $T_o \to \infty$ and $t_i \geq T_o$ for all t_i in the last equality.

Next, we prove that the estimator is asymptotically consistent, *i.e.,* (3.34). Without loss of generality, we let $t_1 < t_2 < \ldots < t_{|\mathcal{T}_k|}$ as the sampling instances and drop the dependence on k in $\boldsymbol{A}(t;k)$ for simplicity. The following shorthand notation will be useful:

$$\boldsymbol{\Phi}(s,t) \triangleq \boldsymbol{A}(t)\boldsymbol{A}(t-1)\ldots\boldsymbol{A}(s+1)\boldsymbol{A}(s) \ , \tag{3.71}$$

where $t \geq s$ and $\boldsymbol{\Phi}(s,t)$ is a random matrix. Our proof involves the following lemma:

Lemma 3.3 *When $|t-s| \to \infty$, the random matrix $\boldsymbol{\Phi}(s,t)$ converges almost surely to the following:*

$$\lim_{|t-s|\to\infty} \boldsymbol{\Phi}(s,t) = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B}(s,t) & \boldsymbol{0} \end{pmatrix}, \tag{3.72}$$

where $\boldsymbol{B}(s,t) = \sum_{q=s}^{t}(\boldsymbol{D}(t)\ldots\boldsymbol{D}(q))\boldsymbol{B}(q)$. Moreover, $\boldsymbol{B}(s,t)$ is bounded almost surely.

Proof: We first establish the almost sure convergence of $\boldsymbol{D}(t)\boldsymbol{D}(t-1)\ldots\boldsymbol{D}(s)$ to $\boldsymbol{0}$. Define

$$\beta(s,t) \triangleq \|\boldsymbol{D}(t)\boldsymbol{D}(t-1)\ldots\boldsymbol{D}(s)\|_2 \ , \tag{3.73}$$

and observe the following chain

$$\begin{aligned} \mathbb{E}[\beta(s,t)|\beta(s,t-1),...,\beta(s,s)] &= \mathbb{E}[\|\boldsymbol{D}(t)\boldsymbol{D}(t-1)\ldots\boldsymbol{D}(s)\|_2|\beta(s,t-1)] \\ &\leq \mathbb{E}[\|\boldsymbol{D}(t)\|_2\|\boldsymbol{D}(t-1)\ldots\boldsymbol{D}(s)\|_2|\beta(s,t-1)] \\ &= \mathbb{E}[\|\boldsymbol{D}(t)\|_2]\beta(s,t-1) \leq c\beta(s,t-1) \ , \end{aligned} \tag{3.74}$$

where $c = \|\overline{\boldsymbol{D}}\|_2 < 1$. The almost sure convergence of $\beta(s,t)$ follows from [Polyak(1987), Lemma 7]. Now, expanding the multiplication (3.71) yields:

$$\boldsymbol{\Phi}(s,t) = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{B}(s,t) & \boldsymbol{D}(t)\ldots\boldsymbol{D}(s) \end{pmatrix} . \tag{3.75}$$

The desired result is achieved by observing $\boldsymbol{D}(t)\ldots\boldsymbol{D}(s) \to \boldsymbol{0}$ as $|t-s| \to \infty$.

Lastly, the almost sure boundedness of $\boldsymbol{B}(s,t)$ can be obtained from the obvious fact that $\boldsymbol{\Phi}(s,t)$ is a non-negative and stochastic matrix. **Q.E.D.**

We consider the following:

$$\mathbb{E}[\|\hat{\boldsymbol{x}}(\mathcal{T}_k) - \overline{\boldsymbol{x}}(\infty)\|_2^2 | \boldsymbol{x}(0)] = \mathbb{E}\Big[\Big\| \frac{1}{|\mathcal{T}_k|} \sum_{t_i \in \mathcal{T}_k} \big(\hat{\boldsymbol{x}}(t_i) - \overline{\boldsymbol{x}}(\infty)\big) \Big\|_2^2 \;\Big|\; \boldsymbol{x}(0)\Big] \;. \tag{3.76}$$

Recall that $\hat{\boldsymbol{x}}(t_i) = \boldsymbol{x}(t_i) + \boldsymbol{n}(t_i)$ and the noise term $\boldsymbol{n}(t_i)$ is independent of $\boldsymbol{A}(t)$ for all t. The above expression reduces to:

$$\mathbb{E}\Big[\Big\| \tfrac{1}{|\mathcal{T}_k|} \textstyle\sum_{t_i \in \mathcal{T}_k} \big(\boldsymbol{x}(t_i) - \overline{\boldsymbol{x}}(\infty)\big) \Big\|_2^2 | \boldsymbol{x}(0)\Big] + \mathbb{E}\Big[\Big\| \tfrac{1}{|\mathcal{T}_k|} \textstyle\sum_{t_i \in \mathcal{T}_k} \boldsymbol{n}(t_i) \Big\|_2^2\Big] \;. \tag{3.77}$$

It is easy to check that the latter term vanishes when $|\mathcal{T}_k| \to \infty$. We thus focus on the former term, which gives

$$\begin{aligned} \mathbb{E}\Big[\Big\| \frac{1}{|\mathcal{T}_k|} \sum_{t_i \in \mathcal{T}_k} \big(\boldsymbol{x}(t_i) - \overline{\boldsymbol{x}}(\infty)\big) \Big\|_2^2 | \boldsymbol{x}(0)\Big] &= \frac{1}{|\mathcal{T}_k|^2} \mathbb{E}\Big[\Big\| \sum_{t_i \in \mathcal{T}_k} \big(\boldsymbol{\Phi}(0,t_i) - \boldsymbol{A}^\infty\big)\boldsymbol{x}(0) \Big\|_2^2\Big] \\ &= \frac{1}{|\mathcal{T}_k|^2} \mathbb{E}\big[\mathrm{Tr}\big(\boldsymbol{\Xi} \boldsymbol{x}(0)\boldsymbol{x}(0)^\top\big)\big] \;, \end{aligned} \tag{3.78}$$

where

$$\boldsymbol{\Xi} = \sum_{t_j \in \mathcal{T}_k} \big(\boldsymbol{\Phi}(0,t_j) - \boldsymbol{A}^\infty\big)^\top \sum_{t_i \in \mathcal{T}_k} \big(\boldsymbol{\Phi}(0,t_i) - \boldsymbol{A}^\infty\big) \;. \tag{3.79}$$

Expanding the above product yields two groups of terms — when $t_i = t_j$ and when $t_i \neq t_j$. When $t_i = t_j$, using $T_o \to \infty$ and Lemma 3.3, it is straightforward to show that:

$$\|\mathbb{E}\big[\big(\boldsymbol{\Phi}(0,t_i) - \boldsymbol{A}^\infty\big)^\top \big(\boldsymbol{\Phi}(0,t_i) - \boldsymbol{A}^\infty\big)\big]\| \leq C \;, \tag{3.80}$$

for some constant $C < \infty$. As a matter of fact, we observe that the above term will not vanish at all.

For the latter case, we assume $t_j > t_i$. We have

$$\begin{aligned} &\big(\boldsymbol{\Phi}(0,t_j) - \boldsymbol{A}^\infty\big)^\top \big(\boldsymbol{\Phi}(0,t_i) - \boldsymbol{A}^\infty\big) \\ &= \big(\boldsymbol{\Phi}(t_i+1,t_j)\boldsymbol{\Phi}(0,t_i) - \boldsymbol{A}^\infty\big)^\top \big(\boldsymbol{\Phi}(0,t_i) - \boldsymbol{A}^\infty\big) \;. \end{aligned} \tag{3.81}$$

Taking expectation of the above term gives:

$$\mathbb{E}\big[\big(\boldsymbol{\Phi}(0,t_i)-\boldsymbol{A}^{\infty}\big)^{\top}\boldsymbol{A}^{t_j-t_i}\big(\boldsymbol{\Phi}(0,t_i)-\boldsymbol{A}^{\infty}\big)\big]\;, \tag{3.82}$$

where we used the fact that $\boldsymbol{\Phi}(t_i+1,t_j)$ is independent of the other random variables in the expression and $\boldsymbol{A}^{\infty}\boldsymbol{A}^{\ell}=\boldsymbol{A}^{\infty}$ for any finite ℓ. Now, note that

$$\boldsymbol{A}^{t_j-t_i}=\boldsymbol{A}^{\infty}+\mathcal{O}(\lambda^{t_j-t_i})\;, \tag{3.83}$$

for some $0<\lambda\triangleq\lambda_{max}(\boldsymbol{D})<1$. This is due to the fact that $\boldsymbol{D}$ is sub-stochastic.

As $T_o\to\infty$ and by invoking Lemma 3.3, the matrix $(\boldsymbol{\Phi}(0,t_i)-\boldsymbol{A}^{\infty})$ has almost surely *only* non-empty entries in the lower left block. Carrying out the block matrix multiplications and using the boundedless of $\boldsymbol{\Phi}(0,t_i)$ gives

$$\big\|\mathbb{E}\big[\big(\boldsymbol{\Phi}(0,t_j)-\boldsymbol{A}^{\infty}\big)^{\top}\big(\boldsymbol{\Phi}(0,t_i)-\boldsymbol{A}^{\infty}\big)\big]\big\|\le\mathcal{O}(\lambda^{t_j-t_i})\;. \tag{3.84}$$

Combining these results, we can show

$$\frac{\mathbb{E}\big[\mathrm{Tr}\big(\boldsymbol{\Xi}\boldsymbol{x}(0)\boldsymbol{x}(0)^{\top}\big)\big]}{|\mathcal{T}_k|^2}\le\frac{C'}{|\mathcal{T}_k|}\Big(\sum_{i=0}^{|\mathcal{T}_k|-1}\lambda^{\min_k|t_{k+i}-t_k|}\Big)\;, \tag{3.85}$$

for some $C'<\infty$. Notice that $\min_\ell|t_{\ell+i}-t_\ell|\ge i$ and the terms inside the bracket can be upper bounded by the summable geometric series $\sum_{i=0}^{|\mathcal{T}_k|-1}\lambda^i$, since $\lambda<1$. Consequently, the mean square error goes to zero as $|\mathcal{T}_k|\to\infty$. The estimator (3.32) is consistent.

3.F Proof of Proposition 3.1

Denote the rank-C approximation of $\tilde{\boldsymbol{D}}$ as $[\tilde{\boldsymbol{D}}]_C:=\boldsymbol{V}_C\mathrm{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^{\top}$, and define the shorthand notation that $\tilde{\boldsymbol{B}}:=\boldsymbol{B}\boldsymbol{Q}_C$, we observe that

$$\mathcal{R}([\tilde{\boldsymbol{D}}]_C)=\mathcal{R}([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}})\;, \tag{3.86}$$

where $\mathcal{R}(\boldsymbol{X})$ denotes the range space of a matrix $\boldsymbol{X}$ and the equality above is due to the assumption that $[\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}}$ has rank C, such that the linear transformation on the right does not modify the range space of $[\tilde{\boldsymbol{D}}]_C$. If we denote the columns of $\tilde{\boldsymbol{V}}_C$ as the top C left singular vectors of $[\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}}$, the equality above shows that the two products are equal $\boldsymbol{V}_C\boldsymbol{V}_C^\top = \tilde{\boldsymbol{V}}_C\tilde{\boldsymbol{V}}_C^\top$.

Similarly, we define $[\tilde{\boldsymbol{D}}\boldsymbol{B}]_C = \boldsymbol{P}_C\text{diag}(\boldsymbol{\sigma}_C)\boldsymbol{Q}_C^\top$ as the rank-C approximation of the sketch and observe that:

$$\mathcal{R}([\tilde{\boldsymbol{D}}\boldsymbol{B}]_C) = \mathcal{R}(\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}) \tag{3.87}$$

where the equality is due to the fact that $\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}} = \tilde{\boldsymbol{D}}\boldsymbol{B}\boldsymbol{Q}_C = \boldsymbol{P}_C\text{diag}(\boldsymbol{\sigma}_C)$ as the columns of $\boldsymbol{Q}_C$ are the top C right singular vectors. Likewise, if the columns of $\tilde{\boldsymbol{P}}_C$ are the top C left singular vectors of $\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}$, then $\boldsymbol{P}_C\boldsymbol{P}_C^\top = \tilde{\boldsymbol{P}}_C\tilde{\boldsymbol{P}}_C^\top$.

Furthermore, we observe that

$$\mathcal{R}([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}})\perp\mathcal{R}((\tilde{\boldsymbol{D}} - [\tilde{\boldsymbol{D}}]_C)\tilde{\boldsymbol{B}})\ . \tag{3.88}$$

Invoking [Boutsidis *et al.*(2015), Lemma 8] through setting $\mathbf{D} = \tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}$, $\mathbf{C} = [\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}}$ and $\mathbf{E} = [\tilde{\boldsymbol{D}}]_{N-C}\tilde{\boldsymbol{B}} := (\tilde{\boldsymbol{D}} - [\tilde{\boldsymbol{D}}]_C)\tilde{\boldsymbol{B}}$ therein, we can show the following:

$$\|\tilde{\boldsymbol{V}}_C\tilde{\boldsymbol{V}}_C^\top - \tilde{\boldsymbol{P}}_C\tilde{\boldsymbol{P}}_C^\top\|_2^2 = 1 - \beta_C\Big([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}}\big((\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}})^\top\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}\big)^\dagger([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}})^\top\Big)\ . \tag{3.89}$$

Denote the matrix in the middle of the expression above as $\mathbf{\Pi} := (\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}})^\top\tilde{\boldsymbol{D}}\tilde{\boldsymbol{B}}$. Now, under the stated assumptions the $C\times C$ matrix $\mathbf{\Pi}$ is non-singular. We observe the following chain:

$$\begin{aligned}\beta_C\Big([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}}\mathbf{\Pi}^{-1}([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}})^\top\Big) &= \beta_C\Big(\text{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}}\mathbf{\Pi}^{-1}(\text{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}})^\top\Big)\\ &= \frac{1}{\beta_1\Big((\text{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}})^{-\top}\mathbf{\Pi}(\text{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}})^{-1}\Big)}\ ,\end{aligned} \tag{3.90}$$

where the first equality is due to $\beta_C(\boldsymbol{V}\boldsymbol{A}\boldsymbol{V}^\top) = \beta_C(\boldsymbol{A})$ for any $\boldsymbol{V}\in\mathbb{R}^{N\times C}$ with orthogonal columns. Moreover, observe that $\mathbf{\Pi}$ has the following decomposition:

$$\begin{aligned}\mathbf{\Pi} = (\tilde{\boldsymbol{D}}\boldsymbol{B}\boldsymbol{Q}_C)^\top \tilde{\boldsymbol{D}}\boldsymbol{B}\boldsymbol{Q}_C = (\mathrm{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top \boldsymbol{B}\boldsymbol{Q}_C)^\top(\mathrm{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top \boldsymbol{B}\boldsymbol{Q}_C)\\ + \boldsymbol{Q}_C^\top \boldsymbol{B}^\top \boldsymbol{U}_{N-C}\mathrm{diag}(\boldsymbol{\lambda}_{N-C})^2\boldsymbol{U}_{N-C}^\top \boldsymbol{B}\boldsymbol{Q}_C\ .\end{aligned} \tag{3.91}$$

This yields:

$$\begin{aligned}&\beta_C\Big([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}}\mathbf{\Pi}^{-1}([\tilde{\boldsymbol{D}}]_C\tilde{\boldsymbol{B}})^\top\Big)\\ &= \Big(1 + \beta_1\Big((\mathrm{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}})^{-\top}\tilde{\boldsymbol{B}}^\top \boldsymbol{U}_{N-C}\\ &\qquad \mathrm{diag}(\boldsymbol{\lambda}_{N-C})^2\boldsymbol{U}_{N-C}^\top\tilde{\boldsymbol{B}}(\mathrm{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}})^{-1}\Big)\Big)^{-1}\\ &= \frac{1}{1 + \Big\|\big(\mathrm{diag}(\boldsymbol{\lambda}_{N-C})\boldsymbol{U}_{N-C}^\top\tilde{\boldsymbol{B}}\big)\big(\mathrm{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}}\big)^{-1}\Big\|_2^2} = \Big(1+\gamma^2\Big)^{-1}\ ,\end{aligned} \tag{3.92}$$

where we have defined γ such that:

$$\begin{aligned}\gamma &:= \|\mathrm{diag}(\boldsymbol{\lambda}_{N-C})\boldsymbol{U}_{N-C}^\top\tilde{\boldsymbol{B}}(\mathrm{diag}(\boldsymbol{\lambda}_C)\boldsymbol{U}_C^\top\tilde{\boldsymbol{B}})^{-1}\|_2\\ &\leq \left(\frac{\lambda_{C+1}}{\lambda_C}\right)\cdot\|\boldsymbol{U}_{N-C}^\top\boldsymbol{B}\boldsymbol{Q}_C\|_2\|(\boldsymbol{U}_C^\top\boldsymbol{B}\boldsymbol{Q}_C)^{-1}\|_2\ ,\end{aligned} \tag{3.93}$$

as desired. This concludes the proof of our claim.

4 Network RADAR for Gene Dynamics

This chapter is the second part of our study on *modeling and identification* of networks. In particular, our focus is on the gene regulatory networks (GRNs) dynamics.

4.1 Context and Background

Similar to social networks, studies on GRNs' dynamics modeling can be found in a number of prior work, e.g., [Kang *et al.*(2015), Barzel and Biham(2009)]. Here, the states of genes are characterized by their concentration levels. The state-of-the-art model, e.g., the Michaelis-Menten dynamics [Menten and Michaelis(1913)], postulates that the rate of change in the concentration levels of a gene is a *linear* combination of the nonlinearly distorted concentration levels of its regulating genes. This is in contrast to that for the opinion dynamics, where the changes in opinions is dependent on the convex combination of a nonlinear distorted version of the *difference* in the opinions, as illustrated below:

$$\begin{aligned} &\textit{(Opinion Dynamics)} \quad x_i(t+1) - x_i(t) = \sum_{j=1}^{n} A_{ij} h\big(x_j(t) - x_i(t)\big) \;, \\ &\textit{(Gene Dynamics)} \quad \frac{dx_i(t)}{dt} = f(x_i(t)) \cdot \sum_{j=1}^{n} A_{ij} h(x_j(t)) \;. \end{aligned} \tag{4.1}$$

The difference between the two types of dynamics above may appear to be subtle at the first sight. However, there are two important differences for the effects of perturbing the steady states in these dynamics. Firstly, the gene dynamics are perturbed through *knocking out* one or a few genes; while the perturbation model studied for the opinion dynamics relies on injecting different initial opinions from the stubborn agents. Secondly, the structure of the dynamics equations leads to a substantially different characterization of the steady states when the system is perturbed.

Motivated by the applications in understanding diseases and discovering new drugs [Barabasi and Oltvai(2004)], the GRN identification problem has been considered in various work [Huynh-Thu *et al.*(2010), Haury *et al.*(2012)]. However, most of the prior work are based on machine learning heuristics or they rely on observing *high-rank* data, the latter is typically unavailable since the data collection requires performing actual experiments on organisms, which are costly and time consuming. One of the aims of this chapter is to study the minimum number of experiments required to accurately identify the GRN. We apply tools from sparse recovery and prove that a *sparse* GRN can be recovered with only observations from a *few* perturbation experiments. Compared to opinion dynamics model in the previous chapter, our recoverability result requires a few more assumptions, yet a similar conclusion can be drawn on the number of perturbation experiments required.

In the rest of the chapter, Section 4.2 set up the dynamics model and a mathematical description of the type of perturbation experiments involved. Then, Section 4.4 shows the provable guarantees on network recoverability; Section 4.5 presents a robustified framework for network inference. Finally, Section 4.6 concludes the chapter with results from our numerical experiments.

4.2 Gene Dynamics Model

We following the common notations in Section 2.1 to define the gene regulatory networks (GRNs). We consider an GRN with n genes such that $V = [n]$, and the weight matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ encodes the strengths of regulation between the genes. The time dependent expression (or concentration) level of gene i, $x_i(t)$, follows the nonlinear dynamics:

$$\dot{x}_i(t) = -x_i(t) + \sum_{j=1}^{n} A_{ij} \cdot h(x_j(t); \boldsymbol{b}) \ , \ \forall \ i \in [n] \ . \tag{4.2}$$

where $\dot{x}_i(t) := dx_i(t)/dt$ is the rate of change of the expression level $x_i(t)$. The first term on the right hand side captures the gene i's self dynamics, capturing processes such as degradation, and the sum captures the impact of i's interacting partners; $h(x; \boldsymbol{b})$ is the nonlinear continuous response function describing the regulatory mechanism such that $\boldsymbol{b}$ describes its parameters. For instance, setting $h(x) = c \cdot x^b$ describes chemical activation, where according to the law of mass action b is the level of cooperation in the activating process. Another frequently used response function is the Hill function $h(x) = x^b/(1+x^b)$, a saturating function such that $\lim_{x\to\infty} h(x) = 1$, which captures a *switch-like* process. In matrix form, we can express any *unperturbed* steady state $\overline{\boldsymbol{x}}$ of the system (4.2) as:

$$\overline{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{h}(\overline{\boldsymbol{x}};\boldsymbol{b}) \ , \tag{4.3}$$

where $\boldsymbol{h}(\boldsymbol{x};\boldsymbol{b}) := (h(x_1;\boldsymbol{b}),\ h(x_2;\boldsymbol{b}),\ldots,\ h(x_n;\boldsymbol{b}))^\top$ is a column vector.

Our ultimate goal is to identify the GRN from *steady state* observations on the gene dynamics. A common practice to this end is to introduce *perturbations* to the system through suppressing or over-expressing a selected set of genes. The corresponding steady states are then recorded for network identification in the next stage. As a practical concern, only a small number of genes (e.g., one or maybe a few) can be perturbed at a time. For simplicity, here we we consider a set of K distinct perturbation experiments, where *only one* node is perturbed per experiment. Note that the extension to simultaneous multiple nodes' perturbation is straightforward. In each experiment $k = 1, 2, ..., K$, we *fix* the state of gene k at a desired value $z_k \in \mathbb{R}$, *i.e.,* $x_k(t) = z_k$ for all t. The perturbed steady state $\boldsymbol{x}[k]$ takes the form

$$\boldsymbol{x}[k] = (\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\boldsymbol{h}(\boldsymbol{x}[k];\boldsymbol{b}) + z_k\mathbf{e}_k \ . \tag{4.4}$$

Notice that when $z_k = 0$, this corresponds to the deletion of gene k in the experiment. Moreover, the perturbation experiment described can be seen as setting a *boundary condition* on the gene dynamics (4.2).

The expression (4.4) reveals both a good news and bad news for gene dynamics identification using perturbation experiments. On the upside, Eq. (4.4) reveals that the values of all $\boldsymbol{x}[k]$ can be extracted from the experimental results when the nonlinear function $h(\cdot)$ is known or approximated, hence despite the nonlinear interactions, the equation is *linear* in the unknown $\boldsymbol{A}$. This reveals that it is possible to retrieve the unknown using simple techniques, e.g., solving least square estimation problems. On the downside, for each k, Eq. (4.4) constitutes a set of $(n-1)$-*linear* equations in the unknown parameter $\boldsymbol{A}$. Naively, we must conduct perturbation experiments for all n genes, yielding the required $n(n-1)$ equations to reconstruct the $n \times (n-1)$ off-diagonal terms of the unknown GRN. However, such comprehensive perturbation experiments are seldom available. Indeed, the GRN of most organisms comprises $n \sim 10^3$ genes, far exceeding the scale of the majority of microarray experiments, which, given the level of available resources, consists of $K \sim 10^1 - 10^2$ experiments. Hence, we shall focus on the limit where $K \ll n$. In other words, we face a similar situation as in the last chapter with only *low-rank* observations on the unknown network. Similarly, our strategy is to rely on the common assumption that $\boldsymbol{A}$ is sparse and derive a sparse optimization algorithm to extract $\boldsymbol{A}$ from the resulting underdetermined linear system (4.4).

In addition to exploiting sparsity, our analysis shows that it is also necessary to obtain certain side information about the GRN $\boldsymbol{A}$ to guarantee the identifiability of gene dynamics. The following describes a curious property pertaining to the perturbed system states that reveals the support information of $\boldsymbol{A}$ *without* solving any optimization problem. To describe the result, we denote $\boldsymbol{a}_{col,k}$ as the kth column vector of $\boldsymbol{A}$. Let $\nabla \boldsymbol{h}(\boldsymbol{x}[k]);\boldsymbol{b})$ be the diagonal matrix with the ith diagonal element being $h'(x_i[k];\boldsymbol{b})$, *i.e.,* the derivative of $h(x_i[k];\boldsymbol{b})$ with respect to $x_i[k]$ evaluated at $x_i[k]$, we have:

Proposition 4.1 *Consider the dynamics* (4.2). *Assume that the perturbation in the steady states for the kth experiment, $\overline{\boldsymbol{x}} - \boldsymbol{x}[k]$, is small and $\lambda_{max}((\boldsymbol{I} - \mathbf{e}_k \mathbf{e}_k^\top)\boldsymbol{A}\nabla \boldsymbol{h}(\boldsymbol{x}[k];\boldsymbol{b})) < 1$. The perturbation in the steady states can be approximated by:*

$$\overline{\boldsymbol{x}} - \boldsymbol{x}[k] \approx ([\overline{\boldsymbol{x}}]_k - z_k)\mathbf{e}_k + ([\overline{\boldsymbol{x}}]_k - z_k)h'(z_k)\boldsymbol{a}_{col,k} \; . \tag{4.5}$$

The proof can be found in Appendix 4.A. Proposition 4.1 implies that the perturbation introduced by the kth experiment is *limited* only to the *direct out-neighbor* of the perturbed node k. This matches the observation made in [Barzel and Barabasi(2013)], which showed that the influence of a perturbation on a node decays exponentially fast with respect to the shortest distance to the perturbed node.

4.3 Gene Dynamics Identification

This section describes the main algorithm that we propose for identifying the GRN from the perturbation data. As the first step, we focus on the case when the dynamics parameter $\boldsymbol{b}$ is known and relegate the more complicated case with unknown $\boldsymbol{b}$ to Section 4.5. The available data can be described as a response matrix consisting of $(\overline{\boldsymbol{x}}, \{\boldsymbol{x}[k]\}_{k=1}^{K})$ in which we gather the gene expression data from K perturbation experiments.

Motivated by our analysis from the last section, the algorithm consists of two stages — we first identify the partial support of the GRN of the 1-hop neighbors of the perturbed genes, then we apply a sparse regularization to the sparsest fit of the GRN to the observed perturbed steady states.

Step 1: Finding the partial support. From Proposition 4.1, the difference vector $\overline{\boldsymbol{x}} - \boldsymbol{x}[k]$ is approximately supported on $\mathrm{supp}(\boldsymbol{a}_{col,k})$, *i.e.,* the kth column of the original $\boldsymbol{A}$. As such, if we let $\delta > 0$ be a pre-defined threshold and consider the index set:

$$\mathcal{S} = \bigcup_{j=1}^{K} \left\{ (i,j) \in [n] \times [n] : \frac{[\overline{\boldsymbol{x}} - \boldsymbol{x}[j]]_i}{[\overline{\boldsymbol{x}}]_j - z_j} < \delta \right\} , \tag{4.6}$$

then the set $\mathcal{S}$ identifies the locations of *zeros or non-edges* in the first K columns of the GRN adjacency matrix $\boldsymbol{A}$. Notice that the above set can be computed without solving any optimization problem. Moreover, we define $\mathcal{S}_i$ as the restriction of $\mathcal{S}$ to the ith row of $\boldsymbol{A}$, notice that $\mathcal{S}_i \subseteq [K]$.

Step 2: Sparse recovery of GRN. We observe that (4.4) can be split in a row-by-row fashion such that:

$$(\boldsymbol{x}[k])_i = \boldsymbol{a}_i^\top \boldsymbol{h}(\boldsymbol{x}[k]; \boldsymbol{b}), \ \forall \ k \in [K] \setminus \{i\} , \tag{4.7}$$

where $\boldsymbol{a}_i$ is the ith row of $\boldsymbol{A}$. From the above, we have $\approx n$ unknowns in the variable $\boldsymbol{a}_i$ yet only $\approx K$ linear equations pertaining to $\boldsymbol{a}_i$. As $K \ll n$, we exploit the common assumption that $\boldsymbol{a}_i$ is sparse. This further suggests us to consider the following sparse optimization:

$$\min_{\hat{\boldsymbol{a}}_i} \;\; \|\hat{\boldsymbol{a}}_i\|_1 \;\; \text{s.t.} \;\; [\overline{\boldsymbol{x}}]_i = \hat{\boldsymbol{a}}_i^\top \boldsymbol{h}(\overline{\boldsymbol{x}};\boldsymbol{b}), \; (\boldsymbol{x}[k])_i = \hat{\boldsymbol{a}}_i^\top \boldsymbol{h}(\boldsymbol{x}[k];\boldsymbol{b}), \; \forall \; k \in [K] \setminus \{i\} \; , \tag{4.8a}$$

$$[\hat{\boldsymbol{a}}_i]_i = 0, \; [\hat{\boldsymbol{a}}_i]_j = 0, \; \forall \; j \in \mathcal{S}_i, \; \hat{\boldsymbol{a}}_i \in \mathbb{R}^n \; . \tag{4.8b}$$

The solution to the above problem, $\hat{\boldsymbol{a}}_i$, serves as an estimate for the ith row of $\boldsymbol{A}$.

Lastly, let us comment on the complexity of the proposed method above. The first step merely involves a thresholding operation and can be computed easily with a complexity of $\mathcal{O}(Kn)$. As for the second step, each of the proposed problem (4.8) can be converted into a linear program (LP) with (at most) $2n$ unknowns and K linear equality constraints [Foucart and Rauhut(2013)]. In general, such an LP can be solved at a worst-case complexity of $\mathcal{O}((K+2n)^{3/2}(2n)^2\log(1/\epsilon)) = \mathcal{O}(n^{3.5}\log(1/\epsilon))$ where $\epsilon > 0$ is the desired solution accuracy [Ben-Tal and Nemirovski(2001)]. Moreover, since each of the sub-problems are *decoupled* and can be solved independently, a practical way is to solve these problems *in parallel*. Our numerical results indicate that the proposed procedure can be completed in reasonable time for a relatively large network with $n \approx 1000$ genes.

4.4 Guarantees for Gene Dynamics Identification

To understand the fundamental limits of recovering the GRN with (4.8), we study the scenario when the steady states $\overline{\boldsymbol{x}}$ and $\boldsymbol{x}[k]$ are obtained *with no noise*, *i.e.,* they satisfy the equalities (4.3) and (4.4), and the dynamics parameter $\boldsymbol{b}$ is known. The challenge in the analysis is that the *undetermined* linear system (4.8a) depends on the true network $\boldsymbol{A}$ itself which is a sparse matrix, and the dynamical system is non-linear. We develop a new sparse recoverability condition that is different from [Candes and Tao(2005)]. We have:

Theorem 4.1 *Assume that (a) the set $\mathcal{S}$ is the complement of the support of $\boldsymbol{A}$; (b) the matrix $\boldsymbol{A}$ is non-negative; (c) the approximation in Proposition 4.1 is exact; (d) $\boldsymbol{h}(\overline{\boldsymbol{x}};\boldsymbol{b})$ admits an exact first order Taylor approximation at $\boldsymbol{x}[k]$. For each $i \in [n]$, if the support of the matrix $([\boldsymbol{A}]_{:,\mathcal{S}_i})^\top$ corresponds to an (α,δ)-unbalanced expander graph with left degree* *bounded in* $[d_l, d_u]$ *such that*

$$\|\boldsymbol{a}_i\|_0 \leq \frac{\alpha}{1 + (d_l/d_u)\delta}\ n,\ \forall\ i \in [n]\ , \tag{4.9}$$

$$2(d_l/d_u)\delta > \sqrt{5} - 1,\ \ \overline{x}_k - z_k \geq 0,\ \forall\ k \in [K]\ , \tag{4.10}$$

then solving (4.8) *with the additional constraint* $\hat{\boldsymbol{a}}_i \geq 0$ *yields a unique solution such that* $\boldsymbol{a}_i = \hat{\boldsymbol{a}}_i$*, where* $\boldsymbol{a}_i$ *is the ith row of* $\boldsymbol{A}$*.*

The formal definition of an expander (bipartite) graph can be found in [Gilbert and Indyk(2010)] or in Definition 3.2 of Chapter 3. The proof of Theorem 4.1 is relegated to Appendix 4.B. A curious fact about Theorem 4.1 is that the condition (4.9) depends on the graph structure of $\boldsymbol{A}$ as well as the sparsity of each row $\boldsymbol{a}_i$ of $\boldsymbol{A}$. This is due to the fact that the linear system (4.8a) depends on $\boldsymbol{A}$ itself. Regarding Theorem 4.1, we have the following comments:

- The assumptions made in the theorem above are more stringent than those for the case with opinion dynamics [cf. Theorem 3.1]. In particular, we require the partial support detection in (4.6) to be exact and the Taylor expansion to be exact. These are in general not true. However, the conditions given provide insights towards the types of the GRNs that can be identified easily.

- In contrast to our previous result for opinion dynamics [cf. Theorem 3.1], the sufficient condition of identifiability depends on the choice of perturbed genes and their corresponding local topology [cf. the expander graph assumption]. As we show in the short discussion below, a possible scenario satisfying the conditions in Theorem 4.1 requires choosing the set of perturbed genes $[K]$ such that each gene in V is regulated by a similar number ($\sim \ell$) of the genes in $[K]$.

- Eq. (4.9) requires the sparsity of *each* $\boldsymbol{a}_i$ to be uniformly bounded. This predicts that a GRN more similar to a *regular* graph will be easier to identify. However, the actual GRNs are usually endowed with non-uniform degrees, or even with a power law degree distribution. That said, the proposed method is still able to identify the sparser rows of $\boldsymbol{A}$ accurately and our numerical experiments on empirical data confirms its good performance.

Special case satisfying the perfect recovery conditions. We describe a case where the perfect recovery conditions in Theorem 4.1 can be satisfied. In particular, we begin our construction by choosing the set of perturbed genes V_p, $|V_p| = K$, such that for each $j \in V$, the jth gene is regulated by ℓ genes in the chosen perturbed genes, *i.e.,* it has ℓ in-neighbors from the set V_p. In other words, the sub-matrix $([\boldsymbol{A}]_{:,V_p})^\top \in \mathbb{R}^{K \times n}$ corresponds to the bipartite graph with a constant degree ℓ.

Now, the sub-sub-matrix $([\boldsymbol{A}]_{:,\mathcal{S}_i})^\top$ is formed by deleting a random subset of ℓ rows from $([\boldsymbol{A}]_{:,V_p})^\top$ such that $K = |\mathcal{S}_i| + \ell$. Using a similar set of arguments in Proposition 3.3 of the previous chapter, it can be shown that the support of the sub-matrix $([\boldsymbol{A}]_{:,\mathcal{S}_i})^\top$ corresponds to a random bipartite graph with bounded degree in $[\ell - 1, \ell]$ with high probability. Finally, we apply Proposition 3.4 to show that this bipartite graph is an $(\alpha, 1 - 1/(\ell - 1))$-expander with a high probability. This is done by checking

$$\ell - 1 > \max\left\{4, \frac{H(\alpha) + \beta H(\alpha/\tilde{\beta})}{\alpha \log(\tilde{\beta}/\alpha)}\right\}, \tag{4.11}$$

where $|\mathcal{S}_i| = \tilde{\beta} \cdot n$, and $\tilde{\beta} > \alpha$ (note that $|\mathcal{S}_i| < K$). Moreover, the conditions (4.9) and (4.10) imply that

$$\alpha \geq \frac{d_{\mathsf{max}}}{n} \cdot \big(1 + \delta(d_l/d_u)\big) > 1.62 \cdot \frac{d_{\mathsf{max}}}{n}, \tag{4.12}$$

where d_{max} is the maximum in-degree for the genes in the network. A sufficient condition satisfying the above requirement can be found by checking Table 3.1 while substituting the β' therein with the $\tilde{\beta}$ in the above. Now, similar to the discussion in the previous chapter, for a fixed ℓ, it can be shown that the ratio $\tilde{\beta}/\alpha$ approaches a constant as $\alpha, \beta \to 0$. Consequently, the number K of perturbation experiments required satisfies $K = \Omega(d_{\mathsf{max}})$, *i.e.,* independent of the network size, showing that $K = \Omega(d_{\mathsf{max}})$ is a sufficient condition for perfect recovery. Notice that these are only sufficient conditions for perfect recovery, as demonstrated in Section 4.6.

4.5 Robust Identification of Sparse Networks

So far, the theoretical model above assumes that the steady state expression data are measured noiselessly and the parameter $\boldsymbol{b}$ in the model response function $h(x;\boldsymbol{b})$ is known. This section proposes several practical heuristics to tackle the scenarios when $\boldsymbol{b}$ is unknown and we have noisy expression data. In particular, we consider the following noisy observation model:

$$\tilde{\overline{\boldsymbol{x}}} = \overline{\boldsymbol{x}} + \overline{\boldsymbol{\epsilon}} \quad \text{and} \quad \tilde{\boldsymbol{x}}[k] = \boldsymbol{x}[k] + \boldsymbol{\epsilon}[k] \ , \tag{4.13}$$

where the vectors $\overline{\boldsymbol{\epsilon}}$, $\boldsymbol{\epsilon}[k]$ represent additive noise which is bounded. Let $i \in [n]$, $\hat{\boldsymbol{b}}_i$ be an estimate of the parameter and define the matrix/vector:

$$\boldsymbol{y}_i := \begin{pmatrix} \tilde{x}_i[1] \\ \vdots \\ \tilde{x}_i[K] \\ \tilde{\overline{x}}_i \end{pmatrix} \quad \text{and} \quad \boldsymbol{H}_i(\hat{\boldsymbol{b}}_i) := \begin{pmatrix} \boldsymbol{h}(\tilde{\boldsymbol{x}}[1];\hat{\boldsymbol{b}}_i)^\top \\ \vdots \\ \boldsymbol{h}(\tilde{\boldsymbol{x}}[K];\hat{\boldsymbol{b}}_i)^\top \\ \boldsymbol{h}(\tilde{\overline{\boldsymbol{x}}};\hat{\boldsymbol{b}}_i)^\top \end{pmatrix} \ . \tag{4.14}$$

We assume the underlying model parameter $\boldsymbol{b}_i$ for each gene to be different for better fitting. Naturally, one would like to relax the equalities in (4.8a) and minimize the cost $\lambda\|\hat{\boldsymbol{a}}_i\|_1 + \|\boldsymbol{y}_i - \boldsymbol{H}_i(\hat{\boldsymbol{b}}_i)\hat{\boldsymbol{a}}_i\|_2$. However, we observe that the kth element of $\boldsymbol{y}_i$ is expressed as:

$$\tilde{x}_i[k] = \boldsymbol{h}(\boldsymbol{x}[k];\boldsymbol{b}_i)^\top \boldsymbol{a}_i + \epsilon_i[k] = \boldsymbol{h}(\tilde{\boldsymbol{x}}[k];\hat{\boldsymbol{b}}_i)^\top \boldsymbol{a}_i + \boldsymbol{\delta}[k]^\top \boldsymbol{a}_i + \epsilon_i[k] \ , \tag{4.15}$$

where $\boldsymbol{\delta}[k] := \boldsymbol{h}(\boldsymbol{x}[k];\boldsymbol{b}_i) - \boldsymbol{h}(\tilde{\boldsymbol{x}}[k];\hat{\boldsymbol{b}}_i)$ is an unknown vector that scales with the magnitude of $\boldsymbol{\epsilon}[k]$. The difference vector $\boldsymbol{y}_i - \boldsymbol{H}_i(\boldsymbol{b}_i)\hat{\boldsymbol{a}}_i$ is dependent on $\boldsymbol{a}_i$ and can not be modeled as an additive noise.

From (4.15), for each $i \in [n]$, we can model the vector $\boldsymbol{y}_i$ as

$$\boldsymbol{y}_i = \boldsymbol{H}_i \boldsymbol{a}_i + \boldsymbol{\Delta}_i \boldsymbol{a}_i + \boldsymbol{\epsilon} \ , \tag{4.16}$$

where $\boldsymbol{\Delta}_i$ models the error in the observed measurement matrix $\boldsymbol{H}_i$. Let $r > 0$, the uncertainty set for $\boldsymbol{\Delta}_i$ is defined such that each row vector in the matrix has a bounded norm, *i.e.,*

$$\mathcal{U}_r = \{ \boldsymbol{\Delta}_i \ : \ \|\boldsymbol{d}_k\|_2 \leq r, \ \forall \ k \in [K+1] \} \ , \tag{4.17}$$

where $\boldsymbol{d}_k$ is the kth row vector of $\boldsymbol{\Delta}_i$. To recover $\boldsymbol{a}_i$, we consider minimizing the following robust objective:

$$J(\hat{\boldsymbol{a}}_i) = \lambda \|\hat{\boldsymbol{a}}_i\|_1 + \max_{\boldsymbol{\Delta}_i \in \mathcal{U}_r} \|\boldsymbol{y}_i - \boldsymbol{H}_i \hat{\boldsymbol{a}}_i - \boldsymbol{\Delta}_i \hat{\boldsymbol{a}}_i\|_2 \ , \tag{4.18}$$

which can be upper bounded by:

$$\begin{aligned} J(\hat{\boldsymbol{a}}_i) &\leq \lambda \|\hat{\boldsymbol{a}}_i\|_1 + \|\boldsymbol{y}_i - \boldsymbol{H}_i \hat{\boldsymbol{a}}_i\|_2 + \max_{\boldsymbol{\Delta}_i \in \mathcal{U}_r} \|\boldsymbol{\Delta}_i \hat{\boldsymbol{a}}_i\|_2 \\ &= \lambda \|\hat{\boldsymbol{a}}_i\|_1 + \|\boldsymbol{y}_i - \boldsymbol{H}_i \hat{\boldsymbol{a}}_i\|_2 + r\sqrt{K+1} \cdot \|\hat{\boldsymbol{a}}_i\|_2 \ , \end{aligned} \tag{4.19}$$

where the last equality is achieved by applying Cauchy-Schwarz and setting each row of $\boldsymbol{\Delta}_i$ to $r \cdot \hat{\boldsymbol{a}}_i / \|\hat{\boldsymbol{a}}_i\|_2$. Setting $\gamma = r\sqrt{K+1}$ and minimizing the upper bound function yields the robust network recovery problem (4.20).

On the other hand, the unknown parameter $\boldsymbol{b}_i$ lies in a parameter set $\mathcal{B}$. As such, our robust identification of sparse networks (RIDS) method tackles — for each $i \in [n]$:

$$\min_{\hat{\boldsymbol{a}}_i, \hat{\boldsymbol{b}}_i} \ J(\hat{\boldsymbol{a}}_i; \hat{\boldsymbol{b}}_i) := \|\boldsymbol{y}_i - \boldsymbol{H}_i(\hat{\boldsymbol{b}}_i)\hat{\boldsymbol{a}}_i\|_2 + \rho \|\hat{\boldsymbol{a}}_i\|_1 + \gamma \|\hat{\boldsymbol{a}}_i\|_2 \tag{4.20a}$$

$$\text{s.t.} \ \ [\hat{\boldsymbol{a}}_i] = 0, \ [\hat{\boldsymbol{a}}_i]_j = 0, \ \forall \ j \in \mathcal{S}_i, \ \hat{\boldsymbol{a}}_i \in \mathbb{R}^n, \ \hat{\boldsymbol{b}}_i \in \mathcal{B} \ , \tag{4.20b}$$

where $\rho, \gamma > 0$ are fixed regularization parameter. This formulation is akin to the matrix uncertainty (MU) selector in [Rosenbaum and Tsybakov(2010)] for sparse recovery with uncertainty. Despite being robust to measurement error, the above formulation simultaneously solves for the best model parameter that fits with the expression data.

However, (4.20) is a *non-convex* problem due to the multiplicative coupling in the least square objective function. The problem cannot be solved directly using off-the-shelf packages. The RIDS method applies an alternating optimization (AO) approach to get around with the issue, *i.e.,* by running the following iterative procedure for each $i \in [n]$:

$$\boxed{\begin{aligned} &\text{for} \;\; \ell = 1, 2, 3, \ldots, L \\ &\qquad \hat{\boldsymbol{a}}_i^{\ell+1} \leftarrow \arg\min_{\hat{\boldsymbol{a}}_i} \;\; J(\hat{\boldsymbol{a}}_i; \hat{\boldsymbol{b}}_i^{\ell}) \;\; \text{s.t.} \;\; \text{(4.20b) satisfied} \; , \\ &\qquad \hat{\boldsymbol{b}}_i^{\ell+1} \leftarrow \arg\min_{\boldsymbol{b}} \;\; \|(\hat{\boldsymbol{b}}_i^{\ell} - \epsilon \cdot \nabla_{\boldsymbol{b}} J(\hat{\boldsymbol{a}}_i^{\ell+1}; \boldsymbol{b}^{\ell})) - \boldsymbol{b}\|_2 \;\; \text{s.t.} \;\; \boldsymbol{b} \in \mathcal{B} \; , \end{aligned}} \tag{4.21}$$

where $\epsilon > 0$ is a fixed step size and $\nabla_{\boldsymbol{b}} J(\hat{\boldsymbol{a}}_i^{\ell+1}; \hat{\boldsymbol{b}}_i^{\ell})$ is the gradient of the cost function. The last step is a *projected gradient* update step for $\hat{\boldsymbol{b}}_i$.

The RIDS method is summarized in Figure 4.1. In the first stage, we apply a pre-process method to de-noise the experimental data (see the subsection below); in the second stage, we tackle the robust GRN recovery problem (4.20) using the AO procedure in (4.21).

4.5.1 Handling Empirical Gene Expression Data

Empirical gene expression data are typically poorly processed as the actual experimental data is prone to noise. In this section, we describe a set of procedures, which are inspired by our analysis on the gene dynamics, for denoising the empirical gene expression data such that they can be better exploited by the proposed RIDS method.

Step 1: Subspace projection for 'denoising'. As the first step, we apply a subspace projection method as a pre-processing stage in the RIDS method. In particular, let the set of chips (or vectors of expression levels) taken under the *no perturbation* condition be

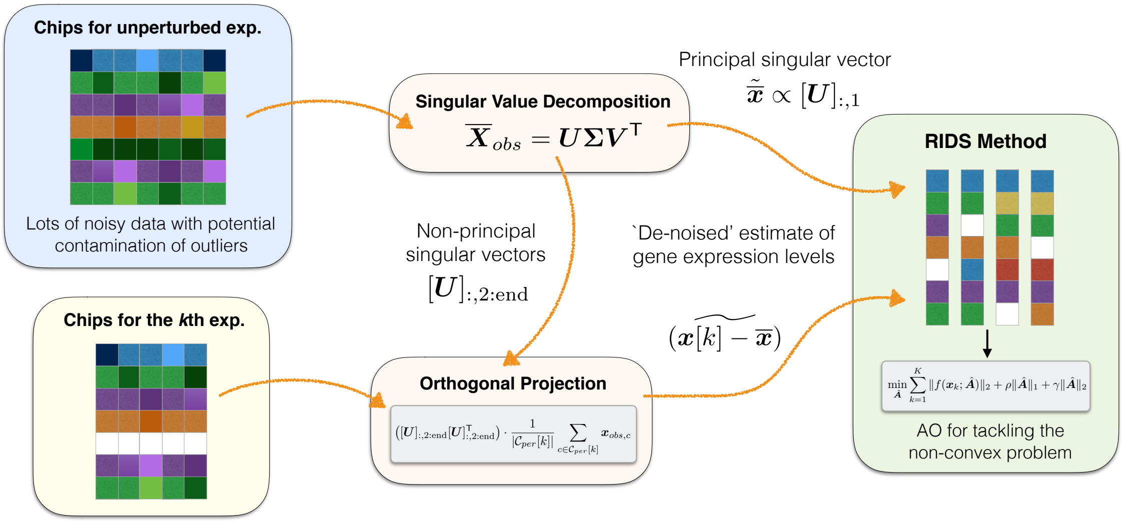
Chips for unperturbed exp.
Lots of noisy data with potential contamination of outliers
Chips for the kth exp.
Singular Value Decomposition
$\overline{\boldsymbol{X}}_{obs} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top}$
Principal singular vector
$\tilde{\overline{\boldsymbol{x}}} \propto [\boldsymbol{U}]_{:,1}$
Non-principal singular vectors
$[\boldsymbol{U}]_{:,2:\text{end}}$
`De-noised' estimate of gene expression levels
$\widetilde{(\boldsymbol{x}[k] - \overline{\boldsymbol{x}})}$
Orthogonal Projection
$([\boldsymbol{U}]_{:,2:\text{end}}[\boldsymbol{U}]_{:,2:\text{end}}^{\top}) \cdot \frac{1}{|\mathcal{C}_{per}[k]|} \sum_{c \in \mathcal{C}_{per}[k]} \boldsymbol{x}_{obs,c}$
RIDS Method
$\min_{\hat{\boldsymbol{A}}} \sum_{k=1}^{K} \|f(\boldsymbol{x}_k; \hat{\boldsymbol{A}})\|_2 + \rho\|\hat{\boldsymbol{A}}\|_1 + \gamma\|\hat{\boldsymbol{A}}\|_2$
AO for tackling the non-convex problem

Figure 4.1: Overview of the processing steps for empirical data. In the pre-processing stage, we first recover the unperturbed steady-state expression levels $\tilde{\overline{\boldsymbol{x}}}$ by analyzing the principal component of the stacked response matrix $\overline{\boldsymbol{X}}_{obs}$. Then, an orthogonal projection is applied to the perturbed steady-state expression levels to recover $\boldsymbol{x}[k] - \overline{\boldsymbol{x}}$ for each perturbation condition. This forms $(K+1)$ vectors where each of them correspond to a distinct perturbation condition (including no perturbation). Finally, we tackle the robust sparse network recovery problem (4.20) for GRN recovery via the AO procedure (4.21).

$\mathcal{C}_{nopert}$, we can write the gene expression levels from the cth chip as

$$\overline{\boldsymbol{x}}_{obs,c} = \overline{\boldsymbol{x}} + \boldsymbol{w}_c, \ \forall \ c \in \mathcal{C}_{nopert} \Longrightarrow \underbrace{\begin{pmatrix} \dots & \overline{\boldsymbol{x}}_{obs,c} & \dots \end{pmatrix}}_{:=\overline{\boldsymbol{X}}_{obs}} = \overline{\boldsymbol{x}}\boldsymbol{1}^{\top} + \boldsymbol{W}_{obs} \ , \tag{4.22}$$

where $\overline{\boldsymbol{x}}$ is the unperturbed steady state satisfying $\overline{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{h}(\overline{\boldsymbol{x}})$ (cf. Eq. (4.3)) and $\boldsymbol{w}_c$ is modeled as an additive noise. When the noise is small, we observe that the $n \times |\mathcal{C}_{nopert}|$ matrix $\overline{\boldsymbol{X}}_{obs}$ formed by stacking up $\overline{\boldsymbol{x}}_{obs,c}$ horizontally is close to rank-one. In light of this, a natural way for recovering $\overline{\boldsymbol{x}}$ is by taking the top left-singular vector of $\overline{\boldsymbol{X}}_{obs}$, *i.e.*,

$$\tilde{\overline{\boldsymbol{x}}} = \sigma_1(\overline{\boldsymbol{X}}_{obs}) \cdot \boldsymbol{u}_1 \quad \text{where} \quad \overline{\boldsymbol{X}}_{obs} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\top} \ , \tag{4.23}$$

and $\sigma_1(\overline{\boldsymbol{X}}_{obs})$ is the largest singular value of $\overline{\boldsymbol{X}}_{obs}$.

On the other hand, for the perturbed steady-states where $k \in [K]$, we can estimate the difference $\boldsymbol{x}[k] - \overline{\boldsymbol{x}}$ by the following subspace projection:

$$(\widetilde{\boldsymbol{x}[k] - \overline{\boldsymbol{x}}}) = \big([\boldsymbol{U}]_{:,2:\text{end}}[\boldsymbol{U}]_{:,2:\text{end}}^{\top}\big) \cdot \frac{1}{|\mathcal{C}_{per}[k]|} \sum_{c \in \mathcal{C}_{per}[k]} \boldsymbol{x}_{obs,c} \, , \tag{4.24}$$

where $\boldsymbol{x}_{obs,c}$ is the gene expression levels from chip c and $\mathcal{C}_{per}[k]$ is the set of chips that corresponds to the kth perturbation experiment. Naturally, we set

$$\tilde{\boldsymbol{x}}[k] = (\widetilde{\boldsymbol{x}[k] - \overline{\boldsymbol{x}}}) + \tilde{\overline{\boldsymbol{x}}} \, , \tag{4.25}$$

to be our estimate of the kth perturbed steady state.

Step 2: Normalizing the gene expression vectors. As a final step of the preprocessing, we normalize the vectors of gene expression level $\tilde{\overline{\boldsymbol{x}}}$ and $\tilde{\boldsymbol{x}}[k]$ obtained from from the preprocessing steps detailed by dividing the vectors by the constant

$$c_{norm} := \frac{1}{2} \max \Big\{ \max_{i \in [n]} \; (\tilde{\overline{x}}_i), \;\; \max_{k \in [K]} \max_{i \in [n]} \; (\tilde{x}_i[k]) \Big\} \tag{4.26}$$

such that the values of the normalized gene expression level ranges in $[0, 2]$.

4.6 Numerical Experiments

This section presents numerical results for verifying our theoretical claims on the performance of the proposed RIDS method. To emphasize on the applications to real world networks, our experiments cover both synthetic and empirical data. Notice that we use the similar performance metrics to those of Section 3.6, *i.e.,* we compare the AUROC and normalized MSE of the algorithms under various settings.

4.6.1 Synthetic Data

In silico *data.* We test the models when the GRN $G = (V, E)$ with $n = 100$. The weight matrix $\boldsymbol{A}$ has entries that are uniformly distributed in $[0, 1]$. We evaluate the steady-state

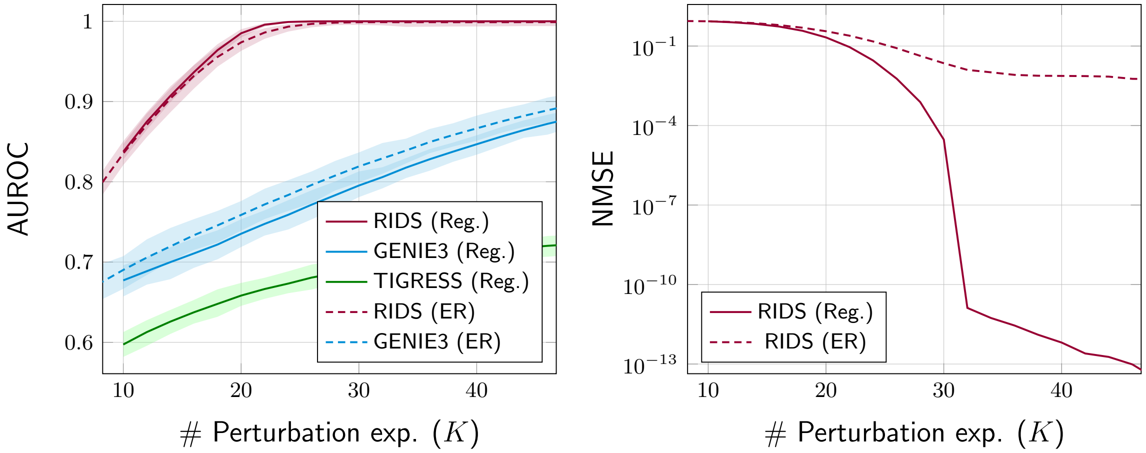

Figure 4.2: Identifying GRNs with noiseless synthetic data. The GRN has $n = 100$ nodes and is generated as a random graphs — ER graph with connectivity of 0.1 and Random regular graphs ('Reg') with constant degree $d = 10$. (Left) Area under ROC. (Right) MSE of $\hat{\boldsymbol{A}}$. The shaded area shows the 5% / 95% percentile interval for the AUROC performances.

gene expression levels subject to gene deletion using the 4th order Runge-Kutta method. We include the GENIE3 method [Huynh-Thu *et al.*(2010)] and TIGRESS method [Haury *et al.*(2012)] for benchmarks. These two were the best performing methods in the `DREAM5` challenge. All algorithms tested are implemented on MATLAB 2016a. For the zeros index set $\mathcal{S}$ in (4.6), we set $\delta = 0.02$. The model response function used is $h(x) = x^{0.5}/(1 + x^{0.5})$ and the parameters are assumed to be known.

Analysis. We notice that the TIGRESS method has encountered numerical issues for the case with ER graphs and therefore it was shown only for the random regular graphs case. Figure 4.2 compares the mean square error of the recovered $\hat{\boldsymbol{A}}$ and the area under an ROC curve (AUROC) for the recovered network G versus the number of perturbation experiments K. We assume noiseless measurements in this case and solve (4.8) to recover the network. We observe that the proposed RIDS method achieves an AUROC of ≥ 0.9 with $K \geq 14$ perturbation experiments, significantly outperforming the GENIE3 and TIGRESS methods under similar conditions. Moreover, we see that the proposed RIDS method has a better performance when the underlying graph is a regular graph. In particular, with

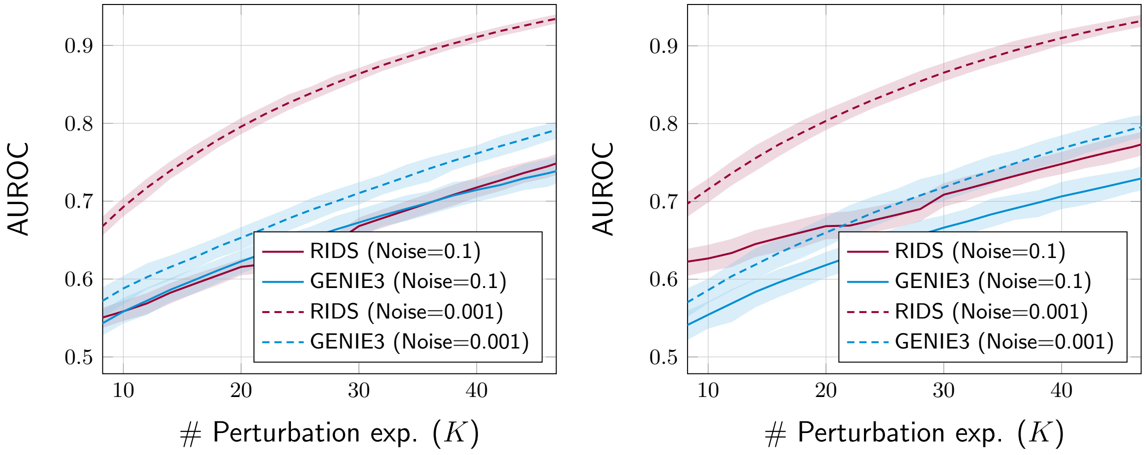

Figure 4.3: Identifying GRNs with noisy synthetic data. The GRNs have $n = 100$ nodes and are generated as random graphs. (Left) ER graphs with connectivity of 0.1. (Right) Random regular graphs with constant degree $d = 10$. The shaded area shows the 5% / 95% percentile interval for the AUROC performances.

$K \approx 32$ perturbation experiments we have perfect recovery of both the inferred links and interaction strengths, A_{ij}, for the regular graph model. Perfect recovery was also observed for the ER model for $\sim 70\%$ of the instances at $K \geq 32$, but however, as the ER graphs tend to have hubs with high degree, its average MSE performance will be affected. Nevertheless, having $K \approx 14$ experiments ($\sim 15\%$ of the total number of genes) is sufficient to yield a good GRN recovery performance. The above result corroborates with our analysis that regular graphs can be identified with less number of experiments.

Figure 4.3 considers the *noisy* measurement scenario. With reference to (4.13), the elements of $\overline{\boldsymbol{\epsilon}}, \boldsymbol{\epsilon}[k]$ are independently extracted from normal distributions $\mathcal{N}(0, 0.1)$ and $\mathcal{N}(0, 0.01)$. We apply the robust formulation (4.20) with the regularizing parameters set to $\rho = 10^{-5}, \gamma = 0.5$, $\gamma = 0.05$ for the case with $\mathcal{N}(0, 0.1)$ noise and $\mathcal{N}(0, 0.01)$ noise, respectively, to recover the network, notice that in this scenario the parameter $\boldsymbol{b}$ is known and the problem (4.20) can be solved directly. Comparing the average AUROC performance shows that the RIDS method has consistently delivered a better performance than GENIE3 and TIGRESS. However, we notice that as the noise power grows, the advantage of applying our method declines, *i.e.,* the RIDS achieves the same performance as GENIE3 when the noise power is 0.1 in the random regular graphs case.

4.6.2 Empirical Data

In vivo *data.* To test our methodology against empirical data, we reconstruct the GRN of *Escherichia coli* (*E. coli*) and *Saccharomyces cerevisiae* (*S. cerevisiae*) directly from gene perturbation experiments, using the highly curated datasets collected for the `DREAM5` Network Inference Challenge[1]. The *E. coli* (*resp. S. cerevisiae*) dataset describes 805 (*resp.* 536) vectors of expression level of $n = 4511$ (*resp.* $n = 5950$) anonymized genes under different experimental conditions; the dataset also lists a subset of $TF = 334$ (*resp.* $TF = 333$) genes that are recognized as known *transcription factors* (TFs), some of whom are decoys, namely wrongly labeled as such.

Our method relies on the expression levels in the steady state only. Therefore, we use only 326 out of 805 (*resp.* 238 out of 536) vectors of expression levels in the *E. coli* (*resp. S. cerevisiae*) dataset, *i.e.,* about $\sim 40\%$ of the data. The remaining vectors correspond to transient states, which our method is not designed to treat. Upon grouping and denoising the dataset using the pre-processing method described, we are left with $K = 56$ (*resp.* $K = 7$) vectors of expression levels for *E. coli* (*resp. S. cerevisiae*), each with a distinct perturbation conditions. From these vectors, we must reconstruct $\boldsymbol{A}$, capturing the GRN between the transcription factors and the genes, a total of 334×4511 (*resp.* 333×5950) potential links for *E. coli* (*resp. S. cerevisiae*).

We use the following model response function for the empirical data:

$$h(x; \boldsymbol{b}) = 0.75 \cdot x^{b_2}/(1 + b_1 x^{b_2}),\ \mathcal{B} = \{\boldsymbol{b} \in \mathbb{R}^2\ :\ b_1, b_2 \geq 0\}\ . \tag{4.27}$$

The above encompasses several kinetic interaction models. When $b_1 \to 0$, the interaction model tends to be that of a chemical activation process; otherwise the interaction model

[1] Available at `https://www.synapse.org/#\!Synapse:syn2787209/wiki/70351`.

has a saturating effect close to a switch-like process; b_2 controls the saturation rate of the gene interaction. We apply the RIDS method to learn simultaneously the GRN and the model parameters. We wish to remark that our system identification algorithm is the only one among the methods commonly used for gene network identification that gives insights on the type of network edge rather than only detecting the existence of an edge.

Parameters of RIDS. For the zeros index set $\mathcal{S}$ in (4.6), we set $\delta = 0.005$. The regularization parameters ρ and γ are set to 3×10^0 and 5×10^0 (*resp.* 3×10^{-1} and 5×10^{-1}) in (4.20) for *E. coli* (*resp. S. cerevisiae*) network. Meanwhile, the AO procedure (4.21) aims at tackling (4.20) with the model response function template (4.27), which has several parameters to be initialized — for each $i \in [n]$, we initialize the AO procedure by setting $b_1 = 0.5$ and $b_2 = 0.5$ and the step size ϵ is set to 1×10^{-2}. Lastly, the AO procedure terminates after $L = 10$ iterations in the numerical experiment for a better trade-off between complexity and accuracy.

Post-processing of the estimated GRN. The proposed algorithm may recover also the negative edge weights in $\hat{\boldsymbol{A}}$. However, we shall treat these negative edge weights as positive ones for the ease of benchmarking using the `DREAM5` evaluation script. In particular, we consider $|\hat{\boldsymbol{A}}|$ as our estimate of the GRN. Furthermore, we normalize the elements in $|\hat{\boldsymbol{A}}|$ by $\max_{i,j} |[\hat{\boldsymbol{A}}]_{ij}|$ such that they range in $[0, 1]$. For the AUROC/AUPR evaluation, we only count the top 100,000 (and 500,000) predicted links made by our algorithm.

Analysis. Table 4.1 shows the GRN recovery result for the truncated top *100,000* and *500,000* predicted links in the GRN, compared to the Gold Standard. For each method, we evaluate the AUROC, AUPR and the prediction score, defined similarly as in [Marbach *et al.*(2012)] by $\mathsf{Score} := (\log_{10}(p_{AUROC}) + \log_{10}(p_{AUPR}))/2$, where p_{AUROC} and p_{AUPR} are the p-values for AUROC/AUPR. The numerical experiment demonstrates that our robust GRN recovery approach is able to infer GRN from *few steady-state data*, while achieving superior performance to the state-of-the-art. In particular, for the *E. coli* network, the RIDS method is the top performer among the compared methods, beating even the community

	E. coli			*S. cerevisiae*		
Methods	AUROC	AUPR	Score	AUROC	AUPR	Score
TIGRESS [Haury *et al.*(2012)]	0.595	0.069	4.41	0.517	0.02	1.082
GENIE3 [Huynh-Thu *et al.*(2010)]	0.617	**0.093**	14.79	0.518	0.021	1.387
RankSum	0.65	0.09	24.90	**0.528**	0.022	6.236
bLARS [Singh and Vidyasagar(2016)]	N/A	N/A	5.841	N/A	N/A	**7.479**
RIDS (top 100k)	0.6808 1.69×10^{-64}	0.0504 9.9×10^{-2}	32.39	0.525 3.84×10^{-8}	**0.022** 2.3×10^{-2}	4.694
RIDS (opt. ***b****, top 100k)*	**0.6823** 3.13×10^{-66}	0.0508 8.5×10^{-2}	**33.29**	0.525 6.47×10^{-8}	0.021 2.9×10^{-2}	4.161
RIDS (no TF, top 100k)	0.6745 2.78×10^{-57}	0.0540 2.0×10^{-2}	29.12	0.524 3.70×10^{-7}	0.0221 4.8×10^{-2}	4.298
iRafNet [Petralia *et al.*(2015)]	0.641	**0.112**	29.26	N/A	N/A	N/A
GENIMS [Wu *et al.*(2016)]	0.705	0.052	48.33	0.533	0.02	8.454
RIDS (top 500k)	**0.7573** 1.04×10^{-184}	0.0574 2.7×10^{-3}	**93.28**	**0.5734** 1.5×10^{-119}	**0.0252** 2.38×10^{-7}	**62.64**

*All values/scores are calculated with the top 100k predictions. Exceptions are the iRafNet, GENIMS and RIDS (top 500k) in the last three rows, that are based on the top 200k, all, top 500k predictions, respectively.

Table 4.1: GRN recovery result on the two *in vivo* dataset. Scores for RankSum, GENIE3 and TIGRESS are taken directly from the DREAM5's leaderboard. Notice that RankSum is the community integrated prediction. The lower columns' scores are the p-value for the AUROC/AUPR metrics. The RIDS method in the 5th row uses the median optimized parameters learnt for the two networks, *i.e.,* $b_1 = 0.047, b_2 = 0.5893$ for *E. coli* and $b_1 = 0.5571, b_2 = 0.3749$ for *S. cerevisiae* and we solve (4.20) with the fixed parameters for all genes; the 6th row tackles (4.20) *without* using the list of transcription factors.

integrated prediction (RankSum) while using ∼ 60% less experimental data. Our method gives good performance even the list of TFs are not provided when tackling (4.20). Overall, we have the best prediction score for *E. coli*, and the score for *S. cerevisiae* is comparable to state-of-the-art.

The model parameters learnt using RIDS are plotted in Figure 4.4. We observe that the parameters learnt are clustered around $b_1 \sim 0.05$, $b_2 \sim 0.6$ for the *E. coli* network and $b_1 \sim 0.4$, $b_2 \sim 0.55$ for the *S. cerevisiae* network. This suggests that the interaction model for the former is close to a chemical activation process, while the latter is close to being

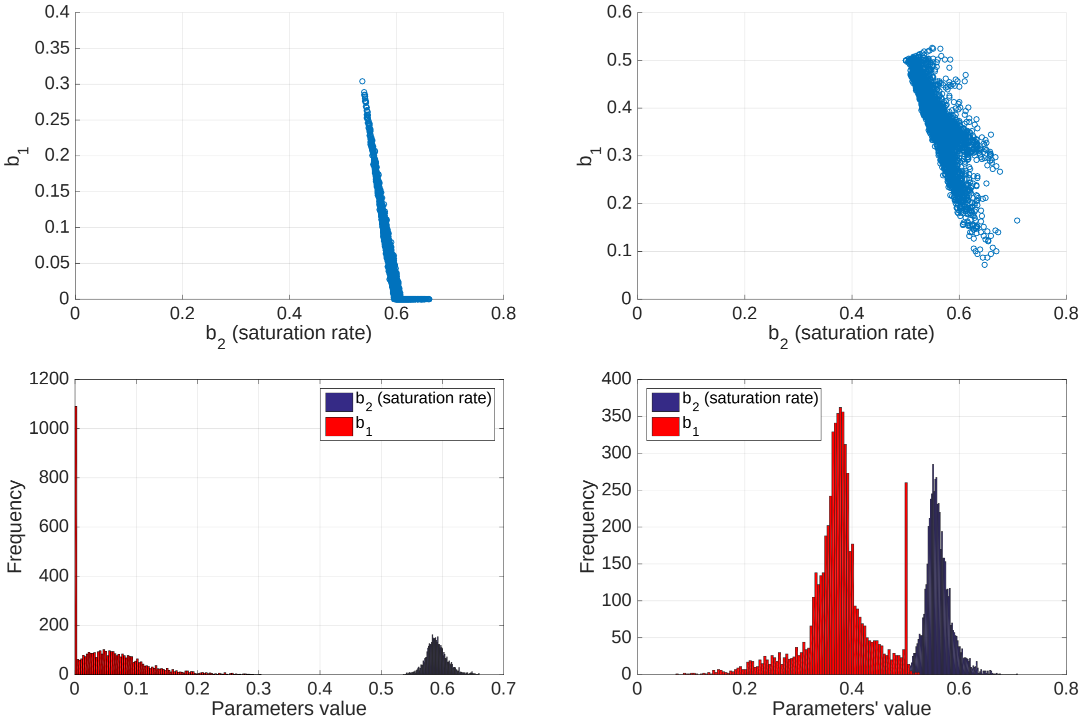

Figure 4.4: Model parameters learnt for different gene dynamics. (Left) *E. coli* network. (Right) *S. cerevisiae* network. The top figures show the scatter plot of the parameters. The bottom figures show the histograms of parameters learnt, where the blue patch is the saturation rate parameter b_2 and the red patch is parameter b_1 which controls if the interaction model is more of the chemical activation type or the switching type, cf. (4.27).

a combination. The parameters learnt are also similar across different genes, which seems to indiciate that the parameter space may be reduced by using a single set of parameters for all genes. This is corroborated by the AUROC/AUPR performance obtained when the RIDS method is re-applied with the same model parameter for all genes in Table 4.1. From the table, we note that the *S. cerevisiae* shows a higher variance for the b_1, b_2 values learnt for each gene, which is due to the fact that the in vivo data available for this network is scarcer than the one for *E. coli*.

4.7 Chapter Summary

This work proposes the RIDS method for GRN identification using a limited number of perturbation experiments. The RIDS method is developed through modeling the gene expression data as the outcome of perturbing a nonlinear dynamic system. To improve robustness, the method first applies a subspace projection method for denoising the gene expression data, then a sparse and robust estimator is applied to recover the network. The model parameters of the dynamic system will also be inferred simultaneously. Our theoretical analysis, conducted under the assumption that the dynamic's parameters are known, shows that it is possible to recover the GRN even when there are only a few sets of perturbation experiment data available. This is in contrary to the common belief that it requires a large number of experiments to apply similar methods. Moreover, our experiments on empirical data shows that the RIDS method compares favorably to the state-of-the-art methods while requiring $\sim 60\%$ less data. The RIDS method paves the way to study the dynamics of gene interactions by having the ability to infer the model parameters, which is important as studied by [Ronen *et al.*(2002)]. For example, our preliminary result suggests that the genes in the *E. coli* and *S. cerevisiae* networks tend to have a different interaction model with its neighboring genes.

Appendix

4.A Proof of Proposition 4.1

Define

$$\overline{\boldsymbol{x}} - \boldsymbol{x}[k] = ([\overline{\boldsymbol{x}}]_k - z_k)\mathbf{e}_k + \boldsymbol{\epsilon}, \tag{4.28}$$

such that the kth component of $\boldsymbol{\epsilon}$ is zero. Our goal is to find $\boldsymbol{\epsilon}$. We have:

$$\overline{\boldsymbol{x}} - \boldsymbol{x}[k] = \boldsymbol{A}(\boldsymbol{h}(\overline{\boldsymbol{x}}) - \boldsymbol{h}(\boldsymbol{x}[k])) + \mathbf{e}_k\mathbf{e}_k^\top \boldsymbol{A}\boldsymbol{h}(\boldsymbol{x}[k]) + z_k\mathbf{e}_k. \tag{4.29}$$

Notice that $\mathbf{e}_k$ is in the null space of $(\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)$. Left-multiplying with $(\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)$ to both sides of the equation yields:

$$\begin{aligned} \boldsymbol{\epsilon} &= (\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}(\boldsymbol{h}(\overline{\boldsymbol{x}}) - \boldsymbol{h}(\boldsymbol{x}[k])) \\ &\approx (\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}[k])(([\overline{\boldsymbol{x}}]_k - z_k)\mathbf{e}_k + \boldsymbol{\epsilon}) \end{aligned} \tag{4.30}$$

where we have taken Taylor's expansion for $\boldsymbol{h}(\overline{\boldsymbol{x}})$ centered at $\boldsymbol{x}[k]$ and assumed that the approximation is accurate when the perturbation $\overline{\boldsymbol{x}} - \boldsymbol{x}[k]$ is small. This gives

$$\boldsymbol{\epsilon} \approx (\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}[k])\boldsymbol{\epsilon} + ([\overline{\boldsymbol{x}}]_k - z_k)(\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}[k])\mathbf{e}_k \tag{4.31}$$

Moreover, we notice that $\nabla\boldsymbol{h}(\boldsymbol{x}[k])$ is a diagonal matrix with $h'(z_k)$ on its kth entry. Therefore the latter term can be simplified as

$$\begin{aligned} ([\overline{\boldsymbol{x}}]_k - z_k)(\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}')\mathbf{e}_k &= ([\overline{\boldsymbol{x}}]_k - z_k)h'(z_k)(\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{a}_{col,k} \\ &= ([\overline{\boldsymbol{x}}]_k - z_k)h'(z_k)\boldsymbol{a}_{col,k}\ , \end{aligned} \tag{4.32}$$

where the last equality is due to the fact that the kth element of $\boldsymbol{a}_{col,k}$ is zero.

Finally, we observe that

$$\begin{aligned}\boldsymbol{\epsilon} &= \big(\boldsymbol{I} - (\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}[k])\big)^{-1} \cdot ([\overline{\boldsymbol{x}}]_k - z_k)h'(z_k)\boldsymbol{a}_{col,k} \\ &= \big(\boldsymbol{I} + (\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}') \\ &\qquad\qquad + ((\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}'))^2 + \cdots\big) \cdot ([\overline{\boldsymbol{x}}]_k - z_k)h'(z_k)\boldsymbol{a}_{col,k} \\ &\approx ([\overline{\boldsymbol{x}}]_k - z_k)h'(z_k)\boldsymbol{a}_{col,k}\ ,\end{aligned} \tag{4.33}$$

where the second equality is due to Taylor's expansion. Notice that the series expansion holds when $\lambda_{max}((\boldsymbol{I} - \mathbf{e}_k\mathbf{e}_k^\top)\boldsymbol{A}\nabla\boldsymbol{h}(\boldsymbol{x}')) < 1$.

4.B Proof of Theorem 4.1

Using the assumptions stated in the theorem, we first reduce the linear system into a simple form that involves an underdetermined system with a *sparse* sensing matrix. The sensing matrix is then found to have the same structure/support as the bipartite graph formed by the edges *from* the perturbed nodes *to* all other nodes. Finally, the desired perfect recovery condition is given as a consequence of the expander property of this bipartite graph.

We are ready to begin our proof. Let $\boldsymbol{a}_i$ be the ith row of $\boldsymbol{A}$ and define a set of vectors $\{\boldsymbol{y}_i\}_{i=1}^n$ such that $[\boldsymbol{y}_i]_k := \overline{x}_i - x_i[k]$. We observe that $\boldsymbol{y}_i$ is a collection of the data points that depend on $\boldsymbol{a}_i$. For simplicity, let us focus on the case when $i \notin [K]$,

$$[\boldsymbol{y}_i]_k = \boldsymbol{a}_i^\top(\boldsymbol{h}(\overline{\boldsymbol{x}}) - \boldsymbol{h}(\boldsymbol{x}[k])) \approx (\overline{x}_k - z_k) \cdot \boldsymbol{a}_i^\top \nabla\boldsymbol{h}(\overline{\boldsymbol{x}})\big(h'(z_k)\boldsymbol{a}_{col,k} + \mathbf{e}_k\big)\ , \tag{4.34}$$

where we have applied Proposition 4.1 and the first order Taylor approximation on $\boldsymbol{h}(\overline{\boldsymbol{x}}) - \boldsymbol{h}(\boldsymbol{x}[k])$ to yield the result above. After some manipulations and applying the assumptions stated in the theorem, we can express the equation above as

$$\boldsymbol{y}_i = \begin{pmatrix} \boldsymbol{\Lambda} & \boldsymbol{0}_{K\times(n-K)} \end{pmatrix} \boldsymbol{a}_i + \underbrace{\begin{pmatrix} (\overline{x}_1 - z_1)h'(z_1) \cdot \nabla \boldsymbol{h}(\overline{\boldsymbol{x}}) \boldsymbol{a}_{col,1}^\top \\ \vdots \\ (\overline{x}_K - z_K)h'(z_K) \cdot \nabla \boldsymbol{h}(\overline{\boldsymbol{x}}) \boldsymbol{a}_{col,K}^\top \end{pmatrix}}_{:=\tilde{\boldsymbol{E}}_i} \boldsymbol{a}_i \ , \tag{4.35}$$

where $\boldsymbol{\Lambda}$ is an $K \times K$ diagonal matrix with the kth element being $[\boldsymbol{\Lambda}]_{kk} = \overline{x}_k - z_k$. The challenge is that as the non-zero elements of $\boldsymbol{\Lambda}$ has a larger magnitude than the matrix $\tilde{\boldsymbol{E}}_i$ in the latter matrix-vector product, the overall sensing matrix $(\boldsymbol{\Lambda}\ \boldsymbol{0}) + \tilde{\boldsymbol{E}}_i$ is dominated by the diagonal matrix component. That is, $(\boldsymbol{\Lambda}\ \boldsymbol{0}) + \tilde{\boldsymbol{E}}_i \approx (\boldsymbol{\Lambda}\ \boldsymbol{0})$. However, this implies that the elements of $\boldsymbol{a}_i$ over the coordinates $[n] \setminus [K]$ cannot be recovered from $\boldsymbol{y}_i$ as the rows of the sensing matrix supported only on $[K]$. It is thus necessary to exploit extra information to recover $\boldsymbol{a}_i$.

In light of this, we note that $[\boldsymbol{a}_i]_j$ is zero for all j in $\mathcal{S}_i$. The first matrix-vector product is thus an $K - |\mathcal{S}_i|$-sparse vector which is supported on the set $\mathcal{S}_i^c = [K] - \mathcal{S}_i$. As we are interested in studying a sufficient condition for perfect recovery, we see that (4.35) implies the following linear system with the rows corresponding to $\mathcal{S}_i^c$ removed,

$$\big[\boldsymbol{y}_i\big]_{\mathcal{S}_i} = \big[\tilde{\boldsymbol{E}}_i\big]_{\mathcal{S}_i,:} \boldsymbol{a}_i \ . \tag{4.36}$$

Compared to the original model (4.35), we can suppress the dominating diagonal component in $\tilde{\boldsymbol{E}}_i$ using the support knowledge on $\boldsymbol{a}_i$.

The remaining task is to verify if the reduced sensing matrix $\big[\tilde{\boldsymbol{E}}_i\big]_{\mathcal{S}_i,:}$ is a good sensing matrix. Notice that *dense* and random matrices are known to exhibit good properties in which one only requires $|\mathcal{S}_i| \geq 2\|\boldsymbol{a}_i\|_0 \cdot \log n$ to achieve perfect recovery. On the other hand, $\big[\tilde{\boldsymbol{E}}_i\big]_{\mathcal{S}_i,:}$ is a *sparse* matrix whose support depends on the out-neighbors of the nodes in $\mathcal{S}_i$. In particular, we have $\operatorname{supp}(\big[\tilde{\boldsymbol{E}}_i\big]_{\mathcal{S}_i,:}) = \operatorname{supp}(\boldsymbol{A}_{:,\mathcal{S}_i}^\top)$. An interesting observation is that the support of the sensing matrix depends on the support of $\boldsymbol{A}$ itself or the network that we wish to recover.

As it turns out, the perfect recovery condition for $\boldsymbol{a}_i$ boils down to studying conditions on the *support* of $\big[\tilde{\boldsymbol{E}}_i\big]_{\mathcal{S}_i,:}$. In particular, let $G_{bi}(A,B)$ with $|A| = n$ and $|B| = |\mathcal{S}_i| \leq K$ be the bi-partite graph representation of $[\boldsymbol{E}_i]_{\mathcal{S}_i,:}$ such that the adjacency matrix of G_{bi}, *i.e.,* $\boldsymbol{A}_{bi} \in \mathbb{R}^{|\mathcal{S}_i| \times n}$, has the same support as $[\boldsymbol{E}_i]_{\mathcal{S}_i,:}$.

Theorem 4.2 *If G_{bi} is an (α,δ)-unbalanced expander graph with left degree bounded in $[d_l, d_u]$ such that $2(d_l/d_u)\cdot\delta > \sqrt{5}-1$ and $k \leq \frac{\alpha}{1+\rho\delta}n$, and $\tilde{\boldsymbol{E}}_i$ or $-\tilde{\boldsymbol{E}}_i$ is non-negative, then the set*

$$\mathcal{C} = \{\hat{\boldsymbol{a}}_i \ : \ \big[\tilde{\boldsymbol{E}}_i\big]_{\mathcal{S}_i,:}(\hat{\boldsymbol{a}}_i - \boldsymbol{a}_i) = \boldsymbol{0}\} \tag{4.37}$$

is a singleton for all k-sparse vector $\boldsymbol{a}_i$.

Proof: To prove Theorem 4.2, the following lemma will be instrumental. Denote $\text{Null}(\boldsymbol{E})$ as the null space of a matrix $\boldsymbol{E}$, *i.e.,* $\text{Null}(\boldsymbol{E}) := \{\boldsymbol{y} \ : \ \boldsymbol{E}\boldsymbol{y} = \boldsymbol{0}\}$.

Lemma 4.1 *If (i) the vector $\overline{\boldsymbol{x}}$ is k-sparse, (ii) the matrix $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ satisfies $\boldsymbol{0} \neq \boldsymbol{w} \in \text{Null}(\boldsymbol{E})$, $|S_-(\boldsymbol{w})| \geq k+1$ where $S_-(\boldsymbol{w}) = \{i \in [n] : w_i < 0\}$, then the set $\mathcal{C} = \{\boldsymbol{x} : \boldsymbol{E}(\boldsymbol{x} - \overline{\boldsymbol{x}}) = \boldsymbol{0},\ \boldsymbol{x} \geq \boldsymbol{0}\}$ is a singleton.*

Proof: Suppose $|\mathcal{C}| > 1$ such that there exists $\tilde{\boldsymbol{x}} \in \mathcal{C}$, $\tilde{\boldsymbol{x}} \neq \overline{\boldsymbol{x}}$. It is straightforward to see that $\tilde{\boldsymbol{x}} = \overline{\boldsymbol{x}} + \boldsymbol{w}$, where $\boldsymbol{w} \in \text{Null}(\boldsymbol{E})$. The assumption implies that $|S_-(\boldsymbol{w})| \geq k+1$ and $\tilde{\boldsymbol{x}} = \overline{\boldsymbol{x}} + \boldsymbol{w} \not\geq \boldsymbol{0}$ as $\overline{\boldsymbol{x}}$ is k-sparse. This contradicts $\tilde{\boldsymbol{x}} \in \mathcal{C}$. **Q.E.D.**

The next step is to apply a generalization of [Wang *et al.*(2011a), Theorem 4]:

Lemma 4.2 *Let $n > m$ and $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ be a non-negative matrix that has the same support as the adjacency matrix of an (α,δ)-unbalanced bipartite expander graph with left degrees bounded in $[d_l, d_u]$ and $\rho = d_l/d_u$. If $\rho\delta > (\sqrt{5}-1)/2$, then for all k-sparse vector $\overline{\boldsymbol{x}}$, the set*

$$\mathcal{C} = \{\boldsymbol{x} : \boldsymbol{E}(\boldsymbol{x} - \overline{\boldsymbol{x}}) = \boldsymbol{0},\ \boldsymbol{x} \geq \boldsymbol{0}\}, \tag{4.38}$$

is a singleton if $k \leq \frac{\alpha}{1+\rho\delta}n$.

Proof: Using Lemma 4.1, it suffices to prove that any $\boldsymbol{w} \in \text{Null}(\boldsymbol{E})$ with $S_-(\boldsymbol{w}) \subseteq A$, we have $|S_-(\boldsymbol{w})| \geq k+1$. We shall proceed by contradiction. Suppose that there exists $\boldsymbol{w} \in \text{Null}(\boldsymbol{E})$ such that $|S_-(\boldsymbol{w})| \leq k$. Since $|S_-(\boldsymbol{w})| \leq k \leq \alpha n$, the expander property implies:

$$\delta d_l \cdot |S_-(\boldsymbol{w})| \leq \delta |E(S_-(\boldsymbol{w}), B)| \leq |N(S_-(\boldsymbol{w}))|. \tag{4.39}$$

The right hand side can be further upper bounded as:

$$|N(S_-(\boldsymbol{w}))| \leq |E(S_-(\boldsymbol{w}), B)| \leq d_u \cdot |S_-(\boldsymbol{w})| \tag{4.40}$$

Moreover, we know that $N(S_-(\boldsymbol{w})) = N(S_+(\boldsymbol{w})) = N(S_-(\boldsymbol{w}) \cup S_+(\boldsymbol{w}))$. This is because $\boldsymbol{Ew} = \boldsymbol{0}$ and $\boldsymbol{A}$ is non-negative, thus the neighborhood sets must coincide to enforce nullity, otherwise this will result in $\boldsymbol{Ew} \neq \boldsymbol{0}$. Note that in [Wang *et al.*(2011a)], the proof is achieved by assuming that $\boldsymbol{E}$ is the binary adjacency matrix. We extend the same argument to the case when $\boldsymbol{E}$ is non-negative. From the above and applying the inequality (4.39) on $N(S_+(\boldsymbol{w})) = N(S_-(\boldsymbol{w}))$, we have:

$$|S_+(\boldsymbol{w})| \geq |N(S_+(\boldsymbol{w})|/d_u \geq (d_l/d_u)\delta |S_-(\boldsymbol{w})| \tag{4.41}$$

As $S_-(\boldsymbol{w})$ and $S_+(\boldsymbol{w})$ are disjoint, we have $|S_+(\boldsymbol{w}) \cup |S_-(\boldsymbol{w})| \geq (1+\rho\delta)|S_-(\boldsymbol{w})|$. As such, we choose an arbitrary subset $\tilde{S} \subseteq S_+(\boldsymbol{w}) \cup S_-(\boldsymbol{w})$ such that $|\tilde{S}| = (1+\rho\delta)|S_-(\boldsymbol{w})|$. Notice that $|\tilde{S}| \leq (1+\rho\delta)|S_-(\boldsymbol{w})| \leq \alpha n$. Using the expander property again gives:

$$|N(\tilde{S})| \geq \delta |E(\tilde{S}, B)| \geq d_u \delta (1+\rho\delta)|S_-(\boldsymbol{w})| > d_u |S_-(\boldsymbol{w})| \tag{4.42}$$

As $\rho\delta > (\sqrt{5}-1)/2$, the last inequality is valid as $\rho\delta(1+\rho\delta) > 1$.

Finally, we reach a contradiction as

$$|N(\tilde{S})| \leq |N(S_-(\boldsymbol{w}) \cup S_+(\boldsymbol{w}))| = |N(S_-(\mathbf{w})| \leq d_u|S_-(\boldsymbol{w})|, \tag{4.43}$$

leading to $d_u|S_-(\boldsymbol{w})| > d_u|S_-(\boldsymbol{w})|$. The lemma is thus proven. **Q.E.D.**

Finally, by identifying that G_{bi} satisfies the conditions in Lemma 4.2, our desirable results in Theorem 4.2 can be obtained.

Consequently, we observe that all the conditions in Theorem 4.2 hold, therefore solving (4.8) yields $\hat{\boldsymbol{A}} = \boldsymbol{A}$, *i.e.,* we achieve perfect recovery of the network.

— PART II —

Algorithms on Networks

5 Consensus-based Projection-free Optimization

This chapter studies optimization algorithms that run on networks. In contrast to the previous part on modeling and identification of network dynamics, here we *design* the dynamics on the networks. The latter leads to solution of optimization problems relevant applications in signal estimation or machine learning.

5.1 Context and Background

In this chapter, our focus is to devise algorithms to tackle the optimization problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \; \frac{1}{N} \sum_{i=1}^{N} f_i(\boldsymbol{\theta}) \quad \text{s.t.} \quad \boldsymbol{\theta} \in \mathcal{C} \; , \tag{5.1}$$

on a network of N agents. We shall work with a smooth optimization setting where

- the function $f_i(\boldsymbol{\theta})$ is continuously differentiable (possibly non-convex), it is held by the ith agent.
- the set $\mathcal{C} \subseteq \mathbb{R}^d$ is closed, bounded and convex.

A common instance of (5.1) is the empirical risk minimization (ERM) problem, where the private risk function $f_i(\boldsymbol{\theta})$ models the loss of $\boldsymbol{\theta}$ incurred over the private data held by agent i, e.g.,

$$f_i(\boldsymbol{\theta}) = \frac{1}{|\Omega_i|} \sum_{k \in \Omega_i} \ell_i(\boldsymbol{\theta}, \boldsymbol{y}_{i,k}) \; , \tag{5.2}$$

where $\ell_i(\boldsymbol{\theta}, \boldsymbol{y}_{i,k})$ quantifies the mismatch between a statistical model parameterized by $\boldsymbol{\theta}$ and the kth data entry, $\boldsymbol{y}_{i,k}$, held by agent i. In this instance, $\mathcal{C}$ corresponds to a regularization constraint imposed on $\boldsymbol{\theta}$ that promotes desirable properties such as sparsity or a low-rank, which capture prior knowledge about the solutions that can help overcome the curse of dimensionality in searching for a solution in $\mathbb{R}^d$. Problem (5.1) also covers a number of applications in control theory and signal processing, including system identification [Liu and Vandenberghe(2010)], matrix completion [Candès and Recht(2009)] and sparse learning [Ravazzi *et al.*(2016), Patterson *et al.*(2014)]. While in most instances the inclusion of high-dimensional constraint is a fundamental ingredient to attain good estimation performance, the curse of dimensionality returns to haunt us due to the significant computational cost added when enforcing such constraints through a *projection* step, aimed at ensuring feasible iterates. As reviewed in Section 1.4, most of the prior work on decentralized optimization algorithms for constrained optimizations are projection-based, and may therefore suffer from the high computational complexity.

This chapter proposes a new decentralized *projection-free* optimization algorithm, called the decentralized Frank-Wolfe (DeFW) algorithm, which addresses the computational issues above with high-dimensional constraints. Specifically, we derive the DeFW algorithm through a careful combination of the average consensus protocol with the efficient Frank-Wolfe (FW) algorithm. Note that the latter has been applied successfully to a number of high dimensional problems in machine learning. In fact, the FW algorithm replaces the costly projection step in projection based algorithms with a constrained linear optimization, which often admits a computationally efficient solution. Importantly, we provide *convergence rates* of the DeFW algorithm for both convex and non-convex instances of (5.1).

In the rest of this chapter, after introducing the notations and necessary mathematical concepts, Section 5.2 reviews on the classical decentralized projected gradient method [Ram *et al.*(2012)] and presents the state-of-the-art convergence (rate) results. Section 5.3 gives a brief introduction to FW algorithm and develops the decentralized FW (DeFW) algorithm from the former. The convergence guarantees of DeFW algorithm will then be summarized in Section 5.4. We then present the application examples in Section 5.5 and a faster DeFW algorithm for low-rank regression in Section 5.6. Finally, we conclude with numerical experiments in Section 5.7.

Notations. We use the notations introduced in Section 2.1 on networks with the additional conditions below. In particular, the network's graph is *undirected* with N nodes, and the weighted adjacency matrix is non-negative doubly stochastic, such that $\boldsymbol{A}^\top \mathbf{1} = \boldsymbol{A}\mathbf{1} = \mathbf{1}$. The second largest singular value, $\sigma_2(\boldsymbol{A})$, is strictly less than one, implying that the graph is connected. We also define a *communication round* as the network's nodes sharing a message through the network edges once. We focus on the *static* network setting where $\boldsymbol{A}$ is time invariant.

5.2 Review: Decentralized Projected Gradient (DPG) Algorithm

The DPG algorithm emulates the centralized projected gradient descent (PG) [Bertsekas(1999)]. In particular, let $t \in \mathbb{N}$ be the iteration number, the projection can be described as:

$$\boldsymbol{\theta}_{t+1} = \mathcal{P}_{\mathcal{C}}(\boldsymbol{\theta}_t - \gamma_t \nabla F(\boldsymbol{\theta}_t)) \ , \tag{5.3}$$

where $\gamma_t \in (0, 1]$ is a step size and $\mathcal{P}_{\mathcal{C}}(\cdot)$ is the projection operator onto $\mathcal{C}$:

$$\mathcal{P}_{\mathcal{C}}(\boldsymbol{x}) := \arg\min_{\boldsymbol{\theta} \in \mathcal{C}} \ \|\boldsymbol{\theta} - \boldsymbol{x}\|_2^2 \ . \tag{5.4}$$

To mimic the centralized PG algorithm in the decentralized setting, the agents need to retrieve information about the global gradient $\nabla F(\boldsymbol{\theta}_t)$. The DPG algorithm achieves this using the following recursions — let $\boldsymbol{\theta}_t^i$ be the local iterate held by agent i at iteration t,

$$\textit{(Consensus step)} \quad \bar{\boldsymbol{\theta}}_t^i = \textstyle\sum_{j=1}^N A_{ij} \boldsymbol{\theta}_t^j \ , \tag{5.5a}$$

$$\textit{(PG step)} \quad \boldsymbol{\theta}_{t+1}^i = \mathcal{P}_{\mathcal{C}}\Big(\bar{\boldsymbol{\theta}}_t^i - \gamma_t \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\Big) \ , \tag{5.5b}$$

where $\bar{\boldsymbol{\theta}}_t^i$ is an auxiliary variable that holds a local approximation of the global average parameter $(1/N) \sum_{j=1}^N \boldsymbol{\theta}_t^j$. The consensus step (5.5a) is similar to the average consensus protocol in [Tsitsiklis(1984)] while the PG step (5.5b) is analogous to the centralized PG algorithm (5.3), with the exception that the global gradient $\nabla F(\boldsymbol{\theta}_t)$ is replaced by the local gradient function $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$, evaluated at the approximate global iterate. Despite using the local gradient vector in lieu of the global one, the DPG algorithm achieves convergence since the evaluation of $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$ incorporate information about the local functions held by the other agents that propagates through the mixing step. More specifically, the algorithm exhibits sub-linear convergence for convex problems with a diminishing step size γ_t.

As seen, the iteration steps of the DPG algorithm are conceptually simple to implement. However, for high-dimensional problems, the projection operation (5.4) can be computationally prohibitive, even when a closed form solution is available for its update. For example, when $\mathcal{C}$ is a trace norm ball for matrices of dimension $m_1 \times m_2$, *i.e.,*

$$\mathcal{C} = \{\boldsymbol{\theta} \in \mathbb{R}^{m_1 \times m_2} \ : \ \|\boldsymbol{\theta}\|_{\sigma,1} \leq R\} \ , \tag{5.6}$$

the projection operator admits a closed form solution as:

$$\mathcal{P}_{\mathcal{C}}(\boldsymbol{X}) = \boldsymbol{U} \max\{\mathbf{0}, \boldsymbol{\Sigma} - \boldsymbol{\Lambda}^\star\} \boldsymbol{V}^\top, \quad \text{where} \quad \boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \ , \tag{5.7}$$

for some diagonal $\boldsymbol{\Lambda}^\star$ such that $\|\mathcal{P}_{\mathcal{C}}(\boldsymbol{X})\|_{\sigma,1} \leq R$. Clearly, the projection step amounts to computing a full singular value decomposition (SVD) of the operand. The associated complexity of such step grows as $\mathcal{O}(\max\{m_1 m_2^2, m_2 m_1^2\})$ is a cost endured by all the N agents in all iterations. This is highly undesirable for big-data applications where $m_1, m_2 \gg 0$. The DPG has served as a prototype algorithm for a number of sophisticated decentralized optimization algorithm, e.g., [Jakovetic *et al.*(2014), Shi *et al.*(2015), Nedić *et al.*(2016), Qu and Li(2016)]. These algorithms require fewer iterations to convergence, but are equally burdened by the high complexity of the projection step.

Convergence Analysis. The convergence properties of the DPG algorithm are known for a general setting with time varying mixing matrix (*i.e.,* the matrix $\boldsymbol{A}[t]$ may change at every iteration) [Ram *et al.*(2012)]. Specifically, it has been established in [Ram *et al.*(2012)] that the algorithm converges almost surely when the step size is chosen such that $\sum_{t=1}^{\infty} \gamma_t = \infty$ and $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$ (and the time varying network is connected in an ergodic sense). However, the convergence rate of the DPG algorithm has not been studied in [Ram *et al.*(2012)].

Here, we describe the convergence rate analysis conducted in [Chen(2012)] for convex problems, whose result can be summarized as follows:

Theorem 5.1 *[Chen(2012)] Consider Problem* (5.1) *and suppose that each of f_i is convex and L-smooth. If we apply the DPG algorithm to solve* (5.46) *and choose the step size as $\gamma_t \leq 1/L$, then it holds for all $T \geq 1$ that:*

$$\min_{1 \leq t \leq T} F(\boldsymbol{\theta}_t^i) - \big(\min_{\boldsymbol{\theta} \in \mathcal{C}} F(\boldsymbol{\theta})\big) \leq \frac{D_1 + D_2 \sum_{t=1}^T \gamma_t^2}{\sum_{t=1}^T \gamma_t} \ , \tag{5.8}$$

where D_1, D_2 are some finite constants that depend on $\sigma_2(\boldsymbol{A})$. If we set $\gamma_t = C/\sqrt{t}$ for some $C < \infty$, then $\min_{1 \leq t \leq T} F(\boldsymbol{\theta}_t^i) - (\min_{\boldsymbol{\theta} \in \mathcal{C}} F(\boldsymbol{\theta})) = \mathcal{O}(\log T/\sqrt{T})$. Moreover, the algorithm attains consensus, that is $\lim_{t\to\infty} \|\boldsymbol{\theta}_t^i - (1/N)\sum_{j=1}^N \boldsymbol{\theta}_t^j\| = 0 \ \ \forall \ \ i$.

Theorem 5.1 proves that, in terms of the difference between objective values at iteration T and at an optimal solution (a.k.a. the primal optimality gap), the DPG algorithm converges sublinearly at a rate of $\mathcal{O}(\log T/\sqrt{T})$. The proof of Theorem 5.1 proceeds first in showing that the algorithm attains consensus asymptotically, and then in bounding the optimality gap by the descent lemma in [Bertsekas(1999)] (as f_i is assumed to be a smooth function).

If we relax the assumption that the objective function of Problem (5.1) is convex, little is known about the convergence (rate) of the DPG algorithm. Recent work [Tatarenko and Touri(2017)] has shown that a decentralized gradient descent method applied to the unconstrained version of (5.1) converges at a sublinear rate for non-convex problems, *i.e.*, $\|\boldsymbol{\theta}_t^i - \bar{\boldsymbol{\theta}}\| = \mathcal{O}(1/\sqrt{t})$, where $\bar{\boldsymbol{\theta}}$ is a stationary point to (5.1), yet the algorithm considered therein is different from the DPG algorithm in (5.46). The state of the art in understanding the convergence of the DPG applied to non-convex problem can be found in [Bianchi and Jakubowicz(2013)] and is quoted below:

Theorem 5.2 *[Bianchi and Jakubowicz(2013)] Consider Problem* (5.1) *and the DPG algorithm* (5.46). *Suppose that each of f_i is L-smooth. If we choose the step size such that*

$\sum_{t=1}^{\infty} \gamma_t = \infty$ *and* $\sum_{t=1}^{\infty} \gamma_t^2 < \infty$*, then the sequence* $\{\boldsymbol{\theta}_t^i\}_{t\geq 1}$ *satisfies:*

1. *(Consensus)* $\lim_{t\to\infty} \|\boldsymbol{\theta}_t^i - (1/N)\sum_{j=1}^N \boldsymbol{\theta}_t^j\| = 0$ *for all* $i \in [N]$.

2. *(Stationary point)* $\lim_{t\to\infty} \|\boldsymbol{\theta}_t^i - \bar{\boldsymbol{\theta}}\| = 0$*, where* $\bar{\boldsymbol{\theta}}$ *is a stationary point of* (5.1).

Finally, we remark that in a centralized setting, the PG algorithm is known to converge at a linear rate for strongly convex objective functions. Such convergence rate is not observed for the DPG algorithm since the latter requires a diminishing step size to guarantee convergence. An active research area is to develop DPG-like algorithms that achieve linear convergence using a constant step size, e.g., [Shi *et al.*(2015), Nedić *et al.*(2016), Qu and Li(2016)].

5.3 Decentralized Frank-Wolfe (DeFW) Algorithm

Before we move on, let us describe a few properties of Problem (5.1) that will be useful. In the following, we shall work with both Euclidean norms and general norms, denoted by $\|\cdot\|$. Firstly, the constraint set $\mathcal{C} \subseteq \mathbb{R}^d$ is convex and bounded with the diameter defined as:

$$\rho := \sup_{\boldsymbol{\theta},\boldsymbol{\theta}'\in\mathcal{C}} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_\star, \quad \bar{\rho} := \sup_{\boldsymbol{\theta},\boldsymbol{\theta}'\in\mathcal{C}} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2 \ , \tag{5.9}$$

where ρ is defined with respect to (w.r.t.) the dual norm $\|\cdot\|_\star$ while $\bar{\rho}$ is defined w.r.t. the Euclidean norm. When the objective function F is μ-strongly convex with $\mu > 0$, the optimal solution to (5.1) is unique and denoted by $\boldsymbol{\theta}^\star$, we also define

$$\delta := \inf_{\boldsymbol{s}\in\partial\mathcal{C}} \|\boldsymbol{s} - \boldsymbol{\theta}^\star\|_2 \ , \tag{5.10}$$

where $\partial\mathcal{C}$ is the boundary set of $\mathcal{C}$. If $\delta > 0$, the solution $\boldsymbol{\theta}^\star$ is in the interior of $\mathcal{C}$.

We develop the decentralized Frank-Wolfe (DeFW) algorithm from the classical FW algorithm [Frank and Wolfe(1956)]. Let $t \in \mathbb{N}$ be the iteration number and let the initial point $\boldsymbol{\theta}_0 \in \mathcal{C}$ is feasible. Recalling the definition $F(\boldsymbol{\theta}) := (1/N)\sum_{i=1}^N f_i(\boldsymbol{\theta})$, the *centralized*

FW algorithm for problem (5.1) proceeds by:

$$\boldsymbol{a}_{t-1} \in \arg\min_{\boldsymbol{a} \in \mathcal{C}} \ \langle \nabla F(\boldsymbol{\theta}_{t-1}), \boldsymbol{a} \rangle \ , \tag{5.11a}$$

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \gamma_{t-1}(\boldsymbol{a}_{t-1} - \boldsymbol{\theta}_{t-1}) \ , \tag{5.11b}$$

where $\gamma_{t-1} \in (0, 1]$ is a step size to be determined. Observe that $\boldsymbol{\theta}_t$ is a convex combination of $\boldsymbol{\theta}_{t-1}$ and $\boldsymbol{a}_{t-1}$ which are both feasible, therefore $\boldsymbol{\theta}_t \in \mathcal{C}$ as $\mathcal{C}$ is a convex set. When the step size is chosen as $\gamma_t = 2/(t+1)$, the FW algorithm is known to converge at a rate of $\mathcal{O}(1/t)$ if F is L-smooth and convex [Jaggi(2013)]. A main feature of the FW algorithm is that the linear optimization[1] (LO) (5.11a) can be solved more efficiently than computing a projection, leading to a *projection-free* algorithm. At the end of this section, we will illustrate a few examples of $\mathcal{C}$ with efficient LO computations.

To this end, one would be tempted to develop a DeFW algorithm in a similar fashion as the DPG algorithm, *i.e.,* simply modifying the centralized FW algorithm by replacing (5.11b) with an average consensus update while using the local gradient $\nabla f_i(\cdot)$ for the update direction (5.11a). However, as we shall explain later, this procedure may not converge to a meaningful solution of the problem (5.1).

Instead, we consider extending the FW algorithm to a decentralized setting with a double-consensus scheme. To do so, we replace *both* the global gradient and iterate, *i.e.,* $\nabla F(\boldsymbol{\theta}_t)$, $\boldsymbol{\theta}_t$, in (5.11) with their respective local approximations in a similar fashion as the strategy developed in [Johansson *et al.*(2008), Simonetto and Jamali-Rad(2016)]. In particular, let $\boldsymbol{\theta}_t^i$ denotes an auxiliary iterate kept by agent i at iteration t. Define the average iterate:

$$\bar{\boldsymbol{\theta}}_t := \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\theta}_t^i \ . \tag{5.12}$$

Also, define the local iterate $\bar{\boldsymbol{\theta}}_t^i$ as an approximation of the average iterate above kept by agent i. We require $\bar{\boldsymbol{\theta}}_t^i$ to track $\bar{\boldsymbol{\theta}}_t$ with an increasing accuracy. Let $\{\Delta p_t\}_{t \geq 1}$ be a

[1] Notice that (5.11a) is a convex optimization problem with a linear objective.

non-negative, decreasing sequence with $\Delta p_t \to 0$, we assume

H5.1 **($\{\Delta p_t\}_{t\geq 1}$)** *For all $t \geq 1$, it holds that:*

$$\max_{i\in[N]} \|\bar{\boldsymbol{\theta}}_t^i - \bar{\boldsymbol{\theta}}_t\|_2 \leq \Delta p_t \; . \tag{5.13}$$

To compute (5.11a), ideally each agent has to access the *global gradient*, $\nabla F(\bar{\boldsymbol{\theta}}_t)$. However, just the local function $f_i(\cdot)$ is available and agent i can only compute the local gradient $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$. Therefore, we also need to track the average gradient as

$$\overline{\nabla_t F} := \frac{1}{N}\sum_{j=1}^{N} \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \; , \tag{5.14}$$

by the local approximation $\overline{\nabla_t^i F}$. Note that $\overline{\nabla_t F}$ is close to $\nabla F(\bar{\boldsymbol{\theta}}_t)$ when each of the function $f_i(\boldsymbol{\theta})$ is smooth and $\bar{\boldsymbol{\theta}}_t^i$ is close to $\bar{\boldsymbol{\theta}}_t$. Let $\{\Delta d_t\}_{t\geq 1}$ be a non-negative, decreasing sequence with $\Delta d_t \to 0$, we assume:

H5.2 **($\{\Delta d_t\}_{t\geq 1}$)** *For all $t \geq 1$, it holds that:*

$$\max_{i\in[N]} \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2 \leq \Delta d_t \; . \tag{5.15}$$

Naturally, from the local approximation $\overline{\nabla_t^i F}$, the ith agent can compute the update direction $\boldsymbol{a}_t^i = \arg\min_{\boldsymbol{a}\in\mathcal{C}} \langle \overline{\nabla_t^i F}, \boldsymbol{a}\rangle$ and update $\boldsymbol{\theta}_{t+1}^i$ similarly as in (5.11b). To summarize, a prototype of the DeFW algorithm can be found in Algorithm 5.1.

Compared to the DPG method (5.46), we note that the DeFW algorithm requires an additional *aggregation* step to compute the approximate global gradient, while the global gradient is not required in the DPG method. The primary reason for this is the fact that the FW step computation (5.11a) is not *smooth* in general with respect to the gradient $\nabla F(\bar{\boldsymbol{\theta}}_t)$. Concretely, consider $\mathcal{C} = \{\boldsymbol{\theta} \in \mathbb{R}^2 : \|\boldsymbol{\theta}\|_1 \leq 1\}$ and let $\nabla F(\boldsymbol{\theta}) = (1, 1-\epsilon)$ and

Algorithm 5.1 Decentralized Frank-Wolfe (DeFW) — a prototype.

```
Input: Initial point $\boldsymbol{\theta}_1^i$ for $i = 1, ..., N$.
for $t = 1, 2, ...$ do
    Consensus: obtain the average parameter:
        $\bar{\boldsymbol{\theta}}_t^i \leftarrow \texttt{NetAvg}_t^i(\{\boldsymbol{\theta}_t^j\}_{j=1}^N), \quad \forall\ i \in [N] \ . \qquad (5.16)$
    Aggregating: obtain the average gradient:
        $\overline{\nabla_t^i F} \leftarrow \texttt{NetAvg}_t^i(\{\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\}_{j=1}^N), \quad \forall\ i \in [N] \ . \qquad (5.17)$
    Frank-Wolfe Step: update
        $\boldsymbol{\theta}_{t+1}^i \leftarrow (1-\gamma_t)\bar{\boldsymbol{\theta}}_t^i + \gamma_t \boldsymbol{a}_t^i \ \text{ where } \ \boldsymbol{a}_t^i = \arg\min_{\boldsymbol{\theta} \in \mathcal{C}} \langle \overline{\nabla_t^i F}, \boldsymbol{\theta} \rangle \ , \qquad (5.18)$
    for all agent $i \in [N]$ and $\gamma_t \in (0, 1]$ is a step size.
end for
Return: Approximate stationary point $\bar{\boldsymbol{\theta}}_{t+1}^i, \forall\ i \in [N]$.
```

$\nabla F(\boldsymbol{\theta}') = (1, 1+\epsilon)$ be two gradient vectors for any $\epsilon > 0$, we observe that

$$(-1, 0) = \arg\min_{\boldsymbol{a} \in \mathcal{C}} \ \langle \nabla F(\boldsymbol{\theta}), \boldsymbol{a} \rangle, \quad (0, -1) = \arg\min_{\boldsymbol{a} \in \mathcal{C}} \ \langle \nabla F(\boldsymbol{\theta}'), \boldsymbol{a} \rangle \ . \tag{5.19}$$

Therefore, a small perturbation to the gradient direction may lead to a huge difference in the FW direction $\boldsymbol{a}_t$ found. On the other hand, the projection operator in the DPG method is *non-expansive* such that it tolerates small changes in the gradient direction and retains the information in the gradient after the projection. Now, if the DeFW algorithm proceeds by taking $\boldsymbol{a}_t^i = \arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \nabla f_i(\bar{\boldsymbol{\theta}}_t^i), \boldsymbol{a} \rangle$ in a similar fashion as in the DPG method, the computed direction $\boldsymbol{a}_t^i$ can be greatly different from that of taking it with respect to the global gradient $\nabla F(\bar{\boldsymbol{\theta}}_t)$. Intuitively, this would prevent convergence to a stationary point of (5.1) since the computed directions are likely to be completely unrelated to the global gradient which the algorithm is supposed to follow. It is, therefore, necessary to adopt a two-steps average consensus procedure to implement the DeFW algorithm.

Algorithm 5.1 requires each agent to solve the LO (5.11a) independently at each iteration. As we have mentioned, this can be done more efficiently than its projection counterpart for several interesting cases of the constraint set $\mathcal{C}$. For example:

- When $\mathcal{C}$ is the ℓ_1 ball, $\mathcal{C} = \{\boldsymbol{\theta} \in \mathbb{R}^d : \|\boldsymbol{\theta}\|_1 \leq R\}$,

  $$\boldsymbol{a}_t^i = -R \cdot \mathbf{e}_k, \quad \text{where} \quad k \in \arg\max_{j \in [d]} \left| [\overline{\nabla_t^i F}]_j \right| . \tag{5.20}$$

  The solution above amounts to finding the coordinate index of $\overline{\nabla_t^i F}$ with the maximum magnitude. Importantly, this solution is only 1-sparse. Consequently, the tth iterate $\bar{\boldsymbol{\theta}}_t$ will be at most tN-sparse. The worst-case complexity of computing $\boldsymbol{a}_t^i$ is $\mathcal{O}(d)$; in comparison, the worst-case complexity for the projection into an ℓ_1 ball is $\mathcal{O}(d \log d)^2$.

- When $\mathcal{C}$ is the trace norm ball, $\mathcal{C} = \{\boldsymbol{\theta} \in \mathbb{R}^{m_1 \times m_2} : \|\boldsymbol{\theta}\|_{\sigma,1} \leq R\}$, where $\|\boldsymbol{\theta}\|_{\sigma,1}$ is the sum of the singular values of $\boldsymbol{\theta}$. Let $\boldsymbol{u}_1, \boldsymbol{v}_1$ be the top-1 left/right singular vector of $\overline{\nabla_t^i F}$, we have

  $$\boldsymbol{a}_t^i = -R \cdot \boldsymbol{u}_1 \boldsymbol{v}_1^\top . \tag{5.21}$$

  Importantly, at a target solution accuracy of δ, the top singular vectors can be computed with a complexity of $\mathcal{O}(\max\{m_1, m_2\} \log(1/\delta))$ using the power/Lanczos method if $\|\text{vec}(\overline{\nabla_t^i F})\|_0 = \mathcal{O}(\max\{m_1, m_2\})$. In comparison, the projection onto the trace norm ball requires a complexity of $\mathcal{O}(\max\{m_1 m_2^2, m_2 m_1^2\} \log(1/\delta))$ for computing the full SVD of an $m_1 \times m_2$ matrix [Golub and van Loan(2013)].

The examples above are relevant to the two applications described in Section 5.5. More recently, efficient implementations are found when $\mathcal{C}$ admits additional structure such as being the convex hull of all perfect matchings of a bipartite graph; see [Jaggi(2013)].

In Section 5.4, we show that the two conditions H5.1 and H5.2 are sufficient to show the convergence of Algorithm 5.1 for a wide class of optimization problems (including non-

[2] There exists a randomized, accelerated algorithm for projection in [Duchi *et al.*(2008)] with an *expected* complexity of $\mathcal{O}(d)$.

convex optimizations). However, before delving into the details of convergence analysis, let us demonstrate that these conditions can be satisfied by running a consensus based implementation of DeFW algorithm.

5.3.1 Implementation: Consensus-based DeFW

We now design a message exchange protocol such that H5.1 and H5.2 can be enforced by employing the average consensus (AC) protocol [Tsitsiklis(1984),Dimakis *et al.*(2010)] with a *fixed* number of update/communication rounds for the $\texttt{NetAvg}_t^i(\cdot)$ subroutine. Specifically, the following discussions are based on the static AC; note that it is possible to extend the protocol to a randomized setting for time-varying networks (e.g., with random link failures), see [Boyd *et al.*(2006)].

For each round of the AC update, the agents take a weighted average of the values from its neighbors according to the weighted adjacency matrix $\boldsymbol{A}$, e.g., by using an update equation like [cf. (2.2)]:

$$\boldsymbol{x}_i^{s+1} = \sum_{j=1}^{N} A_{ij} \cdot \boldsymbol{x}_j^s, \ \forall \ s \geq 1 \ . \tag{5.22}$$

As shown in Fact 2.1, the above recursion is non-expansive and computes the average of $\{\boldsymbol{x}_i^0\}_{i=1}^N$ at a geometric rate of $\sigma_2(\boldsymbol{A})$. Now, let us consider the in-network computation of $\bar{\boldsymbol{\theta}}_t^i$ in line 3 of the DeFW algorithm. Here, the $\texttt{NetAvg}_t^i(\cdot)$ subroutine in the *consensus step* is implemented by:

$$\bar{\boldsymbol{\theta}}_t^i = \sum_{j=1}^{N} A_{ij} \cdot \boldsymbol{\theta}_t^j \ , \tag{5.23}$$

i.e., we perform one round of the AC update. Since $A_{ij} = 0$ if $(i,j) \notin E$, the above operation is implemented using message exchanges among the neighbors of agent i.

Now, for some $\alpha \in (0,1]$, we define $t_0(\alpha)$ as the smallest integer such that

$$\sigma_2(\boldsymbol{A}) \leq \Big(\frac{t_0(\alpha)}{t_0(\alpha)+1} \Big)^{\alpha} \cdot \frac{1}{1 + (t_0(\alpha))^{-\alpha}} \ . \tag{5.24}$$

Notice that $t_0(\alpha)$ is upper bounded by:

$$t_0(\alpha) \leq \left\lceil (\sigma_2(\boldsymbol{A})^{-\frac{1}{1+\alpha}} - 1)^{-1} \right\rceil \ . \tag{5.25}$$

The following lemma can be easily proven:

Lemma 5.1 *Set the step size γ_t as $1/t^\alpha$ in the DeFW algorithm for some $\alpha \in (0,1]$, then $\bar{\boldsymbol{\theta}}_t^i$ in (5.23) satisfies H5.1 with*

$$\Delta p_t \leq \frac{C_p}{t^\alpha} \ , \ \forall \ t \geq 1 \ , \ \textit{where } C_p := (t_0(\alpha))^\alpha \cdot \sqrt{N} \bar{\rho} \ . \tag{5.26}$$

The proof is postponed to Appendix 5.A, which relies on using the fact that $\boldsymbol{\theta}_t^i$ is a linear combination of $\bar{\boldsymbol{\theta}}_{t-1}^i$ and $\boldsymbol{a}_{t-1}^i$, *i.e.,* iterates from the previous iteration. In particular, $\bar{\boldsymbol{\theta}}_{t-1}^i$ is already $\mathcal{O}(1/(t-1)^\alpha)$-close to the network average from the last iteration and the update direction $\boldsymbol{a}_{t-1}^i$ has to be weighted by the decaying step size γ_{t-1}.

In comparison to what we were able to establish above, the in-network computation of $\overline{\nabla_t^i F}$ in line 4 of the DeFW algorithm is less straightforward. Unlike the computation of $\bar{\boldsymbol{\theta}}_t$, computing $N^{-1} \sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$ to an accuracy of $\mathcal{O}(1/t^\alpha)$ by communicating the local gradient $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$ requires $\Omega(\log t)$ rounds of updates when the AC protocol is employed. One of the main technical issues is that the local gradient $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$ computed by the ith agent is in general different from the local gradient computed at the other agent, even when $\bar{\boldsymbol{\theta}}_t^i$ is close to $\bar{\boldsymbol{\theta}}_t^j$ for $j \neq i$.

We propose an approach that is inspired by the fast stochastic average gradient (SAGA) method [Defazio *et al.*(2014)] which re-uses the gradient approximate $\overline{\nabla_{t-1}^i F}$ from the last iteration. Notice that a similar technique is adopted in [Qu and Li(2016),Nedić *et al.*(2016), Lorenzo and Scutari(2016)] under the name of 'gradient tracking' for various decentralized methods. Here we provide a rate analysis with non-asymptotic constants. Specifically, define the following surrogate of local gradient at iteration t:

$$\nabla_t^i F := \overline{\nabla_{t-1}^i F} + \nabla f_i(\bar{\boldsymbol{\theta}}_t^i) - \nabla f_i(\bar{\boldsymbol{\theta}}_{t-1}^i),\ \forall\ i \in [N]\ . \tag{5.27}$$

When $t = 1$, we set $\nabla_1^i F = \nabla f_i(\bar{\boldsymbol{\theta}}_1^i)$. Notice that (5.27) computes the incremental update of the gradient from $\overline{\nabla_{t-1}^i F}$. Similar to (5.23), the $\texttt{NetAvg}_t^i(\cdot)$ subroutine in the *aggregating* step is implemented by:

$$\overline{\nabla_t^i F} = \sum_{j=1}^N A_{ij} \cdot \nabla_t^j F\ , \tag{5.28}$$

i.e., using just one round of the AC update on $\nabla_t^i F$. Below we show that the average gradient is preserved by the surrogate $\nabla_t^i F$ and $\overline{\nabla_t^i F}$ achieves an approximation error similar to that in Lemma 5.1:

Lemma 5.2 *Set the step size γ_t as $1/t^\alpha$ in the DeFW algorithm for some $\alpha \in (0, 1]$. Suppose that each of f_i is L-smooth, $\bar{\boldsymbol{\theta}}_t^i$ is updated according to* (5.23)*, then $\overline{\nabla_t^i F}$ in* (5.28) *satisfies*

$$N^{-1} \textstyle\sum_{i=1}^N \nabla_t^i F = N^{-1} \sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_t^i),\ \forall\ t \geq 1\ , \tag{5.29}$$

and H5.2 with

$$\Delta d_t \leq C_g / t^\alpha,\ \forall\ t \geq 1\ , \tag{5.30}$$

where

$$C_g := (t_0(\alpha))^\alpha \cdot 2\sqrt{N}(2C_p + \bar{\rho})L\ . \tag{5.31}$$

The proof can be found in Appendix 5.B. Similar intuition as in Lemma 5.1 was used in the proof. In particular, we observe that $\overline{\nabla_{t-1}^i F}$ is $\mathcal{O}(1/(t-1)^\alpha)$-close to the network average $\overline{\nabla_{t-1} F}$ from the previous iteration and $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i) - \nabla f_i(\bar{\boldsymbol{\theta}}_{t-1}^i)$ scales as $\|\bar{\boldsymbol{\theta}}_t^i - \bar{\boldsymbol{\theta}}_{t-1}^i\|_2 \leq \Delta p_{t-1} = \mathcal{O}(1/(t-1)^\alpha)$ since f_i is L-smooth.

Remark 5.1 *It is possible for the agents to repeat the updates in* (5.23), (5.28) *for multiple rounds. Mathematically, this is equivalent to replacing A_{ij} in the above mentioned equations*

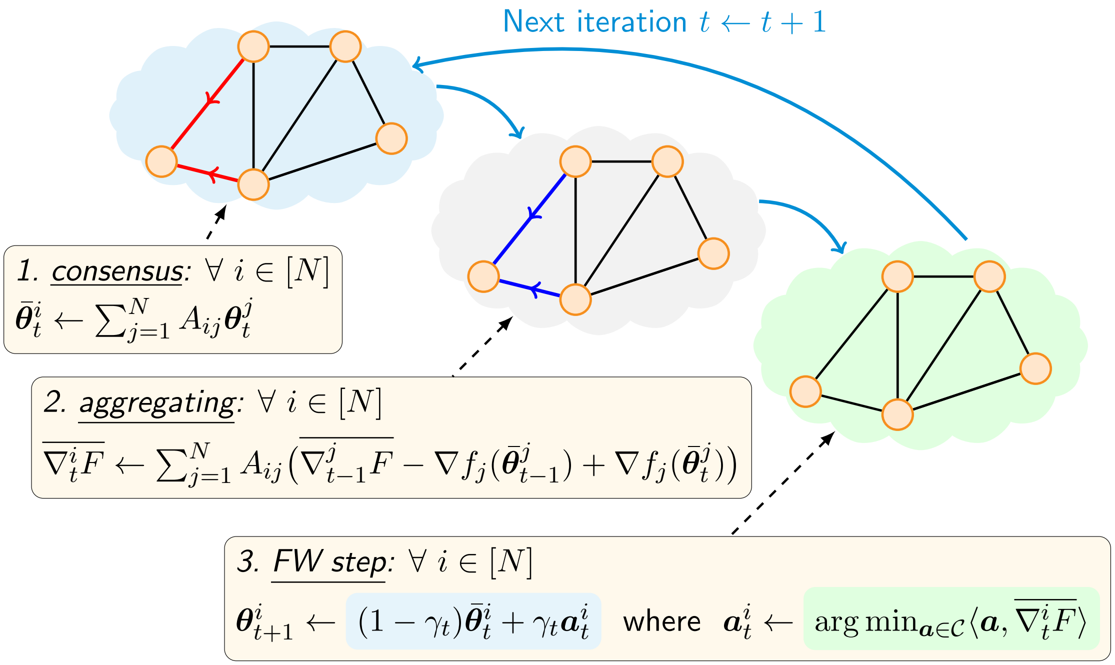
Next iteration $t \leftarrow t+1$
1. consensus: $\forall\, i \in [N]$
$\bar{\boldsymbol{\theta}}_t^i \leftarrow \sum_{j=1}^N A_{ij}\boldsymbol{\theta}_t^j$
2. aggregating: $\forall\, i \in [N]$
$\overline{\nabla_t^i F} \leftarrow \sum_{j=1}^N A_{ij}\big(\overline{\nabla_{t-1}^j F} - \nabla f_j(\bar{\boldsymbol{\theta}}_{t-1}^j) + \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\big)$
3. FW step: $\forall\, i \in [N]$
$\boldsymbol{\theta}_{t+1}^i \leftarrow (1-\gamma_t)\bar{\boldsymbol{\theta}}_t^i + \gamma_t \boldsymbol{a}_t^i$ where $\boldsymbol{a}_t^i \leftarrow \arg\min_{\boldsymbol{a}\in\mathcal{C}} \langle \boldsymbol{a}, \overline{\nabla_t^i F} \rangle$

Figure 5.1: Illustrating the DeFW algorithm as a three-steps recursive procedure.

by $[\boldsymbol{A}^\ell]_{ij}$. As $\sigma_2(\boldsymbol{A}^\ell) = \sigma_2(\boldsymbol{A})^\ell$, the constants $t_0(\alpha), C_p, C_g$ can be greatly reduced. Moreover, when $\bar{\rho}$ is large, performing multiple rounds of GAC updates within one iteration of DeFW can be beneficial for speeding up the algorithm in practice.

A summary of the above implementation of the DeFW algorithm can be found in Figure 5.1. We remark that the DeFW algorithm is not limited to the gossip-based implementation described above. In fact, any average consensus protocols which produce in-network averages satisfying H5.1, H5.2 with the desirable rates on $\Delta p_t, \Delta d_t$ can be applied. For example, when the graph G of the communication network is directed, one may apply the push-sum average consensus algorithm in [Tsianos *et al.*(2012)].

5.4 Convergence Analysis for General DeFW

We provide convergence analysis on the DeFW algorithm under general assumptions H5.1 and H5.2 for each agent $i \in [N]$. Under the said assumptions, the FW update step, *i.e.,* line 5, in Algorithm 5.1 can be regarded as performing the (centralized) FW updates

(5.11) on $\bar{\boldsymbol{\theta}}_t$ in an inexact manner. Below we characterize the convergence of the DeFW algorithm. For convex objective functions, we have:

Theorem 5.3 *Set the step size as $\gamma_t = 2/(t+1)$. Suppose that each of f_i is convex and L-smooth. Let C_p and C_g be two positive constants. Under H5.1-5.2 [$\Delta p_t = C_p/t$, $\Delta d_t = C_g/t$], we have*

$$F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star) \leq \frac{8\bar{\rho}(C_g + LC_p) + 2L\bar{\rho}^2}{t+1} \;, \tag{5.32}$$

for all $t \geq 1$, where $\boldsymbol{\theta}^\star$ is an optimal solution to (5.1)*. Furthermore, if F is μ-strongly convex and the optimal solution $\boldsymbol{\theta}^\star$ lies in the interior of $\mathcal{C}$,* i.e., *$\delta > 0$ (cf.* (5.10)*), we have*

$$F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star) \leq \frac{(4\bar{\rho}(C_g + LC_p) + L\bar{\rho}^2)^2}{2\delta^2\mu} \cdot \frac{9}{(t+1)^2} \;, \tag{5.33}$$

for all $t \geq 1$.

The proof can be found in Appendix 5.C. We remark that $\bar{\boldsymbol{\theta}}_t$ is always feasible. For strongly convex objective functions, the conditions (5.32), (5.33) imply that the sequence $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$ converges to an optimal solution of (5.1). Furthermore, as the consensus error, $\max_{i\in[N]} \|\bar{\boldsymbol{\theta}}_t^i - \bar{\boldsymbol{\theta}}_t\|_2$, decay to zero (cf. H5.1), the local iterates $\{\bar{\boldsymbol{\theta}}_t^i\}_{t\geq 1}$ share similar convergence guarantee as $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$.

For non-convex objective functions, we study the convergence of the FW/duality gap:

$$g_t := \max_{\boldsymbol{\theta}\in\mathcal{C}} \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta} \rangle \;. \tag{5.34}$$

From the definition, when $g_t = 0$, the iterate $\bar{\boldsymbol{\theta}}_t$ will be a stationary point to (5.1). Thus we may regard g_t as a measure of the stationarity of the iterate $\bar{\boldsymbol{\theta}}_t$. Also, define the set of stationary point to (5.1) as:

$$\mathcal{C}^\star = \big\{ \underline{\boldsymbol{\theta}} \in \mathcal{C} : \max_{\boldsymbol{\theta}\in\mathcal{C}} \; \langle \nabla F(\underline{\boldsymbol{\theta}}), \underline{\boldsymbol{\theta}} - \boldsymbol{\theta} \rangle = 0 \big\} \;. \tag{5.35}$$

We consider the following technical assumption:

H5.3 *The set $\mathcal{C}^\star$ is non-empty. Moreover, the function $F(\boldsymbol{\theta})$ takes a finite number of values over $\mathcal{C}^\star$,* i.e., *the set $F(\mathcal{C}^\star) = \{F(\boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathcal{C}^\star\}$ is finite.*

We now have:

Theorem 5.4 *Set the step size as $\gamma_t = 1/t^\alpha$ for some $\alpha \in (0, 1]$. Suppose each of f_i is L-smooth (possibly non-convex). Let C_p, C_g be two positive constants and $G = \max_i G_i$, where G_i is a Lipschitz constant for f_i. Under H5.1-5.2 [$\Delta p_t = C_p/t^\alpha$, $\Delta d_t = C_g/t^\alpha$], it holds that:*

1. *for all $T \geq 6$ that are even, if $\alpha \in [0.5, 1)$,*

$$\min_{t \in [T/2+1, T]} g_t \leq \frac{1}{T^{1-\alpha}} \cdot \frac{1-\alpha}{(1-(2/3)^{1-\alpha})} \cdot \Big(G\rho + (L\bar{\rho}^2/2 + 2\bar{\rho}(C_g + LC_p)) \log 2 \Big) \; ; \tag{5.36}$$

   if $\alpha \in (0, 0.5)$,

$$\min_{t \in [T/2+1, T]} g_t \leq \frac{1}{T^{\alpha}} \cdot \frac{1-\alpha}{(1-(2/3)^{1-\alpha})} \cdot \Big(G\rho + \frac{(L\bar{\rho}^2/2 + 2\bar{\rho}(C_g + LC_p))(1-(1/2)^{1-2\alpha})}{1-2\alpha} \Big) \; . \tag{5.37}$$

2. *if $\alpha \in [0.5, 1)$, exactly one of the following statement holds:*

   (a) for some $t_\star \in [T/2+1, T]$, the FW/duality gap satisfies:

$$\min_{t \in [T/2+1, T]} g_t \leq \frac{1}{(t_\star)^\alpha} \Big(2\bar{\rho}(C_p + LC_g) + \frac{L\bar{\rho}^2}{2} \Big) = \mathcal{O}\Big(\frac{1}{T^\alpha} \Big) \; . \tag{5.38}$$

   (b) the objective value is monotonically decreasing, i.e., *$F(\bar{\boldsymbol{\theta}}_{t+1}) < F(\bar{\boldsymbol{\theta}}_t)$ for all $t \in [T/2+1, T]$.*

3. *additionally, under H5.3 and $\alpha \in (0.5, 1]$, the sequence of objective values $\{F(\bar{\boldsymbol{\theta}}_t)\}_{t \geq 1}$ converges, $\{\bar{\boldsymbol{\theta}}_t\}_{t \geq 1}$ has limit points and each limit point is in $\mathcal{C}^\star$.*

The proof can be found in Appendix 5.D. Note that setting $\alpha = 0.5$ gives the quickest convergence rate of $\mathcal{O}(1/\sqrt{T})$. It is worth mentioning that our results are novel compared to prior work on non-convex FW even in a centralized setting ($N = 1, \Delta p_t = 0, \Delta d_t = 0$). For instance, [Ghosh and Lam(2015)] requires that the local minimizer is unique; [Lacoste-Julien(2016)] gives the same convergence rate but uses an adaptive step size. We remark that the local iterates $\{\bar{\boldsymbol{\theta}}_t^i\}_{t\geq 1}$ share similar convergence property as $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$ due to H5.1.

Notice that as Δp_t decays to zero as required in the theorems, the local iterates $\bar{\boldsymbol{\theta}}_t^i$ also converge to an optimal/stationary solution of (5.1). The above results give the conditions and the respective rates of convergence for the DeFW algorithm. Their proofs can be found in Appendix 5.C and Appendix 5.D. As a remark, the proof of Theorem 5.3 is an extension of our recent findings in [X2 of Section 1.3] on the convergence of online and stochastic FW algorithms; while the proof of Theorem 5.4 relies on bounding the duality gap (a.k.a. FW gap) $N^{-1}\sum_{i=1}^N \langle \overline{\nabla_t^i F}, \bar{\boldsymbol{\theta}}_t^i - \boldsymbol{a}_t^i \rangle$ defined similarly as in [Jaggi(2013)].

Finally, we also observe that the conditions on $\Delta p_t, \Delta d_t$ required by Theorem 5.3 and 5.4 can be satisfied by the $\texttt{NetAvg}_t^i(\cdot)$ subroutine implemented with the GAC protocol in (5.23) and (5.28) described in Section 5.3.1. This leads to the following corollary.

Corollary 5.1 *The convergence guarantees in Theorem 5.3 & 5.4 hold when the* $\texttt{NetAvg}_t^i(\cdot)$ *subroutine in line 3, line 4 of the DeFW algorithm are implemented by* (5.23), (5.28) *respectively.*

In other words, the consensus-based DeFW algorithm converges for both convex and non-convex problems, while using a constant number of communication rounds per iteration.

Remark 5.2 *From Theorem 5.4, for non-convex objectives, the best theoretical rate of convergence can be achieved when if we set* $\alpha = 0.5$ *as the learning rate. However, from Lemma 5.1 and 5.2, we notice that the approximation error also decays the slowest when* $\alpha = 0.5$. *From our numerical experience, we find that the approximation errors* $\Delta p_t, \Delta d_t$ *indeed play an important role in the practical performance of the DeFW algorithm. Therefore,* *we shall set α to be higher than* 0.5 *for better performance.*

5.5 Applications

In this section, we study two applications of the DeFW algorithm in signal processing and machine learning problems. Our aim is to demonstrate the advantages of the DeFW over conventional distributed algorithms in terms of complexity saving.

5.5.1 Example I: Decentralized Matrix Completion

Consider a setting when the network of agents obtain incomplete observations of a matrix $\boldsymbol{\theta}_{\sf true}$ of dimension $m_1 \times m_2$ with $m_1, m_2 \gg 0$. The ith agent has corrupted observations from the *training* set $\Omega_i \subset [m_1] \times [m_2]$ that are expressed as:

$$Y_{k,l} = [\boldsymbol{\theta}_{\sf true}]_{k,l} + Z_{k,l},\ \forall\ (k,l) \in \Omega_i\ . \tag{5.39}$$

To recover a low-rank $\boldsymbol{\theta}_{\sf true}$, we consider the following trace-norm constrained matrix completion (MC) problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^{m_1 \times m_2}} \sum_{i=1}^{N} \sum_{(k,l) \in \Omega_i} \tilde{f}_i([\boldsymbol{\theta}]_{k,l}, Y_{k,l}) \text{ s.t. } \|\boldsymbol{\theta}\|_{\sigma,1} \leq R\ , \tag{5.40}$$

where $\tilde{f}_i : \mathbb{R}^2 \to \mathbb{R}$ is a loss function picked by agent i according to the observations he/she received. Notice that (5.40) is also related to the low rank subspace system identification problem described in [Liu and Vandenberghe(2010)], where $\boldsymbol{Y}$ with $[\boldsymbol{Y}]_{k,l} = Y_{k,l}$, $\boldsymbol{\theta}_{\sf true}$ are modeled as the measured system response and the ground truth low rank response; also see [Scaglione *et al.*(2008)] for a related work.

Similar MC problems have been considered in [Ling *et al.*(2012), Mackey *et al.*(2015), Yu *et al.*(2012), Recht and Ré(2013)], where [Ling *et al.*(2012)] studied a consensus-based optimization method similar to ours and [Mackey *et al.*(2015), Yu *et al.*(2012), Recht and Ré(2013)] studied the parallel computation setting where the agents are working synchronously in a fully connected network. Compared to our approach, these work assume that the rank of $\boldsymbol{\theta}_{\sf true}$ is known in advance and solve the MC problem via matrix factorization. In addition, [Ling *et al.*(2012),Mackey *et al.*(2015)] required that each local observation set Ω_i only have entries taken from a disjoint subset of the columns/rows only. Our approach does not have any of the said restrictions above.

We consider two different observation models. When $Z_{k,l}$ is the i.i.d. Gaussian noise of variance σ_i^2, we choose $\tilde{f}_i(\cdot,\cdot)$ to be the square loss function, *i.e.*,

$$\tilde{f}_i([\boldsymbol{\theta}]_{k,l}, Y_{k,l}) := (1/\sigma_i^2) \cdot (Y_{k,l} - [\boldsymbol{\theta}]_{k,l})^2 \; . \tag{5.41}$$

This yields the classical MC problem in [Candès and Recht(2009)]. The next model considers the sparse+low rank matrix completion in [Chandrasekaran *et al.*(2011)], where the observations are contaminated with a sparse noise. Here, we model $Z_{k,l}$ as a *sparse* noise in the sense that there are a few number of entries in Ω_i where $Z_{k,l}$ is non-zero. We choose $\tilde{f}_i(\cdot,\cdot)$ to be the negated Gaussian loss, *i.e.*,

$$\tilde{f}_i([\boldsymbol{\theta}]_{k,l}, Y_{k,l}) := \Big(1 - \exp\Big(- \frac{([\boldsymbol{\theta}]_{k,l} - Y_{k,l})^2}{\sigma_i} \Big)\Big) \; , \tag{5.42}$$

where $\sigma_i > 0$ controls the robustness to outliers for the data obtained at the ith agent. Here, $\tilde{f}_i(\cdot,\cdot)$ is a *smoothed* ℓ_0 loss [Mohimani *et al.*(2007)] with enhanced robustness to outliers in the data. Notice that the resultant MC problem (5.40) is non-convex.

Note that (5.40) is a special case of problem (5.1) with $\mathcal{C}$ being the trace-norm ball. The consensus-based DeFW algorithm can be applied on (5.40) directly. The projection-free nature of the DeFW algorithm leads to a low complexity implementation (5.40). Lastly, several remarks on the communication and storage cost of the DeFW algorithm are in order:

- The SAGA-like gradient surrogate $\nabla_t^i F$ (5.27) is supported only on $\cup_{i=1}^N \Omega_i$ since for all $i \in [N]$, the local gradient

$$\nabla f_i(\bar{\boldsymbol{\theta}}_t^i) = \sum_{(k,l)\in\Omega_i} \tilde{f}_i'([\bar{\boldsymbol{\theta}}_t^i]_{k,l}, Y_{k,l}) \cdot \mathbf{e}_k(\mathbf{e}_l')^\top \tag{5.43}$$

is supported on Ω_i, where $\bar{\boldsymbol{\theta}}_t^i$ is defined in (5.23). In the above, $\mathbf{e}_k$ ($\mathbf{e}_l'$) is the kth (lth) canonical basis vector for $\mathbb{R}^{m_1}$ ($\mathbb{R}^{m_2}$) and $\tilde{f}_i'(\theta, y)$ is the derivative of $\tilde{f}_i(\theta, y)$ taken with respect to θ. Consequently, the average $\overline{\nabla_t^i F}$ is supported only on $\cup_{i=1}^N \Omega_i$. As $|\cup_{i=1}^N \Omega_i| \ll m_1 m_2$, the amount of information exchanged during the *aggregating* step (Line 3 in DeFW) is low.

- The update direction $\boldsymbol{a}_t^i$ is a rank-one matrix composed of the top singular vectors of $\overline{\nabla_t^i F}$ (cf. (5.21)). Since every iteration in DeFW adds at most N distinct pair of singular vectors to $\bar{\boldsymbol{\theta}}_t$, the rank of $\bar{\boldsymbol{\theta}}_t^i$ is upper bounded by tN if we initialize by $\bar{\boldsymbol{\theta}}_0^i = \mathbf{0}$. We can reduce the communication cost in Line 3 in DeFW by exchanging these singular vectors. Note that $(tN) \cdot (m_1 + m_2)$ entries are stored/exchanged instead of $m_1 \cdot m_2$.

- When the agents are *only* concerned with predicting the entries of $\boldsymbol{\theta}_{\mathsf{true}}$ in the subset $\Xi \subset [m_1] \times [m_2]$, instead of propagating the singular vectors as described above, the *consensus* step can be carried out by exchanging only the entries of $\boldsymbol{\theta}_{t+1}^i$ in $\Xi \cup \big(\cup_{i=1}^N \Omega_i \big)$ without affecting the operations of the DeFW algorithm. In this case, the storage/communication cost is $|\Xi \cup \big(\cup_{i=1}^N \Omega_i \big)|$.

5.5.2 Example II: Communication Efficient DeFW for LASSO

Let $(\boldsymbol{y}_i, \boldsymbol{A}_i)$ be the available data tuple at agent $i \in [N]$ such that $\boldsymbol{A}_i \in \mathbb{R}^{m \times d}$ and $\boldsymbol{y}_i \in \mathbb{R}^m$. The data $\boldsymbol{y}_i$ is a corrupted measurement of an unknown parameter $\boldsymbol{\theta}_{\mathsf{true}}$:

$$\boldsymbol{y}_i = \boldsymbol{A}_i \boldsymbol{\theta}_{\mathsf{true}} + \boldsymbol{z}_i \;, \tag{5.44}$$

where $\boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$ are independent noise vectors. Furthermore, we assume $m \ll d$ such that the matrix $\boldsymbol{A}_i^\top \boldsymbol{A}_i$ is rank-deficient. However, the parameter $\boldsymbol{\theta}_{\mathsf{true}}$ is s-sparse such that $s = \|\boldsymbol{\theta}_{\mathsf{true}}\|_0 \ll d$. This motivates us to consider the following distributed LASSO problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \; \sum_{i=1}^N \frac{1}{2} \|\boldsymbol{y}_i - \boldsymbol{A}_i \boldsymbol{\theta}\|_2^2 \;\; \text{s.t.} \;\; \|\boldsymbol{\theta}\|_1 \leq R \;. \tag{5.45}$$

Notice that equation (5.45) is a special case of (5.1) with $f_i(\boldsymbol{\theta}) = (1/2)\|\boldsymbol{y}_i - \boldsymbol{A}_i\boldsymbol{\theta}\|_2^2$ and $\mathcal{C}$ is an ℓ_1-ball in $\mathbb{R}^d$ with radius R. We assume that (5.45) has an optimal solution $\boldsymbol{\theta}^\star$ that is sparse. The settings above also correspond to identifying a linear system described by a sparse parameter $\boldsymbol{\theta}_{\sf true}$, where $\boldsymbol{A}_i$, $\boldsymbol{y}_i$ are the input, output of the system, respectively; see [Bako(2011)] for a related formulation on the identification of switched linear systems.

A number of decentralized algorithms are easily applicable to (5.45). For example, we recall that the DPG algorithm is described by — at iteration t,

$$\boldsymbol{\theta}_{t+1}^{i,PG} = \mathcal{P}_{\mathcal{C}}\big(\textstyle\sum_{j=1}^N A_{ij}\boldsymbol{\theta}_t^{j,PG} - \alpha_t \nabla f_i\big(\sum_{j=1}^N A_{ij}\boldsymbol{\theta}_t^{j,PG}\big)\big) \;, \tag{5.46}$$

where $\alpha_t \in (0,1]$ is a diminishing step size. For convex problems, the DPG algorithm is shown to converge to an optimal solution $\boldsymbol{\theta}^\star$ of (5.45) at a rate of $\mathcal{O}(1/\sqrt{t})$ [Chen(2012)].

Let us focus on the communication efficiency of the DPG algorithm, which is important when the network between agents is limited in bandwidth. To this end, we define the communication cost as the *number of non-zero real numbers exchanged per agent.* As seen from (5.46), at each iteration the ith agent exchanges its current iterate $\boldsymbol{\theta}_t^{i,PG}$ with the neighboring agents. From the computation step shown, $\boldsymbol{\theta}_t^{i,PG}$ may contain as high as $\mathcal{O}(d)$ non-zeros and the per-iteration communication cost will be $\mathcal{O}(d)$. Despite the high communication cost, the per-iteration computation complexity of (5.46) is also high, *i.e.,* at $\mathcal{O}(d \log d)$ [Duchi *et al.*(2008)]. We notice that [Ravazzi *et al.*(2016),Patterson *et al.*(2014)] have considered distributed sparse recovery algorithm with focus on the communication efficiency. However, their algorithms are based on the iterative hard thresholding (IHT) formulation [Blumensath and Davies(2008)] that requires a-priori knowledge on the sparsity level of $\boldsymbol{\theta}_{\sf true}$. Our consensus-based DeFW algorithm in Section 5.3.1 may also be applied directly to (5.45). However, similar issue as the DPG algorithm may arise during the *aggregating* step, since the gradient surrogate (5.27) may also have $\mathcal{O}(d)$ non-zeros. Lastly, another related work is [Yildiz and Scaglione(2008)] which applies coding to ‘compress’ the message exchanged in the consensus-based algorithms.

This section proposes a *sparsified DeFW algorithm* for solving (5.45). The modified algorithm applies a novel 'sparsification' procedure to reduce communication cost during the iterations, which is enabled by the structure of the DeFW algorithm. To describe the sparsified DeFW algorithm, we first argue that the *consensus step* in the consensus-based DeFW should remain unchanged as it already has a low communication cost. From (5.20) and (5.23), we see that $\boldsymbol{\theta}_t^i$ is at most $(t-1)N+1$-sparse since $\boldsymbol{a}_t^i$ is always a 1-sparse vector[3] [cf. (5.20)]. As such, the communication cost of this step is bounded by tN.

Our focus is to improve the communication efficiency of *aggregating step*. Here, the key idea is that only the *largest magnitude coordinate* in $\overline{\nabla_t^i F}$ is sought when computing $\boldsymbol{a}_t^i$ (cf. (5.20)). As long as the largest magnitude coordinate in $\overline{\nabla_t^i F}$ is preserved, the updates in the DeFW algorithm can remain unaffected. This motivates us to 'sparsify' the gradient information at each iteration before exchanging them with the neighboring agents. Let $\Omega_t \subseteq [d]$ be the coordinates of the gradient information to be exchanged at iteration t. The agents exchange the following gradient surrogate in lieu of (5.27):

$$\widehat{\nabla_t^i F} := \big(\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\big) \odot \mathbf{1}_{\Omega_t}, \quad \text{where} \;\; \mathbf{1}_{\Omega_t} = \sum_{k \in \Omega_t} \mathbf{e}_k \; . \tag{5.47}$$

Let $\ell_t = \lceil C_l + \log(t)/\log \sigma_2(\boldsymbol{A})^{-1}|\rceil$ where C_l is some finite constant and $\sigma_2(\boldsymbol{A})$ is the second largest singular value of the weight matrix $\boldsymbol{A}$, the sparsified DeFW algorithm computes the approximate gradient average $\overline{\nabla_t^i F}$ in line 4 of Algorithm 5.1 by:

$$\overline{\nabla_t^i F} = \sum_{j=1}^N [\boldsymbol{A}^{\ell_t}]_{ij} \cdot \widehat{\nabla_t^j F} \; . \tag{5.48}$$

Note that (5.48) requires ℓ_t *rounds* of AC updates to be performed at iteration t, *i.e.,* a logarithmically increasing number of rounds of AC updates. The update direction $\boldsymbol{a}_t^i$ can then be computed by sorting the vector $\overline{\nabla_t^i F}$. As $\overline{\nabla_t^i F}$ is $|\Omega_t|$-sparse, this update direction can be computed in $\mathcal{O}(|\Omega_t|)$ time.

[3] As pointed out by [Jaggi(2013)], this observation also leads to an interesting sparsity-accuracy trade-off when applying FW on ℓ_1 constrained problems.

We pick the coordinate set Ω_t in a decentralized manner. Consider the following decomposition:

$$\Omega_t = \bigcup_{i=1}^{N} \Omega_{t,i} \;, \tag{5.49}$$

where $\Omega_{t,i} \subset [d]$ is picked by agent i at iteration t. The coordinate set Ω_t needs to be known by all agents before (5.48). This can be achieved with low communication overhead, e.g., by forming a spanning tree on the graph G and broadcasting the required indices in Ω_t to all agents; see [Attiya and Welch(2004)]. Set p_t as the maximum desirable cardinality of $\Omega_{t,i}$, agent i chooses the coordinate set using one of the following two schemes:

- *(Random coordinate)* Each agent selects $\Omega_{t,i}$ by picking p_t coordinates uniformly (with replacement) from $[d]$.
- *(Extreme coordinate)* Each agent selects $\Omega_{t,i}$ as the p_t largest magnitude coordinates of the vector $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$.

For the random coordinate selection scheme, the following lemma shows that the gradient approximation error can be controlled at a desirable rate with an appropriate choice of p_t. Let $\xi_t := (1-(1-1/d)^{p_t N})$, we have:

Lemma 5.3 *Set $\epsilon > 0$ and $\ell_t = \lceil C_l + \log(t)/\log \sigma_2(\boldsymbol{A})^{-1} \rceil$. Let $p_t \geq C_0 t$ for some $C_0 < \infty$. With probability at least $1 - \pi^2\epsilon/6$, the following holds for all $\boldsymbol{\theta} \in \mathcal{C}$:*

$$\Big\| \xi_t^{-1} \overline{\nabla_t^i F} - \frac{1}{N} \sum_{i=1}^{N} \nabla f_i(\bar{\boldsymbol{\theta}}_t^i) \Big\|_\infty = \mathcal{O}\Big(\frac{d\sqrt{\log(t^2/\epsilon)}}{tN} \Big) \;, \tag{5.50}$$

for all $t \geq 1$ and $i \in [N]$.

The proof can be found in Appendix 5.E. Note that the above is given in terms of $\xi_t^{-1}\overline{\nabla_t^i F}$ instead of $\overline{\nabla_t^i F}$. However, the result remains relevant as the LO (5.11a) in the DeFW algorithm is *scale invariant, i.e.,* $\arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \overline{\nabla_t^i F}, \boldsymbol{a} \rangle = \arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \alpha \overline{\nabla_t^i F}, \boldsymbol{a} \rangle$ for any $\alpha > 0$. In other words, performing the FW step with $\overline{\nabla_t^i F}$ is equivalent to doing so with $\xi_t^{-1}\overline{\nabla_t^i F}$.

As $\xi_t^{-1}\overline{\nabla_t^i F}$ is an $\mathcal{O}(1/t)$ approximation to $N^{-1}\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$, therefore H5.2 is satisfied with $\Delta d_t = \mathcal{O}(1/t)$. Lastly, we conclude that

Corollary 5.2 *The sparsified DeFW algorithm using random coordinate selection,* i.e., *with line 3 and 4 in Algorithm 5.1 replaced by* (5.23) *and* (5.48), *respectively, generates iterates that satisfy the guarantees in Theorem 5.3 (with high probability). Under strong convexity and interior optimal point assumption, the communication complexity is* $\mathcal{O}(N \cdot (1/\delta) \cdot \log(1/\delta))$ *to reach a* δ*-optimal solution to* (5.45).

In Corollary 5.2, the first statement is a consequence of Lemma 5.3. The second statement can be verified by noting that reaching a δ-optimal solution requires $\mathcal{O}(1/\sqrt{\delta})$ iterations and the communication cost is $\mathcal{O}(Nt\log t)$ at iteration t, as the agents exchange an $\mathcal{O}(Nt)$-sparse vector for $\Theta(\log t)$ times.

5.6 Faster DeFW for Decentralized Low-rank Regression

This section proposes a tailor-made DeFW algorithm for low rank regression problems, which we shall call the Fast DeFW (F-DeFW). Specifically, we focus on tackling the following special case of (5.1):

$$\min_{\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{m_1 \times m_2}} \frac{1}{N}\sum_{i=1}^N \tilde{f}_i(\tilde{\boldsymbol{\theta}}) \quad \text{s.t.} \quad \|\tilde{\boldsymbol{\theta}}\|_{\sigma,1} \leq \frac{R}{2} \;, \tag{5.51}$$

where $\tilde{f}_i : \mathbb{R}^{m_1 \times m_2} \to \mathbb{R}$ is a proper and differentiable function with Lipschitz-continuous gradient and the constraint $\|\tilde{\boldsymbol{\theta}}\|_{\sigma,1} \leq \frac{R}{2}$ is enforced to promote a low rank solution. The above problem encompasses the case of decentralized matrix completion discussed in the last section. Obviously, the consensus-based DeFW algorithm discussed in the previous sections can be applied directly to solve the above problem. While the DeFW algorithm is projection free and can be used to handle the cases when $m_1, m_2 \gg 1$, each iteration of the algorithm still requires *each agent* to compute the top singular vector of a large matrix. The F-DeFW algorithm aims at *further* reducing the computational complexity of the DeFW algorithm. Our main idea is to distribute the computation burden of the top singular vectors to the agents via a *decentralized power method*. As we show below, the convergence of the F-DeFW algorithm can still be analyzed under the framework studied in Theorem 5.3 and 5.4.

Before we begin, let us introduce an equivalent form of (5.51) that will be easier to work with. Let $\boldsymbol{\theta}_1 \in \mathcal{S}^{m_1}$, $\boldsymbol{\theta}_2 \in \mathbb{R}^{m_1 \times m_2}$, $\boldsymbol{\theta}_3 \in \mathcal{S}^{m_2}$ be the sub-matrices of $\boldsymbol{\theta} \in \mathcal{S}^d$ such that $d := m_1 + m_2$ and $\delta \in \mathbb{R}$ be a constant,

$$\boldsymbol{\theta} := \begin{pmatrix} \boldsymbol{\theta}_1 & \boldsymbol{\theta}_2 \\ \boldsymbol{\theta}_2^\top & \boldsymbol{\theta}_3 \end{pmatrix} \quad \text{and} \quad f_i(\boldsymbol{\theta}) := \tilde{f}_i(\boldsymbol{\theta}_2) + (\delta/N)\mathrm{Tr}(\boldsymbol{\theta}) \; . \tag{5.52}$$

The following problem is equivalent to (5.51):

$$\min_{\boldsymbol{\theta} \in \mathcal{S}^d} \; F(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i=1}^N f_i(\boldsymbol{\theta}) \;\; \text{s.t.} \;\; \mathrm{Tr}(\boldsymbol{\theta}) = R, \; \boldsymbol{\theta} \succeq \mathbf{0} \; . \tag{5.53}$$

The equivalence follows from the following lemma:

Lemma 5.4 *[Jaggi and Sulovsky(2010), Lemma 1] Consider a non-zero matrix $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{m_1 \times m_2}$. We have the following equivalence:*

$$\|\tilde{\boldsymbol{\theta}}\|_{\sigma,1} \leq \frac{R}{2} \Longleftrightarrow \begin{array}{c} \exists \; \boldsymbol{\theta}_1 \in \mathcal{S}^{m_1}, \boldsymbol{\theta}_3 \in \mathcal{S}^{m_2} \text{ such that} \\ \begin{pmatrix} \boldsymbol{\theta}_1 & \tilde{\boldsymbol{\theta}} \\ \tilde{\boldsymbol{\theta}}^\top & \boldsymbol{\theta}_3 \end{pmatrix} \succeq \mathbf{0}, \; \mathrm{Tr}(\boldsymbol{\theta}_1) + \mathrm{Tr}(\boldsymbol{\theta}_3) = R. \end{array} \tag{5.54}$$

In particular, for any feasible solution $\tilde{\boldsymbol{\theta}}$ to (5.51), we can find a point $\boldsymbol{\theta}$ that is feasible to (5.1) and it satisfies $\tilde{f}_i(\tilde{\boldsymbol{\theta}}) = f_i(\boldsymbol{\theta}) - \delta R/N, \; \forall \; i \in [N]$. On the other hand, for any feasible $\boldsymbol{\theta}$ to (5.1), its sub-matrix $\boldsymbol{\theta}_2$ satisfies $\|\boldsymbol{\theta}_2\|_{\sigma,1} \leq R/2$ and is thus feasible to (5.51).

Consider applying the DeFW algorithm [Algorithm 5.1] to problem (5.53). Exploiting the structure of the constraint $\mathcal{C}$, the DeFW algorithm can be be described by the recursion:

$$\boldsymbol{\theta}_{t+1}^i = (1-\gamma_t)\bar{\boldsymbol{\theta}}_t^i + \gamma_t R \cdot \boldsymbol{a}_t^i (\boldsymbol{a}_t^i)^\top,\ \boldsymbol{a}_t^i = \mathsf{TopEV}(-\overline{\nabla_i^t F}) \;, \tag{5.55}$$

where γ_t is a decreasing step size, $\mathsf{TopEV}(\boldsymbol{X})$ is the *top eigenvector* of a symmetric matrix $\boldsymbol{X}$ and we have the approximations:

$$\bar{\boldsymbol{\theta}}_t^i \approx N^{-1} \sum_{j=1}^N \boldsymbol{\theta}_t^j,\ \ \overline{\nabla_t^i F} \approx N^{-1} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \;, \tag{5.56}$$

As explained before, these approximations can be obtained using an average consensus protocol which involves communications between agents on the network.

We observe that in the second equation of (5.55), one actually wishes to find the following unit norm vector:

$$\hat{\boldsymbol{a}}_t = \mathsf{TopEV}\big(- N^{-1} \textstyle\sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\big) \;, \tag{5.57}$$

i.e., the top eigenvector of the *exact* average gradient. Notice that $\hat{\boldsymbol{a}}_t = \boldsymbol{a}_t^i$ when $\overline{\nabla_t^i F} = N^{-1} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$, *i.e.,* $\overline{\nabla_t^i F}$ is exactly equal to the average gradient. Decentralized methods for estimating the top eigenvector from the sample covariance have been proposed in [Scaglione *et al.*(2008), Li *et al.*(2011a)]. Their convergence were only discussed empirically [Scaglione *et al.*(2008)] or in the asymptotic case [Li *et al.*(2011a)]. For us, instead the objective is to use a decentralized power method to obtain $\hat{\boldsymbol{a}}_t$ in (5.57). To this end, let $\boldsymbol{v}^0 \in \mathbb{R}^d$ be an initial random vector and $p \geq 1$, we need to compute

$$\bar{\boldsymbol{v}}^p = \Big(- \frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \Big) \cdot \boldsymbol{v}^{p-1}, \quad \boldsymbol{v}^p = \frac{1}{\|\bar{\boldsymbol{v}}^p\|} \cdot \bar{\boldsymbol{v}}^p \;. \tag{5.58}$$

It is well known that $\boldsymbol{v}^p$ converges to the top eigenvector of $\hat{\boldsymbol{a}}_t$ as $p \to \infty$ under mild conditions [Golub and van Loan(1996)]. In the following, we demonstrate how to compute $\hat{\boldsymbol{a}}_t$ in a decentralized fashion, which will lead to the design of our F-DeFW algorithm.

Decentralized Power Method. An important observation on (5.58) is that evaluating

Algorithm 5.2 Decentralized Power Method (DePM).

```
1: **Input**: Parameters $S, P \in \mathbb{N}$, local gradients $\{\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\}_{i=1}^N$.
2: For each $i \in [N]$, generate an initial point $\boldsymbol{v}_i^0 \neq \mathbf{0}$ as a $d$-dimensional Gaussian random vector.
3: for $p = 1, 2, ..., P$ do
4:     $\bar{\boldsymbol{v}}_i^{p,0} \leftarrow -\nabla f_i(\bar{\boldsymbol{\theta}}_t^i) \cdot \boldsymbol{v}_i^{p-1}, \ \forall \ i \in [N]$ .
5:     for $\ell = 1, 2, ..., S$ do
6:         $\bar{\boldsymbol{v}}_i^{p,\ell} \leftarrow \sum_{j=1}^N A_{ij} \cdot \bar{\boldsymbol{v}}_j^{p,\ell-1}, \ \forall \ i \in [N]$ .
7:     end for
8:     $\boldsymbol{v}_i^p \leftarrow \bar{\boldsymbol{v}}_i^{p,S} / \|\bar{\boldsymbol{v}}_i^{p,S}\|, \ \forall \ i \in [N]$ .
9: end for
10: **Return**: Approximate top eigenvector $\boldsymbol{v}_i^P, \ \forall \ i \in [N]$.
```

$\bar{\boldsymbol{v}}^p$ is equivalent to taking the *average* of the N vectors $\{-\nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \cdot \boldsymbol{v}^{p-1}\}_{j=1}^N$, where each of these vectors is locally computable. This motivates us to replace *each* recursion of the power method (5.58) by an average consensus step, yielding a decentralized power method (DePM), as summarized in Algorithm 5.2. For ease of presentation, we denote the ith agent's output of Algorithm 5.2, $\boldsymbol{v}_i^P$, as the subroutine $\mathsf{DePM}_i(\cdot)$ parameterized by S, P:

$$\boldsymbol{v}_i^P := \mathsf{DePM}_i\big(\{-\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\}_{i=1}^N; P; S\big), \ \forall \ i \in [N] \ . \tag{5.59}$$

Note that Line 6 is the gossip-based average consensus step repeated for L times [Tsitsiklis(1984), Dimakis *et al.*(2010)] where information exchanges occur with the agents transmitting a d-dimensional vector per round.

The DePM method requires only a *matrix-vector* product as indicated by Line 4. It is also privacy preserving as the agents only exchange the product $(-\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\boldsymbol{v}_i^{p-1})$, therefore the other agents do not know who holds what portion of the observations and an eavesdropper on the network cannot steal the data. Now, let us denote $\boldsymbol{M}_t := -N^{-1} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$, and state the following assumption:

H5.4 *The spectral gap $\sigma_1(\boldsymbol{M}_t) - \sigma_2(\boldsymbol{M}_t)$ is lower bounded by $\xi > 0$ and $\sigma_1(\boldsymbol{M}_t)$ is upper bounded by B. Also, $|\boldsymbol{u}_1^\top \boldsymbol{v}_i^0| > 0, \forall\ i \in [N]$ where $\boldsymbol{u}_1$ is the top eigenvector of $\boldsymbol{M}_t$.*

The DePM method with carefully designed parameters S, P attains a desirable accuracy with high probability (w.h.p.) in finite time.

Proposition 5.1 *Under H5.4, fix $1/2 > \epsilon > 0$ and $c > 0$. If the algorithm parameters satisfy that $S = \Omega(\log(1/(\xi \cdot \epsilon))/\log(1/\sigma_2(\boldsymbol{A})))$ and $P = \Omega((B/\xi) \cdot \log(d/c \cdot \epsilon))$, then with probability at least $1 - Nc$, we have:*

$$(\boldsymbol{u}_1^\top \boldsymbol{v}_i^P)^2 \geq 1 - \epsilon^2 \quad \textit{and} \quad (\boldsymbol{u}_j^\top \boldsymbol{v}_i^P)^2 \leq \epsilon^2,\ j = 2, ..., d\ , \tag{5.60}$$

for all $i \in [N]$, and $\boldsymbol{u}_j$ is the jth largest eigenvector of $\boldsymbol{M}_t$. Also,

$$\|\boldsymbol{v}_i^P (\boldsymbol{v}_i^P)^\top - \boldsymbol{v}_j^P (\boldsymbol{v}_j^P)^\top\| = \mathcal{O}(\epsilon),\ \forall\ i, j \in [N]\ . \tag{5.61}$$

The proof can be found in Appendix 5.F. The omitted constants in the $\Omega(\cdot)$ notations for S, P in Proposition 5.1 are logarithmic in the dimension d. The above proposition shows that by controlling the parameters S, P, the DePM can compute the top eigenvectors of the symmetric matrix $\big(- \sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\big)$ at an arbitrary complexity and make the same eigenvector available at all the N agents in the network.

Fast DeFW. Equipped with the DePM method, we now summarize the proposed *fast DeFW* (F-DeFW) algorithm in Algorithm 5.3, which is a two-stage algorithm with an FW update in the outer loop and the DePM method in the inner loop. In comparison to the DeFW algorithm, Algorithm 5.3 does not require a *consensus* step for exchanging the parameter variables $\{\boldsymbol{\theta}_t^i\}_{i=1}^N$. In fact, all the information exchange required are done within the DePM subroutine.

We can establish similar convergence guarantees as DeFW. Let $\boldsymbol{M}_t = - \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)/N$,

Algorithm 5.3 Fast DeFW (F-DeFW) Algorithm.

```
Input: Initial point $\bar{\boldsymbol{\theta}}_0^i$ for $i = 1, ..., N$.
for $t = 1, 2, ...$ do
    DePM Step: apply the decentralized power method:
        $\boldsymbol{a}_t^i \leftarrow \mathsf{DePM}_i(\{-\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)\}_{j=1}^N; P_t; L_t),\ \forall\ i \in [N]\ .$   (5.62)
    Frank-Wolfe Step: update
        $\bar{\boldsymbol{\theta}}_{t+1}^i \leftarrow (1-\gamma_t)\bar{\boldsymbol{\theta}}_t^i + \gamma_t R \cdot \boldsymbol{a}_t^i (\boldsymbol{a}_t^i)^\top,\ \forall\ i \in [N]\ ,$   (5.63)
end for
Return: An approximate solution $\bar{\boldsymbol{\theta}}_{t+1}^i$ for $i = 1, ..., N$.
```

$\bar{\boldsymbol{\theta}}_t := \sum_{i=1}^N \bar{\boldsymbol{\theta}}_t^i / N$ and $\mathcal{C}$ denotes the feasible set of (5.1). We have:

Theorem 5.5 *Suppose that H5.4 holds for all $t \geq 1$. Fix $\tilde{c} > 0$ and set the parameter $S_t = \Omega(\log(t/\xi)/\log(1/\sigma_2(\boldsymbol{A})))$, $P_t = \Omega((B/\xi) \cdot \log(dt(Nt^2/\tilde{c})))$. Algorithm 5.3 satisfies the following with probability at least $1 - (\pi^2/6)\tilde{c}$:*

- *(Convex loss) If each of f_i is convex, S-smooth and the step size is $\gamma_t = 2/(t+1)$, then:*

$$F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star) = \mathcal{O}(1/t),\ \forall\ t \geq 1\ , \tag{5.64}$$

  where $\boldsymbol{\theta}^\star$ is an optimal solution to (5.1).

- *(Non-convex loss) If each of f_i is S-smooth and the step size is $\gamma_t = t^{-\alpha}$ for some $\alpha \in [0.5, 1)$, then for all $T \geq 20$:*

$$\min_{t \in [T/2+1, T]} \max_{\boldsymbol{\theta} \in \mathcal{C}} \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta} \rangle = \mathcal{O}(1/T^{1-\alpha})\ . \tag{5.65}$$

Moreover, for all $i, j \in [N]$, we have $\|\bar{\boldsymbol{\theta}}_j^t - \bar{\boldsymbol{\theta}}_i^t\| = \mathcal{O}(1/t)$.

The proof can be found in Appendix 5.G. We note that the convergence analysis stems from Theorem 5.3 and 5.4 as we analyze the F-DeFW algorithm as instances of the FW algorithm with inexact gradients and iterates.

As a final remark, we note that due to the structure of $\nabla f_i(\boldsymbol{\theta})$ in (5.52), setting $\delta \neq 0$ is necessary to ensure that the spectral gap $\xi_t = \sigma_1(\boldsymbol{M}_t) - \sigma_2(\boldsymbol{M}_t)$ is non-zero, since otherwise the singular values of $\boldsymbol{M}_t$ will have multiplicity two. Unfortunately, there is no known non-trivial lower bound on ξ_t. Thus, one has to set the constant terms in P_t heuristically (this is also true for PG methods).

5.7 Numerical Results

We perform numerical experiments to verify our theoretical findings on the DeFW algorithm. The following discussions will focus on the two applications described in Section 5.5 using synthetic and real data. To simulate the decentralized optimization setting, we artificially construct a network of $N = 50$ agents, where the underlying communication network G is an Erdos-Renyi graph with connectivity of 0.1. For the AC steps (5.23), (5.28) & (5.48), the doubly stochastic matrix $\boldsymbol{A}$ is calculated according to the Metropolis-Hastings rule in [Xiao and Boyd(2004)].

5.7.1 Decentralized Matrix Completion

This section considers the decentralized matrix completion problem, where the goal is to predict the missing entries of an unknown matrix through corrupted partial measurements.

We consider two datasets — the first dataset is synthetically generated where the unknown matrix $\boldsymbol{\theta}_{\sf true}$ is rank-K and has dimensions of $m_1 \times m_2$; the matrix is generated as $\boldsymbol{\theta}_{\sf true} = \sum_{i=1}^{K} \boldsymbol{y}_i \boldsymbol{x}_i^\top / K$ where $\boldsymbol{y}_i, \boldsymbol{x}_i$ have i.i.d. $\mathcal{N}(0,1)$ entries and different settings of m_1, m_2, K will be experimented. The second dataset is the `movielens100k` dataset [Harper and Konstan(2015)]. The unknown matrix $\boldsymbol{\theta}_{\sf true}$ records the movie ratings of $m_1 = 943$ users on $m_2 = 1682$ movies; and a total of 10^5 entries in $\boldsymbol{\theta}_{\sf true}$ are available as integers ranging from 1 to 5. We divide the entries in the dataset into the training and testing sets and evaluate the mean square error (MSE) on the testing set as:

$$\text{MSE} = |\Omega_{test}|^{-1} \textstyle\sum_{(k,l) \in \Omega_{test}} \left| [\boldsymbol{\theta}_{true}]_{k,l} - [\hat{\boldsymbol{\theta}}]_{k,l} \right|^2 , \tag{5.66}$$

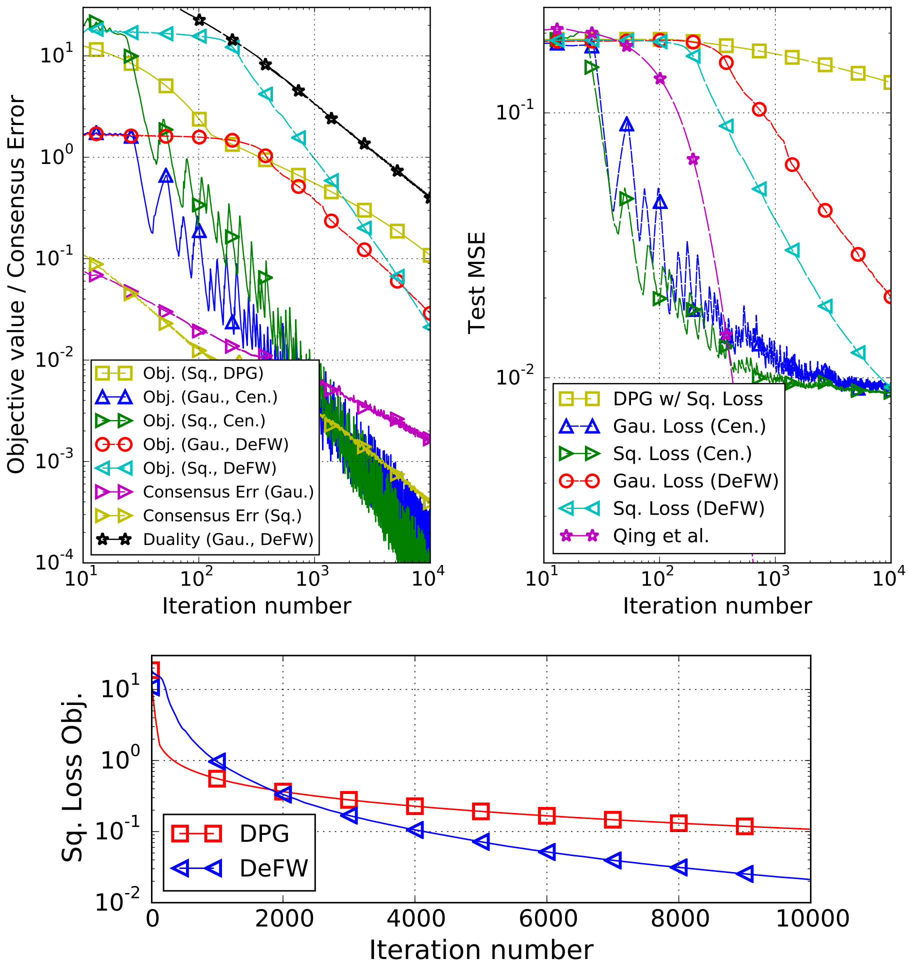

Figure 5.2: Performance of DeFW on noiseless synthetic data with $m_1 = 100, m_2 = 250$ and rank $K = 5$. (Top-Left) Objective value and consensus error of $\bar{\boldsymbol{\theta}}_t^i$ against iteration number t, the objective values are evaluated by $F(\bar{\boldsymbol{\theta}}_t)$. (Top-Right) Worst-case MSE (among agents) against iteration number on the testing set. (Bottom) Objective value (sq. loss) against iteration number t for DeFW and DPG. The legend 'Gau.', 'Sq.' denote the consensus-based DeFW algorithm applied to (5.40) with the negated Gaussian and square loss, respectively.

where $\hat{\boldsymbol{\theta}}$ denotes the estimated $\boldsymbol{\theta}$ produced by the algorithm.

For the synthetic dataset, the training (testing) set contains 20% (80%) entries which are selected randomly. For `movielens100k`, the training (testing) set contains 80×10^3 (20×10^3) entries. The training data of the two datasets are equally partitioned into $N = 50$ parts; for `movielens100k`, each agent holds 1600 entries. We evaluate the performance of the proposed consensus-based DeFW algorithm applied to square loss and negative Gaussian loss, as described in Section 5.5.1. Unless otherwise specified, we fix the number of AC rounds applied at $\ell = 1$ such that the agents only exchange information once per iteration. As the negated Gaussian loss is non-convex, we set the step size as $\gamma_t = t^{-0.75}$. The centralized FW algorithm for both losses will also be compared [cf. (5.11)]; as well as the decentralized algorithm in [Ling *et al.*(2012)] (labeled as 'Qing et al.') and the DPG algorithm [Ram *et al.*(2012)] with step size set to $\alpha_t = 0.1N/(\sqrt{t}+1)$ applied to square loss.

Our first example considers the noiseless synthetic dataset of problem dimension $m_1 = 100, m_2 = 250$, $K = 5$. The results are shown in Figure 5.2. Here, for the DeFW/DPG algorithms, we set the trace-norm radius to $R = 1.2\|\boldsymbol{\theta}_{\sf true}\|_{\sigma,1}$; and the algorithm in [Ling *et al.*(2012)] is supplied with the true rank K of $\boldsymbol{\theta}_{\sf true}$. Notice that for this set of data, the minimum of (5.40) can be achieved by $\boldsymbol{\theta} = \boldsymbol{\theta}_{\sf true} \in \mathcal{C}$ with a zero objective value. From the top-left plot, for the DeFW algorithm applied to the convex square loss function, we observe an $\mathcal{O}(1/t^2)$ trend for the objective values, corroborating with our analysis in Theorem 5.3; for the non-convex negated Gaussian loss function, the objective value and the FW/duality gap g_t also decay with t, the latter indicates the convergence to a stationary point. Moreover, the consensus error of $\bar{\boldsymbol{\theta}}_t^i$ for DeFW applied to the two objective functions decay at the rate predicted by Lemma 5.1. On the other hand, the top-right plot compares mean square error (MSE) of the predicted matrix $\boldsymbol{\theta}$ for the testing set. Here, we also compare the result with the algorithm in [Ling *et al.*(2012)]. We observe that the MSE performance of the DeFW algorithms approach their centralized counterpart as the iteration number grows, yet the algorithm in [Ling *et al.*(2012)] achieves the best performance in this setting. Notice that the true rank of $\boldsymbol{\theta}_{\sf true}$ is provided to this algorithm. From the bottom plot, the DPG method applied to square loss function converges at a relatively fast rate in the beginning, but was overtaken by DeFW as the iteration number grows. It is worth mentioning that the DeFW algorithms have a consistently better MSE performance than DPG.

The second example considers adding noise to the observations for the same synthetic data case in Figure 5.2. In particular, we adopt the same setting as the previous example

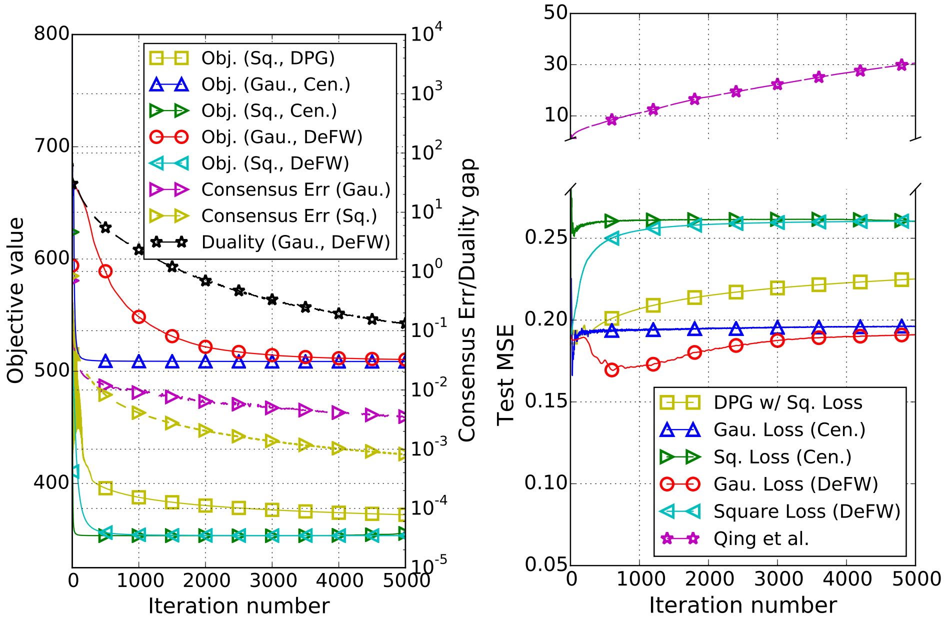

Figure 5.3: Performance of DeFW on sparse-noise contaminated synthetic data with $m_1 = 100, m_2 = 250$ and rank $K = 5$. (Left) Objective value and consensus error of $\bar{\boldsymbol{\theta}}_t^i$ against the DeFW iteration number t. Notice that the consensus error (in purple and yellow) / duality gap (in black) are plotted in a logarithmic scale (cf. the right y-axis) while the objective values are plotted in a linear scale; (Right) MSE against DeFW iteration number t on the testing set.

but include a *sparse* noise in the observations — here, each $Z_{k,l} = p_{k,l} \cdot \tilde{Z}_{k,l}$ where $p_{k,l}$ is Bernoulli with $P(p_{k,l} = 1) = 0.2$ and $\tilde{Z}_{k,l} \sim \mathcal{N}(0, 5)$ (cf. (5.39)). The convergence results are compared in Figure 5.3. For the left plot, we observe similar convergence behaviors for the DeFW algorithms applied to different objective functions as in the previous example. On the right plot, we observe that the DeFW algorithm based on negative Gaussian loss achieves the lowest MSE, demonstrating its robustness to outlier noise. We also see that the algorithm in [Ling *et al.*(2012)] performs poorly on this dataset.

Another interesting discovery is that the algorithm in [Ling *et al.*(2012)] seems to fail when the rank of $\boldsymbol{\theta}_{\sf true}$ is high, even when the true rank is known and the observations are noiseless. In Figure 5.4, we show the MSE against iteration number of the algorithms when the synthetic data is noiseless and generated with $m_1 = 100, m_2 = 250, K = 10$.

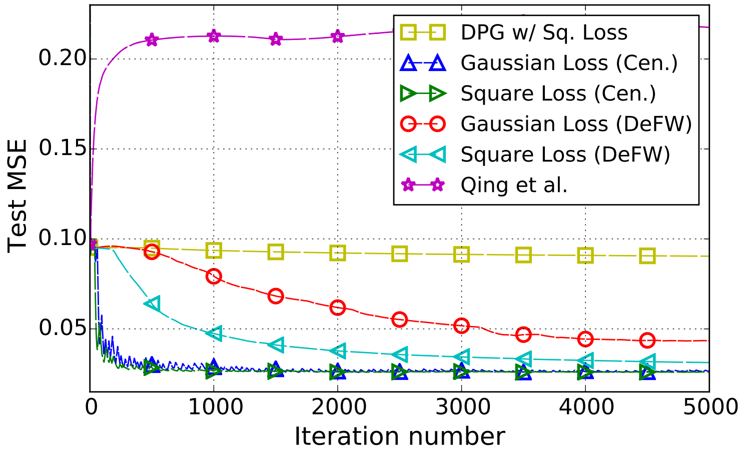

Figure 5.4: Convergence of test MSE against iteration number on the testing set on noise-free synthetic data with $m_1 = 100, m_2 = 250$ and rank $K = 10$.

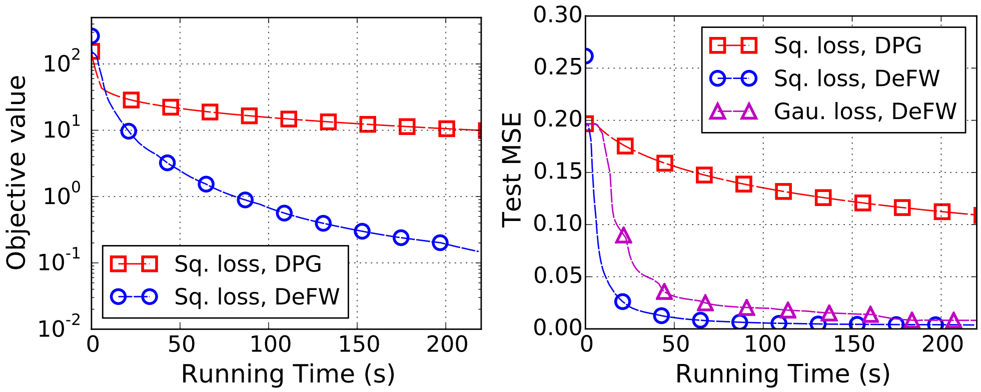

Figure 5.5: Performance of DeFW on noiseless synthetic data with $m_1 = 200, m_2 = 1000$ and rank $K = 5$. (Left) Objective value against running time. (Right) Worst-case MSE (among agents) against running time.

As seen, [Ling *et al.*(2012)] fails to produce a low MSE, while DeFW offers a reasonable performance.

The next example evaluates the objective value and test MSE on synthetic, noiseless data against the average runtime per agent. We focus on comparing the DeFW and DPG algorithms. In Figure 5.5, DeFW demonstrates a significant advantage over DPG since the former does not require the projection computation. In fact, the average running time *per iteration* of DeFW is five times faster than DPG. We also expect the complexity advantages to widen as the problem size grows.

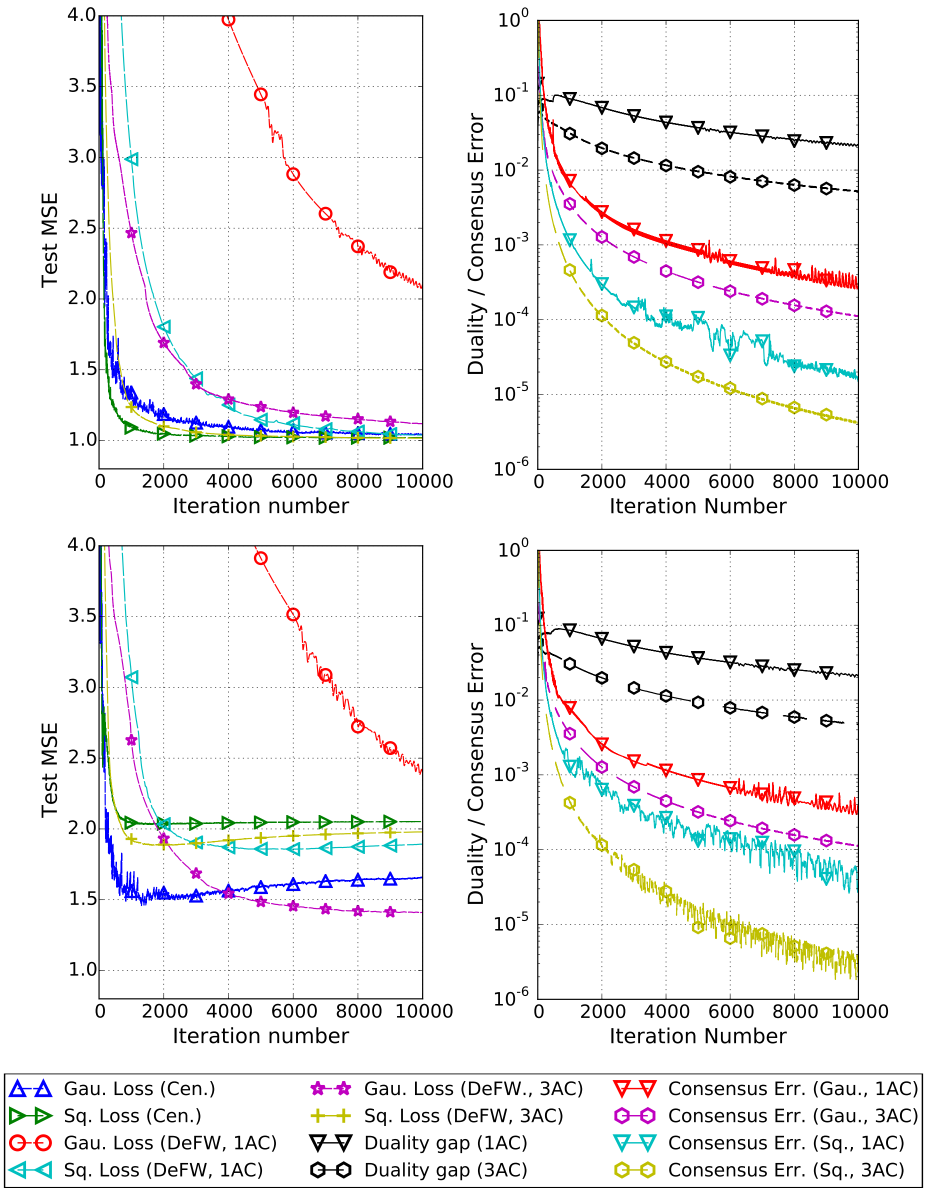

Figure 5.6: Convergence of the DeFW algorithm on `movielens100k` dataset with different loss functions. (Top) Noiseless observations; (Bottom) sparse-noise contaminated observations. Note that the duality gap / consensus errors are plotted in a logarithmic scale in the right figures.

Lastly, we consider the dataset `movielens100k`. We set $R = 10^5$ and focus on the test MSE evaluated against the iteration number for the proposed DeFW algorithm. The numerical results are presented in Figure 5.6, where we also compare the case when we apply multiple ($\ell = 1, 3$) rounds of AC updates per iteration to speed up the algorithm. The left plot in Figure 5.6 considers the noiseless scenario. As seen, the proposed DeFW algorithm applied on different loss functions converge to a reasonable MSE that is attained by the centralized FW algorithm. We see that the DeFW with negated Gaussian loss has a slower convergence compared to the square loss which is possibly attributed to the non-convexity of the loss function. Moreover, the algorithms achieve much faster convergence if we allow $\ell = 3$ AC rounds of network information exchange per iteration. The right plot in Figure 5.6 considers when the observations are contaminated with a sparse noise of the same model as Figure 5.3. We observe that the negated Gaussian loss implementations attain the best MSE as the non-convex loss is more robust against the sparse noise. Interestingly, the DeFW algorithm with $\ell = 3$ AC rounds has even outperformed its centralized counterpart. We suspect that this is caused by the DeFW algorithm converging to a different local minimum for the non-convex problem.

5.7.2 Decentralized Sparse Learning

This section conducts numerical experiments on the decentralized sparse learning problem. We focus on the *sparsified DeFW algorithm* in Section 5.5.2 that has better communication efficiency. We evaluate the performance on both synthetic and benchmark data. For the synthetic data, we randomly generate each $\boldsymbol{A}_i$ as a $(m = 20) \times (d = 10000)$ matrix with $\mathcal{N}(0, 1)$ elements (cf. (5.44)) and $\boldsymbol{\theta}_{\sf true}$ is a random sparse vector with $\|\boldsymbol{\theta}_{\sf true}\|_0 = 50$ such that the non-zero elements are also $\mathcal{N}(0, 1)$. The observation noise $\boldsymbol{z}_i$ has a variance of $\sigma^2 = 0.01$. For benchmark data, we test our method on `sparco7` [Berg *et al.*(2007)], which is a commonly used dataset for benchmarking sparse recovery algorithms. For `sparco7`, we have $\boldsymbol{A}_i \in \mathbb{R}^{12 \times 2560}$ as the local measurement matrix and $\boldsymbol{\theta}_{\sf true}$ is a sparse vector with $\|\boldsymbol{\theta}_{\sf true}\|_0 = 20$.

The sparsified DeFW algorithm is implemented with $p_t = \lceil 2 + \alpha_{comm} \cdot t \rceil$, $\ell_t = \lceil \log(t) + 1 \rceil$ with extreme or random coordinate selection. We compare the algorithms of PG-EXTRA [Shi *et al.*(2015)] (with fixed step size $\alpha = 1/d$), DPG [Ram *et al.*(2012)] (with step size $\alpha_t = 1/t$) and BHT [Ravazzi *et al.*(2016)]. DeFW, PG-EXTRA and DPG are set to solve the

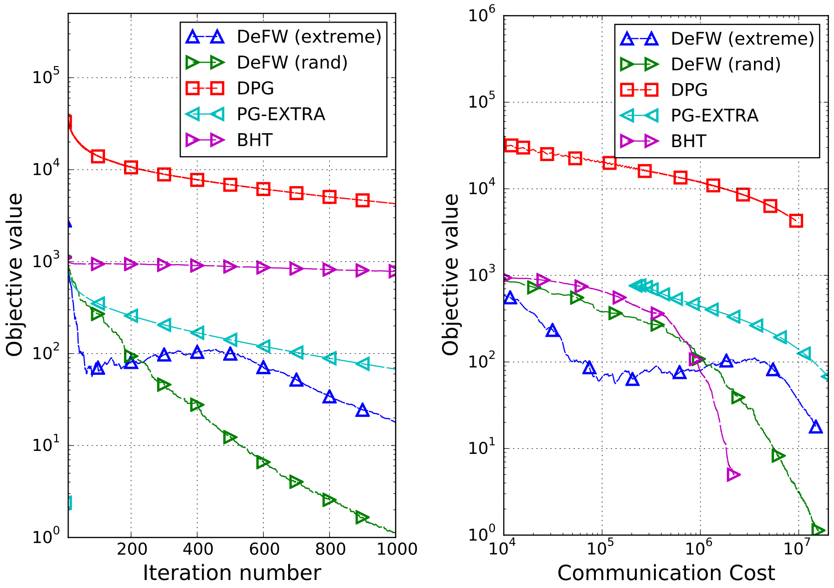

Figure 5.7: Convergence of the objective value on LASSO problem with synthetic dataset. (Left) against the iteration number. (Right) against the communication cost (*i.e.,* total number of values transmitted/received in the network during AC updates). In the legend, 'DeFW (extreme)' refers to the extreme coordinate selection and 'DeFW (rand)' refers to the random coordinate selection scheme.

convex problem (5.45) with $R = 1.1\|\boldsymbol{\theta}_{\mathsf{true}}\|_1$. BHT is a communication efficient decentralized version of IHT and is supplied with the true sparsity level in our simulations.

The first example in Figure 5.7 shows the convergence of the algorithms on the synthetic data, where we compare the objective value against the number of iterations and the communication cost, *i.e.,* total number of values sent during the distributed optimization. We set $\alpha_{comm} = 0.05$ for the DeFW algorithms. From the left plot, we observe that DeFW and PG-EXTRA algorithms have similar iteration complexity while 'DeFW (rand)' seems to have the fastest convergence. Meanwhile, BHT demands a high number of iterations for convergence. On the other hand, in the right plot, the DeFW algorithms demonstrate the best communication efficiency at low accuracy, while they lose to BHT at higher accuracy. Lastly, 'DeFW (extreme)' achieves a better accuracy at the beginning (*i.e.,* less communication cost paid) but is overtaken by 'DeFW (rand)' as the communication cost grows.

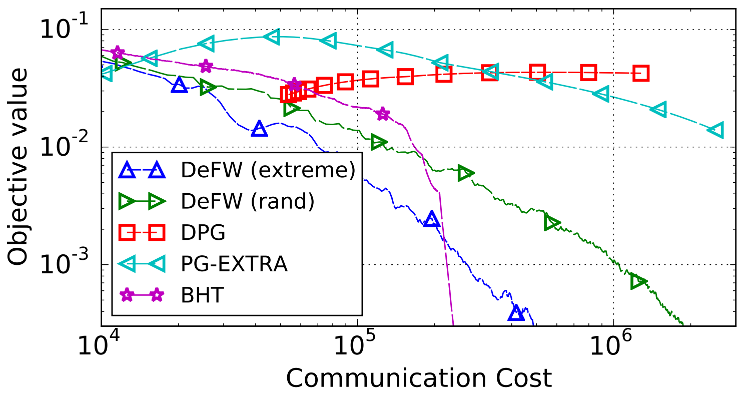

Figure 5.8: Convergence of the objective value against the communication cost on LASSO problem with `sparco7` dataset. In the legend, 'DeFW (extreme)' refers to the extreme coordinate selection and 'DeFW (rand)' refers to the random coordinate selection scheme.

We then compare the performance on `sparco7`, where we show the convergence of objective value against the communication cost in Figure 5.8. We set $\alpha_{comm} = 0.025$ for the sparsified DeFW algorithms. At low accuracy, the DeFW algorithms offer the best communication cost-accuracy trade-off, *i.e.,* it performs the best at an accuracy of above $\sim 10^{-2}$. Moreover, 'DeFW (extreme)' seems to be perform better than 'DeFW (rand)' in this example. Nevertheless, the BHT algorithm achieves the best performance when the communication cost paid is above 3×10^5. Lastly, we comment that although BHT has the lowest communication cost at *high* accuracy, its computational complexity is high as BHT requires a large number of iterations to reach a reasonable accuracy (cf. left plot of Figure 5.7). The sparsified DeFW offers a better balance of the communication and computation complexity.

5.7.3 Fast DeFW Algorithm

We focus on the matrix completion problem and consider the `movielens100k` dataset [Harper and Konstan(2015)] using the same setting considered previously. We simulate the distributed optimization environment by equally dividing the training set into N partitions. To satisfy the convergence conditions in Theorem 5.5, for the F-DeFW algorithm, we set $S_t = \lceil 3 + 2 \log t \rceil$, $P_t = 2S_t$ in DePM and $\delta = 10^{-4}$ in (5.52); for the convex square loss

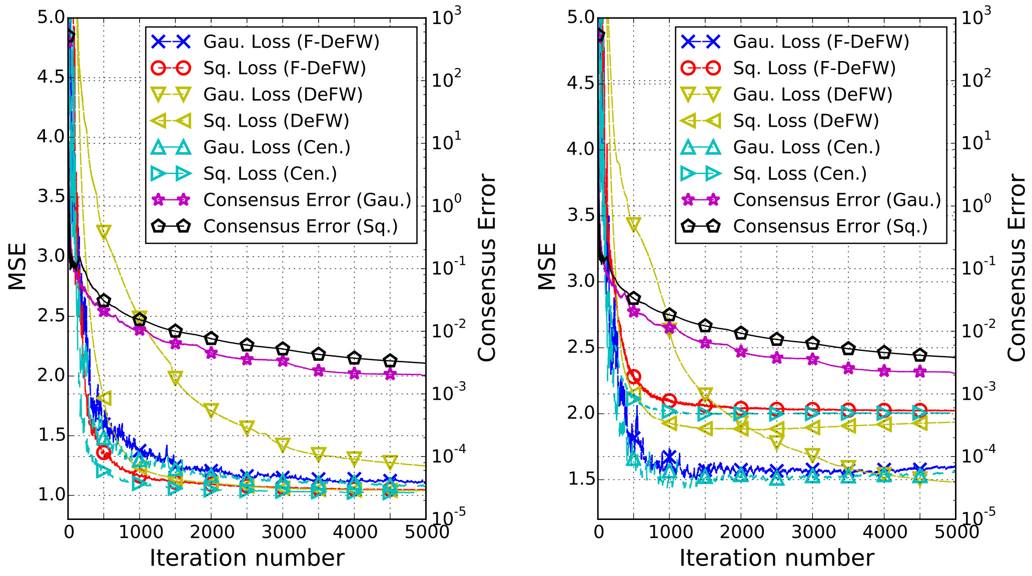

Figure 5.9: MSE on `movielens100k` against the F-DeFW iteration number t: (Left) noise-free observations; (Right) outlier-contaminated observations. We set $\sigma_i = 5$. Note the consensus error, $\max_{j \in [N]} \|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t\|$, of F-DeFW are plotted with the logarithmic scale (observe the different scale on the right y-axis).

(*resp.* non-convex Gaussian loss), we set the step size as $\gamma_t = 2/(t+1)$ (*resp.* $t^{-0.75}$). In addition to the *noise-free* setting when $Z_s^i = 0$ for all s, i, we also consider an *outliers-contaminated* setting when $Z_s^i = p_s^i \cdot \tilde{Z}_s^i$, where p_s^i is Bernoulli with $P(p_s^i = 1) = 0.2$ and $Z_s^i \sim \mathcal{N}(0, 5)$. We also compare the performance of the plain DeFW algorithm (with $\ell = 3$ average consensus steps per iteration) and a centralized FW algorithm.

We plot the mean square error (MSE) against the F-DeFW iteration number t on the testing set in Figure 5.9, where the worst MSE among the agents is evaluated for the decentralized algorithms. Observe that the MSE resulted from the F-DeFW algorithm follows closely with that of centralized FW algorithm. It also converges faster than DeFW and attains consensus gradually.

In Table 5.1, we compare the computation costs of the algorithms. As seen from the moderate number of matrix-vector multiplications required, the F-DeFW algorithm requires less computation time, but it requires more information exchange rounds than the DeFW algorithm. It is important to note that the size of the messages exchanged per round

	Runtime	#Matrix-vec. products	#Info. exchanges
Target MSE $= 1.4$ (Noise-free case, `movielens100k`)			
F-DeFW (Sq. loss)	**5.774 s**	**11978**	**167198**
F-DeFW (Gau. loss)	**10.548 s**	**23016**	**347580**
DeFW (Sq., $\ell = 3$)	28.377 s	N/A	4440
DeFW (Gau., $\ell = 3$)	160.09 s	N/A	18480
Target MSE $= 1.25$ (Noise-free case, `movielens100k`)			
F-DeFW (Sq. loss)	**8.809 s**	**19018**	**279838**
F-DeFW (Gau. loss)	**18.810 s**	**43220**	**700372**
DeFW (Sq., $\ell = 3$)	45.742 s	N/A	5778
DeFW (Gau., $\ell = 3$)	455.91 s	N/A	29556

Table 5.1: Computation and communication costs at different target MSEs. Notice that each communication round in F-DeFW requires sending a $d = (m_1+m_2)$-dimensional vector, while DeFW requires sending an $m_1 \times m_2$ matrix. The runtime represents the computation time *per agent*. It is calculated by dividing the overall time by N for our experiments performed on a single-threaded MATLAB environment.

for F-DeFW is much smaller (since $d = m_1 + m_2 \ll m_1 m_2$). We remark that the original DeFW algorithm already runs 10 to 20 times faster than a PG algorithm (e.g., D-PG [Nedić *et al.*(2010)]) as the latter requires computing a full SVD per iteration.

5.8 Chapter Summary

In this chapter, we have studied a decentralized projection-free algorithm for constrained optimization, which we called the DeFW algorithm. Importantly, we showed that the DeFW algorithm converges for both convex and non-convex loss functions and the respective convergence rates are analyzed. The efficacy of the proposed algorithm is demonstrated through tackling two problems related to machine learning and signal processing, with the advantages over previous state-of-the-art demonstrated through numerical experiments.

As an extension, we have also studied a low complexity modification of the DeFW algorithm, specialized to the low rank regression problems. The proposed Fast DeFW (F-DeFW) algorithm is also proven to converge at a similar rate as the plain DeFW algorithm in terms of the iteration complexity. However, the computation complexity required is shown to be even lower than the plain DeFW algorithm as it only requires the network's agents to perform elementary computations such as matrix-vector multiplications.

Appendix

5.A Proof of Lemma 5.1

For simplicity, we shall drop the dependence of α in the constant $t_0(\alpha)$. It suffices to show that for all $t \geq 1$,

$$\sqrt{\sum_{i=1}^{N} \|\bar{\boldsymbol{\theta}}_t^i - \bar{\boldsymbol{\theta}}_t\|_2^2} \leq \frac{C_p}{t^\alpha},\ C_p = (t_0)^\alpha \cdot \sqrt{N}\bar{\rho}\ . \tag{5.67}$$

We observe that for $t = 1$ to $t = t_0$, the above inequality is true since $\bar{\boldsymbol{\theta}}_t^i, \bar{\boldsymbol{\theta}}_t \in \mathcal{C}$ and the diameter of $\mathcal{C}$ is bounded by $\bar{\rho}$. For the induction step, let us assume that $\sqrt{\sum_{i=1}^N \|\bar{\boldsymbol{\theta}}_t^i - \bar{\boldsymbol{\theta}}_t\|^2} \leq C_p/t^\alpha$ for some $t \geq t_0$. Note that

$$\boldsymbol{\theta}_{t+1}^i = (1 - t^{-\alpha})\bar{\boldsymbol{\theta}}_t^i + t^{-\alpha}\boldsymbol{a}_t^i\ . \tag{5.68}$$

Denote $\tilde{\boldsymbol{a}}_t = N^{-1}\sum_{i=1}^N \boldsymbol{a}_t^i$ and using Fact 2.1, we observe that,

$$\sum_{i=1}^{N} \|\bar{\boldsymbol{\theta}}_{t+1}^i - \bar{\boldsymbol{\theta}}_{t+1}\|_2^2 \leq \sigma_2(\boldsymbol{A})^2 \cdot \sum_{j=1}^{N} \|(1 - t^{-\alpha})(\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t) + t^{-\alpha}(\boldsymbol{a}_t^j - \tilde{\boldsymbol{a}}_t)\|_2^2\ , \tag{5.69}$$

where we have used the fact $\bar{\boldsymbol{\theta}}_{t+1} = (1 - t^{-\alpha})\bar{\boldsymbol{\theta}}_t + t^{-\alpha}\tilde{\boldsymbol{a}}_t$. The right hand side in (5.69) can be bounded by

$$\begin{aligned}
&\sum_{j=1}^{N} \|(1 - t^{-\alpha})(\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t) + t^{-\alpha}(\boldsymbol{a}_t^j - \tilde{\boldsymbol{a}}_t)\|_2^2 \\
&\quad \leq \sum_{j=1}^{N} \big(\|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t\|_2^2 + t^{-2\alpha}\bar{\rho}^2 + 2\bar{\rho}t^{-\alpha}\|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t\|_2\big) \\
&\quad \leq t^{-2\alpha}(C_p^2 + N\bar{\rho}^2) + 2\bar{\rho}t^{-\alpha}\sqrt{N}\sqrt{\sum_{j=1}^{N} \|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t\|_2^2} \\
&\quad \leq t^{-2\alpha}(C_p + \sqrt{N}\bar{\rho})^2 \leq \Big(\frac{(t_0)^\alpha + 1}{(t_0)^\alpha \cdot t^\alpha} \cdot C_p\Big)^2\ ,
\end{aligned} \tag{5.70}$$

where we have used the boundedness of $\mathcal{C}$ in the first inequality, the norm equivalence $\sum_{j=1}^N |c_j| \leq \sqrt{N}\sqrt{\sum_{j=1}^N c_j^2}$ in the second inequality and the induction hypothesis in the third and fourth inequalities. Consequently, from (5.24), we observe that for all $t \geq t_0$,

$$\sigma_2(\boldsymbol{A}) \cdot \frac{(t_0)^\alpha + 1}{(t_0)^\alpha \cdot t^\alpha} \leq \frac{1}{(t+1)^\alpha} \ , \tag{5.71}$$

and the induction step is completed. Finally, Lemma 5.1 is proven by noting that (5.67) implies (5.26).

5.B Proof of Lemma 5.2

We prove the first condition (5.29) with a simple induction. This condition is obviously true for the base step $t = 1$. For induction step, suppose that (5.29) is true up to some t, then

$$\sum_{i=1}^N \nabla_{t+1}^i F = \sum_{i=1}^N (\overline{\nabla_t^i F} - \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)) + \sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_{t+1}^i) \ . \tag{5.72}$$

Note that the first term on the right hand side is zero due to the induction hypothesis. Thus, the induction step is completed and $N^{-1}\sum_{i=1}^N \nabla_t^i F = N^{-1}\sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$ for all $t \geq 1$.

Then, we prove the second condition (5.30). For simplicity, we drop the dependence of α in the constant $t_0(\alpha)$. Recall $\overline{\nabla_t F} := N^{-1}\sum_{i=1}^N \nabla_t^i F$. It suffices to prove:

$$\sqrt{\sum_{i=1}^N \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2^2} \leq \frac{C_g}{t^\alpha}, \ C_g = 2\sqrt{N}(t_0)^\alpha(2C_p + \bar{\rho})L \tag{5.73}$$

for all $t \geq 1$ using induction. For $t = 1$ to $t = t_0$, the inequality can be easily proven using the boundedness of the gradients. For the induction step, suppose $\sqrt{\sum_{i=1}^N \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2^2} \leq C_g/t^\alpha$ for some $t \geq t_0$. Define the slack variable $\delta f_{t+1}^i := \nabla f_i(\bar{\boldsymbol{\theta}}_{t+1}^i) - \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$. We observe that $\nabla^i_{t+1}F = \delta f^i_{t+1} + \overline{\nabla^i_t F}$ and $\overline{\nabla^i_{t+1}F} = \sum_{j=1}^N W_{ij} \nabla^j_{t+1}F$, thus applying Fact 2.1 yields

$$\sum_{i=1}^N \|\overline{\nabla^i_{t+1}F} - \overline{\nabla_{t+1}F}\|_2^2 \le \sigma_2(\boldsymbol{A})^2 \cdot \sum_{i=1}^N \|\overline{\nabla^i_t F} + \delta f^i_{t+1} - \overline{\nabla_{t+1}F}\|_2^2 \,. \tag{5.74}$$

Similarly, define $\delta F_{t+1} := \overline{\nabla_{t+1}F} - \overline{\nabla_t F} = N^{-1} \sum_{i=1}^N \delta f^i_{t+1}$ and observe that we can bound the right hand side of (5.74) as

$$\begin{aligned}
&\sum_{i=1}^N \|\overline{\nabla^i_t F} + \delta f^i_{t+1} - \overline{\nabla_{t+1}F}\|_2^2 \\
&\quad \le \sum_{i=1}^N \Big(\|\overline{\nabla^i_t F} - \overline{\nabla_t F}\|_2^2 + \|\delta f^i_{t+1} - \delta F_{t+1}\|_2^2 \\
&\qquad\qquad + 2 \cdot \|\delta f^i_{t+1} - \delta F_{t+1}\|_2 \cdot \|\overline{\nabla^i_t F} - \overline{\nabla_t F}\|_2 \Big)
\end{aligned} \tag{5.75}$$

where the first inequality is obtained by expanding the squared ℓ_2 norm and applying Cauchy-Schwartz inequality.

Note that for all $i \in [N]$, we have the following chain:

$$\begin{aligned}
\|\delta f^i_{t+1}\|_2 &= \|\nabla f_i(\bar{\boldsymbol{\theta}}^i_{t+1}) - \nabla f_i(\bar{\boldsymbol{\theta}}^i_t)\|_2 \le L \|\bar{\boldsymbol{\theta}}^i_{t+1} - \bar{\boldsymbol{\theta}}^i_t\|_2 \\
&\le L \Big\| \textstyle\sum_{j=1}^N A_{ij} \big((\boldsymbol{\theta}^j_{t+1} - \bar{\boldsymbol{\theta}}^j_t) + (\bar{\boldsymbol{\theta}}^j_t - \bar{\boldsymbol{\theta}}^i_t) \big) \Big\|_2 \\
&\le L \textstyle\sum_{j=1}^N A_{ij} \Big(t^{-\alpha} \bar{\rho} + 2C_p t^{-\alpha} \Big) = (2C_p + \bar{\rho}) L t^{-\alpha} \,,
\end{aligned} \tag{5.76}$$

where the last inequality is due to the convexity of ℓ_2 norm, the update rule in line 5 of Algorithm 5.1 and the results from Lemma 5.1. Using the triangular inequality, we obtain

$$\begin{aligned}
\|\delta f^i_{t+1} - \delta F_{t+1}\|_2 &= \Big\| \big(1 - \tfrac{1}{N}\big) \delta^i_{t+1} + \tfrac{1}{N} \textstyle\sum_{j \neq i} \delta^j_{t+1} \Big\|_2 \\
&\le \big(1 - \tfrac{1}{N}\big) \|\delta^i_{t+1}\|_2 + \tfrac{1}{N} \textstyle\sum_{j \neq i} \|\delta^j_{t+1}\|_2 \\
&\le 2\big(1 - \frac{1}{N}\big)(2C_p + \bar{\rho}) L t^{-\alpha} \le 2(2C_p + \bar{\rho}) L t^{-\alpha} \,.
\end{aligned} \tag{5.77}$$

Finally, applying the induction hypothesis, the right hand side of Eq. (5.75) can be bounded by

$$\begin{aligned} \sum_{i=1}^N \|\overline{\nabla_t^i F} + \delta f_{t+1}^i - \overline{\nabla_{t+1} F}\|_2^2 &\le t^{-2\alpha}\big(C_g^2 + 4N(2C_p+\bar{\rho})^2L^2\big) \\ &\quad + t^{-\alpha}4L(2C_p+\bar{\rho})\sqrt{N}\sqrt{\textstyle\sum_{i=1}^N \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2^2} \\ &\le t^{-2\alpha}\cdot\big(C_g + 2L\sqrt{N}(2C_p+\bar{\rho})\big)^2 \\ &\le \Big(\frac{(t_0)^\alpha+1}{(t_0)^\alpha\cdot t^\alpha}\cdot C_g\Big)^2, \end{aligned} \tag{5.78}$$

where we have used the fact that $\sum_{i=1}^N \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2 \le \sqrt{N}\sqrt{\sum_{i=1}^N \|\overline{\nabla_t^i F} - \overline{\nabla_t F}\|_2^2}$ in the first inequality. Invoking (5.24), we can upper bound the right hand side of (5.74) by $C_g^2/(t+1)^{2\alpha}$ for all $t \ge t_0$. Taking square root on both sides of the inequality completes the induction step. Consequently, (5.30) can be deduced from (5.73).

5.C Proof of Theorem 5.3

The proof of Theorem 5.3 follows from our recent analysis on online/stochastic FW algorithm [X2 of Section 1.3]. Using line 5 of Algorithm 5.1, we obtain:

$$\bar{\boldsymbol{\theta}}_{t+1} = \bar{\boldsymbol{\theta}}_t + \gamma_t(N^{-1}\textstyle\sum_{i=1}^N \boldsymbol{a}_t^i - \bar{\boldsymbol{\theta}}_t) \ . \tag{5.79}$$

Define $h_t := F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star)$ where $\boldsymbol{\theta}^\star$ is an optimal solution to (5.1). From the L-smoothness of F and the boundedness of $\mathcal{C}$, we have:

$$h_{t+1} \le h_t + \frac{\gamma_t}{N}\sum_{i=1}^N \langle \boldsymbol{a}_t^i - \bar{\boldsymbol{\theta}}_t, \nabla F(\bar{\boldsymbol{\theta}}_t)\rangle + \gamma_t^2\frac{L\bar{\rho}^2}{2} \ , \tag{5.80}$$

where $\bar{\rho}$ was defined in (5.9). We have the following chain of inequalities for the inner product: for each $i \in [N]$,

$$\begin{aligned} \langle \boldsymbol{a}_t^i - \bar{\boldsymbol{\theta}}_t, \nabla F(\bar{\boldsymbol{\theta}}_t) \rangle &\leq \langle \boldsymbol{a}_t^i - \bar{\boldsymbol{\theta}}_t, \overline{\nabla_t^i F} \rangle + \bar{\rho} \| \overline{\nabla_t^i F} - \nabla F(\bar{\boldsymbol{\theta}}_t) \|_2 \\ &\leq \langle \boldsymbol{a} - \bar{\boldsymbol{\theta}}_t, \overline{\nabla_t^i F} \rangle + \bar{\rho} \cdot \| \overline{\nabla_t^i F} - \nabla F(\bar{\boldsymbol{\theta}}_t) \|_2, \ \forall \ \boldsymbol{a} \in \mathcal{C} \\ &\leq \langle \boldsymbol{a} - \bar{\boldsymbol{\theta}}_t, \nabla F(\bar{\boldsymbol{\theta}}_t) \rangle + 2\bar{\rho} \cdot \| \overline{\nabla_t^i F} - \nabla F(\bar{\boldsymbol{\theta}}_t) \|_2, \ \forall \ \boldsymbol{a} \in \mathcal{C}, \end{aligned} \tag{5.81}$$

where we have added and subtracted $\overline{\nabla_t^i F}$ in the first inequality; and used the fact $\boldsymbol{a}_t^i \in \arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \boldsymbol{a}, \overline{\nabla_t^i F} \rangle$ in the second inequality. Recalling that $\overline{\nabla_t F} = N^{-1} \sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)$,

$$\begin{aligned} &\| \overline{\nabla_t^i F} - \nabla F(\bar{\boldsymbol{\theta}}_t) \|_2 \\ &\qquad \leq \| \overline{\nabla_t^i F} - \overline{\nabla_t F} \|_2 + \| \overline{\nabla_t F} - \nabla F(\bar{\boldsymbol{\theta}}_t) \|_2 \\ &\qquad \leq \Delta d_t + N^{-1} \textstyle\sum_{i=1}^N \| \nabla f_i(\bar{\boldsymbol{\theta}}_t^i) - \nabla f_i(\bar{\boldsymbol{\theta}}_t) \|_2 \\ &\qquad \leq \Delta d_t + L \cdot N^{-1} \textstyle\sum_{i=1}^N \| \bar{\boldsymbol{\theta}}_t^i - \bar{\boldsymbol{\theta}}_t \|_2 \\ &\qquad \leq \Delta d_t + L \cdot \Delta p_t \ , \end{aligned} \tag{5.82}$$

where the third inequality is due to the L-smoothness of $\{f_i\}_{i=1}^N$. Recalling that $\Delta p_t = C_p/t$, $\Delta d_t = C_g/t$ and substituting the results above into the inequality (5.80), we can see that

$$h_{t+1} \leq h_t + \gamma_t \langle \bar{\boldsymbol{a}}_t - \bar{\boldsymbol{\theta}}_t, \nabla F(\bar{\boldsymbol{\theta}}_t) \rangle + \gamma_t^2 \frac{L\bar{\rho}^2}{2} + 2\bar{\rho}\gamma_t \frac{C_g + LC_p}{t} \ , \tag{5.83}$$

where $\bar{\boldsymbol{a}}_t \in \mathcal{C}$ is the minimizer of the linear optimization (5.11a) using $\nabla F(\bar{\boldsymbol{\theta}}_t)$, *i.e.,*

$$\bar{\boldsymbol{a}}_t \in \arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \boldsymbol{a}, \nabla F(\bar{\boldsymbol{\theta}}_t) \rangle \ . \tag{5.84}$$

Case 1: When F is convex, we have

$$\langle \bar{\boldsymbol{a}}_t - \bar{\boldsymbol{\theta}}_t, \nabla F(\bar{\boldsymbol{\theta}}_t) \rangle \leq \langle \boldsymbol{\theta}^\star - \bar{\boldsymbol{\theta}}_t, \nabla F(\bar{\boldsymbol{\theta}}_t) \rangle \leq -h_t \ , \tag{5.85}$$

where the first inequality is due to the optimality of $\bar{\boldsymbol{a}}_t$ and the last inequality stems from the convexity of F. Plugging (5.85) into (5.83) yields

$$h_{t+1} \leq (1-\gamma_t)h_t + \gamma_t^2 \frac{L\bar{\rho}^2}{2} + \gamma_t \frac{2\bar{\rho}(C_g + LC_p)}{t} \ . \tag{5.86}$$

As $\gamma_t = 2/(t+1)$, from a high-level point of view, the preceding inequality behaves similarly to $h_{t+1} \leq (1-(1/t))h_t + \mathcal{O}(1/t^2)$. Consequently, applying [Polyak(1987), Lemma 4] yields a $\mathcal{O}(1/t)$ convergence rate for h_t. In fact, this is a deterministic version of the case analyzed by [Theorem 10 in X2 of Section 1.3]. In particular, setting $\alpha = 1, K = 2$ in [(56) in X2 of Section 1.3] and using an induction argument yield

$$h_t \leq 2 \cdot (4\bar{\rho}(C_g + LC_p) + L\bar{\rho}^2)/(t+1), \ \forall \ t \geq 1 \ . \tag{5.87}$$

Case 2: For the case when F is μ-strongly convex and $\boldsymbol{\theta}^\star$ lies in the interior of $\mathcal{C}$ with distance $\delta > 0$ (cf. (5.10)). Using [Lemma 6 in X2 of Section 1.3], we have

$$\langle \bar{\boldsymbol{\theta}}_t - \bar{\boldsymbol{a}}_t, \nabla F(\bar{\boldsymbol{\theta}}_t) \rangle \geq \sqrt{2\mu\delta^2 h_t} \ . \tag{5.88}$$

Plugging the preceding inquality into (5.83) gives

$$h_{t+1} \leq \sqrt{h_t}(\sqrt{h_t} - \gamma_t\sqrt{2\mu\delta^2}) + \gamma_t^2 \frac{L\bar{\rho}^2}{2} + \gamma_t \frac{2\bar{\rho}(C_g + LC_p)}{t} \ . \tag{5.89}$$

Compared to the case analyzed in (5.86), when h_t is decreased, the decrement in h_{t+1} is increased, leading to a faster convergence. This is a deterministic version of the case analyzed in [Theorem 7 in X2 of Section 1.3]. Setting $\alpha = 1, K = 2$ in [(48) in X2 of Section 1.3] and using an induction argument yield

$$h_t \leq \frac{(4\bar{\rho}(C_g + LC_p) + L\bar{\rho}^2)^2}{2\delta^2\mu} \cdot \frac{9}{(t+1)^2}, \ \forall \ t \geq 1 \ . \tag{5.90}$$

5.D Proof of Theorem 5.4

The first subsection proves the convergence rate condition in (5.36), while the second subsection proves that the DeFW algorithm converges to a stationary point of (5.1).

5.D.1 Convergence rate

Let us recall the definition of the *FW gap*:

$$g_t := \max_{\boldsymbol{\theta} \in \mathcal{C}} \; \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta} \rangle = \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t - \bar{\boldsymbol{a}}_t \rangle \; , \tag{5.91}$$

where we have used the definition of $\bar{\boldsymbol{a}}_t$ in (5.84). For simplicity, we shall assume that T is an even integer in the following.

From the L-smoothness of F, we have:

$$F(\bar{\boldsymbol{\theta}}_{t+1}) \leq F(\bar{\boldsymbol{\theta}}_t) + \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_{t+1} - \bar{\boldsymbol{\theta}}_t \rangle + \frac{L}{2} \| \bar{\boldsymbol{\theta}}_{t+1} - \bar{\boldsymbol{\theta}}_t \|_2^2 \; . \tag{5.92}$$

Observe that:

$$\bar{\boldsymbol{\theta}}_{t+1} - \bar{\boldsymbol{\theta}}_t = N^{-1} \textstyle\sum_{i=1}^N \gamma_t (\boldsymbol{a}_i^t - \bar{\boldsymbol{\theta}}_t^i) \; . \tag{5.93}$$

As $\boldsymbol{a}_t^i$, $\bar{\boldsymbol{\theta}}_t^i \in \mathcal{C}$, we have $\| \bar{\boldsymbol{\theta}}_{t+1} - \bar{\boldsymbol{\theta}}_t \|_2 \leq \gamma_t \bar{\rho}$. Using (5.81) and (5.82), the inequality (5.92) can be bounded as:

$$\begin{aligned} F(\bar{\boldsymbol{\theta}}_{t+1}) &\leq F(\bar{\boldsymbol{\theta}}_t) - \gamma_t \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t - \bar{\boldsymbol{a}}_t \rangle \\ &\qquad + 2\gamma_t \bar{\rho} \cdot (\Delta d_t + L \cdot \Delta p_t) + \gamma_t^2 L \bar{\rho}^2 / 2 \\ &= F(\bar{\boldsymbol{\theta}}_t) - \gamma_t g_t + 2\gamma_t \bar{\rho} \cdot (\Delta d_t + L \cdot \Delta p_t) + \gamma_t^2 \frac{L \bar{\rho}^2}{2} \; . \end{aligned} \tag{5.94}$$

From the definition, we observe that $g_t \geq 0$. Now, summing the two sides of (5.94) from

$t = T/2 + 1$ to $t = T$ gives:

$$\begin{aligned} \sum_{t=T/2+1}^{T} \gamma_t g_t \; & \leq \sum_{t=T/2+1}^{T} \Big(F(\bar{\boldsymbol{\theta}}_t) - F(\bar{\boldsymbol{\theta}}_{t+1}) \Big) \\ & + \sum_{t=T/2+1}^{T} \Big(2\gamma_t \bar{\rho} \cdot (\Delta d_t + L \cdot \Delta p_t) + \gamma_t^2 \frac{L\bar{\rho}^2}{2} \Big) \; . \end{aligned} \tag{5.95}$$

Canceling duplicated terms in the first term of the right hand side of the preceding inequality gives:

$$\begin{aligned} \sum_{t=T/2+1}^{T} \gamma_t g_t \; & \leq F(\bar{\boldsymbol{\theta}}_{T/2+1}) - F(\bar{\boldsymbol{\theta}}_{T+1}) \\ & + \sum_{t=T/2+1}^{T} \big(2\gamma_t \bar{\rho} \cdot (\Delta d_t + L \cdot \Delta p_t) + \gamma_t^2 \frac{L\bar{\rho}^2}{2} \big) \; . \end{aligned} \tag{5.96}$$

As $g_t, \gamma_t \geq 0$, we can lower bound the left hand side as:

$$\sum_{t=T/2+1}^{T} \gamma_t g_t \geq \Big(\min_{t \in [T/2+1, T]} g_t \Big) \cdot \Big(\sum_{t=T/2+1}^{T} \gamma_t \Big) \; , \tag{5.97}$$

and observe that for all $T \geq 6$ and $\alpha \in (0,1)$,

$$\sum_{t=T/2+1}^{T} \gamma_t \geq \frac{T^{1-\alpha}}{1-\alpha} \cdot \Big(1 - \Big(\frac{2}{3} \Big)^{1-\alpha} \Big) = \Omega(T^{1-\alpha}) \; . \tag{5.98}$$

Define the constant $C := L\bar{\rho}^2/2 + 2\bar{\rho}(C_g + LC_p)$. When $\alpha \geq 0.5$, using the fact that $\gamma_t = t^{-\alpha}$, $\Delta p_t = C_p/t^\alpha$, $\Delta d_t = C_g/t^\alpha$, the right hand side of (5.96) is bounded above by:

$$G \cdot \rho + C \cdot \textstyle\sum_{t=T/2+1}^{T} t^{-2\alpha} \leq G \cdot \rho + C \cdot \log 2 \; , \tag{5.99}$$

note that the series is converging as we are summing from $t = T/2 + 1$ to $t = T$. Dividing the preceding term by the lower bound (5.98) to $\sum_{t=T/2+1}^{T} \gamma_t$ yields (5.36).

On the other hand, when $\alpha < 0.5$, we notice that

$$\sum_{t=T/2+1}^{T} t^{-2\alpha} \leq \int_{T/2}^{T} t^{-2\alpha} \, dt = \frac{2^{1-2\alpha}-1}{1-2\alpha}\Big(\frac{T}{2}\Big)^{1-2\alpha}. \tag{5.100}$$

Therefore, the right hand side of (5.96) is bounded above by

$$G\rho + C \sum_{t=T/2+1}^{T} t^{-2\alpha} \leq \Big(G\rho + C \cdot \frac{1-(1/2)^{1-2\alpha}}{1-2\alpha} \Big) \cdot T^{1-2\alpha} \; . \tag{5.101}$$

Dividing the preceding term by the lower bound (5.98) to $\sum_{t=T/2+1}^{T} \gamma_t$ yields (5.37).

To prove the second statement in the theorem. Let us consider the inequality (5.94) again. We observe that

$$F(\bar{\boldsymbol{\theta}}_{t+1}) - F(\bar{\boldsymbol{\theta}}_t) \leq \gamma_t \Big(- g_t + 2\bar{\rho}(\Delta d_t + L\Delta p_t) + \gamma_t \frac{L\bar{\rho}^2}{2} \Big), \; \forall \; t \in [T/2+1, T] \; . \tag{5.102}$$

Notice that if the right hand side is negative for all $t \in [T/2+1, T]$, then

$$F(\bar{\boldsymbol{\theta}}_{t+1}) < F(\bar{\boldsymbol{\theta}}_t) \; . \tag{5.103}$$

Otherwise, for some $t_\star \in [T/2+1, T]$, it holds that

$$g_{t_\star} \leq 2\bar{\rho}(\Delta d_{t_\star} + L\Delta p_{t_\star}) + \gamma_{t_\star} \frac{L\bar{\rho}^2}{2} = \frac{1}{(t_\star)^\alpha} \Big(2\bar{\rho}(C_p + LC_g) + \frac{L\bar{\rho}^2}{2} \Big) = \mathcal{O}(1/T^\alpha) \; . \tag{5.104}$$

The proof is thus completed.

5.D.2 Convergence to stationary point

Recall that the set of stationary points to (5.1) is defined as:

$$\mathcal{C}^\star := \{ \bar{\boldsymbol{\theta}} \in \mathcal{C} : \max_{\boldsymbol{\theta} \in \mathcal{C}} \langle \nabla F(\bar{\boldsymbol{\theta}}), \bar{\boldsymbol{\theta}} - \boldsymbol{\theta} \rangle = 0 \} \; . \tag{5.105}$$

We state the following Nurminskii's sufficient condition:

Theorem 5.6 *[Nurminskii(1972), Theorem 1] Consider a sequence $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$ in a compact set $\mathcal{C}$. Suppose that the following hold*[4]*:*

A.1 $\lim_{t\to\infty} \|\bar{\boldsymbol{\theta}}_{t+1} - \bar{\boldsymbol{\theta}}_t\| = 0$.

A.2 Let $\underline{\boldsymbol{\theta}}$ be a limit point of $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$ and $\{\boldsymbol{\theta}_{s_t}\}_{t\geq 1}$ be a subsequence that converges to $\underline{\boldsymbol{\theta}}$. If $\underline{\boldsymbol{\theta}} \notin \mathcal{C}^\star$, then for any t and some sufficiently small $\epsilon > 0$, there exists a finite s such that $\|\bar{\boldsymbol{\theta}}_s - \bar{\boldsymbol{\theta}}_{s_t}\| > \epsilon$ and $s > s_t$.

A.3 Let $\underline{\boldsymbol{\theta}}$ be a limit point of $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$ and $\{\boldsymbol{\theta}_{s_t}\}_{t\geq 1}$ be a subsequence that converges to $\underline{\boldsymbol{\theta}}$. If $\underline{\boldsymbol{\theta}} \notin \mathcal{C}^\star$, then for any t and some sufficiently small $\epsilon > 0$, we can define

$$\tau_t := \min_{s > s_t} \; s \;\; \text{s.t.} \;\; \|\bar{\boldsymbol{\theta}}_s - \bar{\boldsymbol{\theta}}_{s_t}\| > \epsilon \tag{5.106}$$

where τ_t is finite. Also, there exists a continuous function $W(\bar{\boldsymbol{\theta}})$ that takes a finite number of values in $\mathcal{C}^\star$ with

$$\limsup_{t\to\infty} \; W(\bar{\boldsymbol{\theta}}_{\tau_t}) < \lim_{t\to\infty} W(\bar{\boldsymbol{\theta}}_{s_t}) \; . \tag{5.107}$$

Then the sequence $\{W(\bar{\boldsymbol{\theta}}_t)\}_{t\geq 1}$ converges and the limit points of the sequence $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$ belong to the set $\mathcal{C}^\star$.

We apply the above theorem to prove that every limit point of $\{\bar{\boldsymbol{\theta}}_t\}_{t\geq 1}$ is in $\mathcal{C}^\star$. First, A.1 can be easily verified since

$$\|\bar{\boldsymbol{\theta}}_{t+1} - \bar{\boldsymbol{\theta}}_t\| \leq \frac{\gamma_t}{N} \sum_{i=1}^N \|\boldsymbol{a}_t^i - \bar{\boldsymbol{\theta}}_t\| \leq \frac{\gamma_t \bar{\rho}}{N} \tag{5.108}$$

and we have $\gamma_t \to 0$ as $t \to \infty$.

As $\mathcal{C}$ is compact, there exists a convergent subsequence $\{\bar{\boldsymbol{\theta}}_{s_t}\}_{t\geq 1}$ of the sequence of iterates generated by the DeFW algorithm. Let $\underline{\boldsymbol{\theta}}$ be the limit point of $\{\bar{\boldsymbol{\theta}}_{s_t}\}_{t\geq 1}$ and $\underline{\boldsymbol{\theta}} \notin \mathcal{C}^\star$.

[4] To give a clearer presentation, we have rephrased conditions A.2 and A.3 from the original Nurminskii's conditions.

We shall verify A.2 by contradiction. In particular, fix a sufficiently small $\epsilon > 0$ and assume that the following holds:

$$\|\bar{\boldsymbol{\theta}}_s - \bar{\boldsymbol{\theta}}_{s_t}\| \leq \epsilon, \; \forall \; s > s_t, \; \forall \; t \geq 1 \; . \tag{5.109}$$

As $\{\bar{\boldsymbol{\theta}}_{s_t}\}_{t \geq 1}$ converges to $\underline{\boldsymbol{\theta}}$, the assumption (5.109) implies that for some sufficiently large t and any $s > s_t$, we have $\bar{\boldsymbol{\theta}}_s \in \mathcal{B}_{2\epsilon}(\underline{\boldsymbol{\theta}})$, *i.e.,* the ball of radius 2ϵ centered at $\underline{\boldsymbol{\theta}}$.

Since $\underline{\boldsymbol{\theta}} \notin \mathcal{C}^\star$, the following holds for some $\delta > 0$,

$$\langle \nabla F(\bar{\boldsymbol{\theta}}_s), \boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_s \rangle \leq -\delta < 0, \; \forall \; \boldsymbol{\theta} \in \mathcal{C}, \; \forall s > s_t \; . \tag{5.110}$$

In particular, we have $\langle \nabla F(\bar{\boldsymbol{\theta}}_s), \bar{\boldsymbol{a}}_s - \bar{\boldsymbol{\theta}}_s \rangle \leq -\delta$ as we recall that $\bar{\boldsymbol{a}}_s = \arg\min_{\boldsymbol{a} \in \mathcal{C}} \langle \nabla F(\bar{\boldsymbol{\theta}}_s), \boldsymbol{a} \rangle$.

On the other hand, from (5.94) and using H5.1, H5.2, it holds true for all $t \geq 1$ that:

$$\begin{aligned} F(\bar{\boldsymbol{\theta}}_{t+1}) - F(\bar{\boldsymbol{\theta}}_t) &\leq \gamma_t \cdot \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{a}}_t - \bar{\boldsymbol{\theta}}_t \rangle \\ &\qquad + \gamma_t \cdot \mathcal{O}(t^{-\alpha}) + \gamma_t^2 L \bar{\rho}^2 / 2 \; . \end{aligned} \tag{5.111}$$

To arrive at a contradiction, we let $s > s_t$ and sum up the two sides of (5.111) from $t = s_t$ to $t = s$. Consider the following chain of inequalities:

$$\begin{aligned} F(\bar{\boldsymbol{\theta}}_s) - F(\bar{\boldsymbol{\theta}}_{s_t}) &\leq \sum_{\ell = s_t}^{s} \gamma_\ell (\nabla F(\bar{\boldsymbol{\theta}}_\ell), \bar{\boldsymbol{a}}_\ell - \bar{\boldsymbol{\theta}}_\ell \rangle + \mathcal{O}(\ell^{-\alpha})) \\ &\leq -\delta \sum_{\ell = s_t}^{s} \gamma_\ell + \sum_{\ell = s_t}^{s} \gamma_\ell \mathcal{O}(\ell^{-\alpha}) \; , \end{aligned} \tag{5.112}$$

where the first inequality is due to the fact that $\gamma_\ell^2 L \bar{\rho}^2 / 2 = \gamma_\ell \mathcal{O}(\ell^{-\alpha})$ and the second inequality is due to (5.110). Rearranging the terms in (5.112), we have

$$F(\bar{\boldsymbol{\theta}}_s) - F(\bar{\boldsymbol{\theta}}_{s_t}) - \sum_{\ell = s_t}^{s} C \cdot \ell^{-2\alpha} \leq -\delta \sum_{\ell = s_t}^{s} \ell^{-\alpha} \; , \tag{5.113}$$

for some $C < \infty$. As $1 \geq \alpha > 0.5$, we have $\lim_{s \to \infty} \sum_{\ell = s_t}^{s} \ell^{-2\alpha} < \infty$ on the left hand side and $\lim_{s \to \infty} \sum_{\ell=s_t}^{s} \ell^{-\alpha} \to +\infty$ on the right hand side. Letting $s \to \infty$ on both sides of (5.113) implies

$$\lim_{s \to \infty} F(\bar{\boldsymbol{\theta}}_s) - F(\bar{\boldsymbol{\theta}}_{s_t}) < -\infty \; , \tag{5.114}$$

This leads to a contradiction to (5.110) since $F(\boldsymbol{\theta})$ is bounded over $\mathcal{C}$. We conclude that A.2 holds for the DeFW algorithm.

The remaining task is to verify A.3. We notice that the indices τ_t in (5.106) are well defined since A.2 holds. Take $W(\boldsymbol{\theta}) = F(\boldsymbol{\theta})$ and observe that the image $F(\mathcal{C}^\star)$ is a finite set [cf. H5.3]. By the definition of τ_t, we have $\bar{\boldsymbol{\theta}}_s \in \mathcal{B}_\epsilon(\bar{\boldsymbol{\theta}}_{s_t})$ for all $s_t \le s \le \tau_t - 1$. Again for some sufficiently large t, we have $\bar{\boldsymbol{\theta}}_s \in \mathcal{B}_\epsilon(\bar{\boldsymbol{\theta}}_{s_t}) \subseteq \mathcal{B}_{2\epsilon}(\underline{\boldsymbol{\theta}})$ and the inequality (5.112) holds for $s = \tau_t - 1$. This gives:

$$F(\bar{\boldsymbol{\theta}}_{\tau_t}) - F(\bar{\boldsymbol{\theta}}_{s_t}) \le \sum_{\ell=s_t}^{\tau_t - 1} \gamma_\ell \cdot (-\delta + \mathcal{O}(\ell^{-\alpha})) \; . \tag{5.115}$$

On the other hand, we have $\bar{\boldsymbol{\theta}}_{\tau_t} \notin \mathcal{B}_\epsilon(\bar{\boldsymbol{\theta}}_{s_t})$ and thus

$$\epsilon < \|\bar{\boldsymbol{\theta}}_{\tau_t} - \bar{\boldsymbol{\theta}}_{s_t}\| \le \sum_{\ell=s_t}^{\tau_t - 1} \gamma_\ell \Big\| \sum_{i=1}^{N} \frac{\boldsymbol{a}_\ell^i}{N} - \bar{\boldsymbol{\theta}}_\ell \Big\| \le \bar{\rho} \sum_{\ell=s_t}^{\tau_t - 1} \gamma_\ell \; . \tag{5.116}$$

The preceding relation implies that $\sum_{\ell=s_t}^{\tau_t-1} \gamma_\ell > \epsilon / \bar{\rho} > 0$. Considering (5.115) again, we obtain that $\mathcal{O}(\ell^{-\alpha})$ decays to zero, for some sufficiently large t, we have $-\delta + \mathcal{O}(\ell^{-\alpha}) \le -\delta' < 0$ if $\ell \ge s_t$. Therefore, (5.115) leads to

$$F(\bar{\boldsymbol{\theta}}_{\tau_t}) - F(\bar{\boldsymbol{\theta}}_{s_t}) \le -\delta' \sum_{\ell=s_t}^{\tau_t - 1} \gamma_\ell < -\frac{\delta' \epsilon}{\bar{\rho}} < 0 \; . \tag{5.117}$$

Taking the limit $t \to \infty$ on both sides leads to (5.107). The proof for the convergence to stationary point in Theorem 5.4 is completed by applying Theorem 5.6.

5.E Proof of Lemma 5.3

We begin the proof by applying the triangle inequality:

$$\begin{aligned}\Big\| \xi_{mean}^{-1} \overline{\nabla_t^i F} - \frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \Big\|_\infty &\leq \xi_{mean}^{-1} \cdot \Big\| \overline{\nabla_t^i F} - \frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \odot \mathbf{1}_{\Omega_t} \Big\|_\infty \\ &\quad + \Big\| \big(\frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \big) \odot (\xi_{mean}^{-1} \mathbf{1}_{\Omega_t} - \mathbf{1}) \Big\|_\infty ,\end{aligned} \tag{5.118}$$

where $\mathbf{1}$ denotes the all-one vector.

For the first term in the right hand side of (5.118), observe that $\overline{\nabla_t^i F}$ is obtained by applying the GAC updates on the sparsified local gradients $\nabla f_i(\bar{\boldsymbol{\theta}}_t^i) \odot \mathbf{1}_{\Omega_t}$ for $\ell_t = \lceil C_l + \log(t)/\log \sigma_2(\boldsymbol{A})^{-1} \rceil$ rounds, applying Fact 2.1 yields the following for all $i \in [N]$:

$$\begin{aligned}&\Big\| \overline{\nabla_t^i F} - \frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \odot \mathbf{1}_{\Omega_t} \Big\|_\infty \\ &\leq \sigma_2(\boldsymbol{A})^{\ell_t} \cdot \Big\| (\nabla f_i(\bar{\boldsymbol{\theta}}_t^i) - \frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)) \odot \mathbf{1}_{\Omega_t} \Big\|_\infty \leq \sigma_2(\boldsymbol{A})^{C_l} \cdot \frac{B}{t} ,\end{aligned} \tag{5.119}$$

for some $B < \infty$ since the gradients are bounded.

For the second term in (5.118), we first apply the inequality $\|\big(N^{-1} \sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\big) \odot (\xi_{mean}^{-1} \mathbf{1}_{\Omega_t} - \mathbf{1})\|_\infty \leq \|N^{-1} \sum_{i=1}^N \nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\|_\infty \|(\xi_{mean}^{-1} \mathbf{1}_{\Omega_t} - \mathbf{1})\|_\infty$ from [Horn and Johnson(1994)]. Now, the probability that coordinate k is included is given by:

$$P(k \in \Omega_t) = 1 - P\left(\bigcap_{i=1}^N k \notin \Omega_{t,i} \right) = 1 - \left(1 - \frac{1}{d}\right)^{p_t N} = \xi_{mean} , \tag{5.120}$$

and that $\mathbb{E}[\mathbf{1}_{\Omega_t}] = \xi_{mean} \mathbf{1}$. Then, observing that each element in $\xi_{mean}^{-1} \mathbf{1}_{\Omega_t}$ is bounded in $[0, \xi_{mean}^{-1}]$ and applying the Hoefding's inequality [Massart(2003)], the following holds true for all $x > 0$:

$$P\big(\|\xi_{mean}^{-1} \mathbf{1}_{\Omega_t} - \mathbf{1}\|_\infty \geq x \big) \leq 2d \cdot e^{-2x^2/\xi_{mean}^{-2}} , \tag{5.121}$$

where we have applied a union bound argument to take care of the ℓ_∞-norm.

Setting $x = \xi_{mean}^{-1}\sqrt{(\log(2dt^2) - \log\epsilon)/2}$ and applying another union bound show that with probability at least $1 - (\pi^2\epsilon/6)$, the following holds for all $t \geq 1$:

$$\left\| \frac{1}{N}\sum_{i=1}^{N} \nabla f_i(\bar{\boldsymbol{\theta}}_t^i) \odot (\xi_{mean}^{-1}\mathbf{1}_{\Omega_t} - \mathbf{1})\right\|_\infty \leq \xi_{mean}^{-1}\left\|\frac{1}{N}\sum_{i=1}^{N}\nabla f_i(\bar{\boldsymbol{\theta}}_t^i)\right\|_\infty \sqrt{\frac{\log(2dt^2/\epsilon)}{2}}\ , \quad (5.122)$$

As $d \gg 0$, we have $\xi_{mean}^{-1} \approx d/(p_t N)$. Recalling $p_t = \Theta(t)$ yields the desired result in Lemma 5.3.

5.F Proof of Proposition 5.1

With our choice of S, the error resulting from the gossiping step of Algorithm 5.2 (cf. Line 6) can be upper bounded as:

$$\begin{aligned}
\|\bar{\boldsymbol{v}}_i^{p,L} - \boldsymbol{M}_t\boldsymbol{v}_i^p\| &\leq \left\|\bar{\boldsymbol{v}}_i^{p,L} - \sum_{j=1}^{N}\frac{\bar{\boldsymbol{v}}_j^{p,0}}{N}\right\| + \left\|\sum_{j=1}^{N}\frac{\bar{\boldsymbol{v}}_j^{p,0}}{N} - \boldsymbol{M}_t\boldsymbol{v}_i^p\right\| \\
&\leq \mathcal{O}(\epsilon) + \frac{1}{N}\left\|\sum_{j=1}^{N}\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)(\boldsymbol{v}_i^p - \boldsymbol{v}_j^p)\right\| \qquad (5.123) \\
&\leq \mathcal{O}(\epsilon) + \frac{B}{N}\sum_{j=1}^{N}\|\boldsymbol{v}_i^p - \boldsymbol{v}_j^p\| \leq \mathcal{O}(\epsilon),\ \forall\ p \geq 1\ ,
\end{aligned}$$

where the second inequality and the last inequality are due to our choice of L and the geometric convergence of the gossip-based average consensus [Dimakis *et al.*(2010)]; the third inequality is due to the boundedness of $\nabla f_j(\bar{\boldsymbol{\theta}}_t^j)$, since f_j is smooth and the constraint set is bounded.

The above shows that the DePM can be analyzed as running N *noisy power methods* in parallel at N agents, each initialized by $\boldsymbol{v}_i^1$. Consequently, using our choice of P and applying [Hardt and Price(2014), Corollary 1.1], the following holds with probability at least $1 - Nc$ (we can get rid of the $e^{-\Omega(d)}$ term in [Hardt and Price(2014), Corollary 1.1] due to H5.4; see [Rudelson and Vershynin(2009)]):

$$\|(\boldsymbol{I} - \boldsymbol{v}_i^P(\boldsymbol{v}_i^P)^\top)\boldsymbol{u}_1\| \leq \epsilon,\ \forall\ i \in [N]\ , \quad (5.124)$$

which taking squares on the both side yields the first inequality in (5.60). The second inequality in (5.60) is derived from decomposing $\boldsymbol{v}_i^P$ into the orthonormal basis $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_d\}$. Lastly, the consensus condition (5.61) follows from our choice of L such that $\|\boldsymbol{v}_i^P - \boldsymbol{v}_j^P\| = \mathcal{O}(\epsilon)$ and the identity $\boldsymbol{v}_i^P(\boldsymbol{v}_i^P)^\top - \boldsymbol{v}_j^P(\boldsymbol{v}_j^P)^\top = ((\boldsymbol{v}_i^P - \boldsymbol{v}_j^P)(\boldsymbol{v}_i^P + \boldsymbol{v}_j^P)^\top + (\boldsymbol{v}_i^P + \boldsymbol{v}_j^P)(\boldsymbol{v}_i^P - \boldsymbol{v}_j^P)^\top)/2$.

5.G Proof of Theorem 5.5

Let $\rho := \max_{\boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathcal{C}} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$ be the diameter of $\mathcal{C}$, which is proportional to R. For both convex and non-convex cases, using the L-smoothness of f_i (and thus F), we have:

$$F(\bar{\boldsymbol{\theta}}_{t+1}) \leq F(\bar{\boldsymbol{\theta}}_t) + \sum_{i=1}^N \frac{\gamma_t}{N} \langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}_t \rangle + \frac{L\rho^2\gamma_t^2}{2}, \tag{5.125}$$

The middle term of the right hand side above can be bounded as follows:

$$\begin{aligned} \langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}_t \rangle \leq \rho \cdot & \left\| \nabla F(\bar{\boldsymbol{\theta}}_t) - \frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \right\| \\ & + \frac{1}{N} \left\langle \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j), R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}_t \right\rangle . \end{aligned} \tag{5.126}$$

As f_i is L-smooth, the first term in (5.126) can be bounded as

$$\left\| \nabla F(\bar{\boldsymbol{\theta}}_t) - \frac{1}{N} \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j) \right\| \leq \frac{L}{N^2} \cdot \sum_{j=1}^N \sum_{k=1}^N \|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_k^t\|. \tag{5.127}$$

Now, for all $j, k \in [N]$, we have

$$\|\bar{\boldsymbol{\theta}}_{t+1}^j - \bar{\boldsymbol{\theta}}_{t+1}^k\| \leq (1 - \gamma_t)\|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t^k\| + \gamma_t R \|\boldsymbol{a}_t^j(\boldsymbol{a}_t^j)^\top - \boldsymbol{a}_t^k(\boldsymbol{a}_t^k)^\top\|. \tag{5.128}$$

Using our choice of S_t and Proposition 5.1, we have $\|\boldsymbol{a}_t^j(\boldsymbol{a}_t^j)^\top - \boldsymbol{a}_t^k(\boldsymbol{a}_t^k)^\top\| = \mathcal{O}(1/t)$. Applying [Polyak(1987), Lemma 4 and 5], we can show that $\|\bar{\boldsymbol{\theta}}_t^j - \bar{\boldsymbol{\theta}}_t^k\| = \mathcal{O}(1/t)$ regardless of the choice of step size rule. We thus conclude that $\|\nabla F(\bar{\boldsymbol{\theta}}_t) - \sum_{j=1}^N \nabla f_j(\bar{\boldsymbol{\theta}}_t^j)/N\| = \mathcal{O}(1/t)$.

For the second term in (5.126), let $\hat{\boldsymbol{a}}_t := \mathsf{TopEV}(\boldsymbol{M}_t)$ and $\bar{\boldsymbol{a}}_t := \mathsf{TopEV}(-\nabla F(\bar{\boldsymbol{\theta}}_t))$. Since $\langle \boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top \rangle \geq \langle \boldsymbol{M}_t, \boldsymbol{a}\boldsymbol{a}^\top \rangle$ for all $\|\boldsymbol{a}\| = 1$, we can show:

$$\begin{aligned}\langle \boldsymbol{M}_t, \bar{\boldsymbol{\theta}}_t - R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top\rangle \leq \rho &\left\| \nabla F(\bar{\boldsymbol{\theta}}_t) - \frac{1}{N}\sum_{j=1}^N \nabla f_j(\boldsymbol{\theta}_t^j) \right\| \\ &+ R\langle \boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top - \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top\rangle + \langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top - \bar{\boldsymbol{\theta}}_t\rangle \ .\end{aligned} \tag{5.129}$$

The first term on the right hand side of the preceding relation is bounded by $\mathcal{O}(1/t)$ as discussed before. For the second term, applying the eigendecomposition $\boldsymbol{M}_t = \sum_{k=1}^d \lambda_k \boldsymbol{u}_k \boldsymbol{u}_k^\top$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ and the fact that $\hat{\boldsymbol{a}}_t = \boldsymbol{u}_1$, we can express $\langle \boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top - \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top\rangle$ as:

$$\langle \boldsymbol{M}_t, \hat{\boldsymbol{a}}_t(\hat{\boldsymbol{a}}_t)^\top - \boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top\rangle = \lambda_1 - \textstyle\sum_{k=1}^d \lambda_k(\boldsymbol{u}_i^\top \boldsymbol{a}_t^i)^2 \ . \tag{5.130}$$

Using our choice of L_t, P_t and Proposition 5.1, the output, $\boldsymbol{a}_t^i = \boldsymbol{v}_i^{P_t}$, of the DePM method, satisfies (5.60) with $\epsilon^2 = \mathcal{O}(1/t^2)$ and the right hand side of (5.130) can be upper bounded by $\mathcal{O}(1/t^2)$ with probability at least $1 - \tilde{c}/t^2$. Consequently, we can upper bound (5.126) as:

$$\langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\boldsymbol{a}_t^i(\boldsymbol{a}_t^i)^\top - \bar{\boldsymbol{\theta}}^t\rangle \leq \langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top - \bar{\boldsymbol{\theta}}_t\rangle + \mathcal{O}(1/t) \ . \tag{5.131}$$

Now, in the convex case where $\gamma_t = 2/(t+1)$, (5.125) and (5.131) imply that the following relation holds with probability at least $1 - (\pi^2/6)\tilde{c}$,

$$F(\bar{\boldsymbol{\theta}}_{t+1}) \leq F(\bar{\boldsymbol{\theta}}_t) + \gamma_t\langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top - \bar{\boldsymbol{\theta}}_t\rangle + \mathcal{O}(1/t^2) \ , \tag{5.132}$$

for all $t \geq 1$. Thus, $\langle \nabla F(\bar{\boldsymbol{\theta}}_t), R\bar{\boldsymbol{a}}_t(\bar{\boldsymbol{a}}_t)^\top\rangle \leq \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \boldsymbol{\theta}\rangle$ for all $\boldsymbol{\theta} \in \mathcal{C}$ since $\bar{\boldsymbol{a}}^t$ is the top eigenvector of $-\nabla F(\bar{\boldsymbol{\theta}}_t)$ and $\mathrm{Tr}(\boldsymbol{\theta}) = R$ if $\boldsymbol{\theta} \in \mathcal{C}$. Taking $\boldsymbol{\theta} = \boldsymbol{\theta}^\star$ and using the convexity of $F(\boldsymbol{\theta})$ yield

$$F(\bar{\boldsymbol{\theta}}_{t+1}) - F(\boldsymbol{\theta}^\star) \leq (1 - \gamma_t)(F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star)) + \mathcal{O}(1/t^2) \ , \tag{5.133}$$

for all $t \geq 1$, and the $\mathcal{O}(1/t)$ convergence of $F(\bar{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^\star)$ follows from [Polyak(1987), Lemma 4].

In the non-convex case, we have $\gamma_t = t^{-\alpha}$. Similarly, we can show that (5.125) and (5.131) lead to the following inequality which holds with probability at least $1 - (\pi^2/6)\tilde{c}$,

$$F(\bar{\boldsymbol{\theta}}_{t+1}) \leq F(\bar{\boldsymbol{\theta}}_t) - \gamma_t g_t + \mathcal{O}(1/t^{2\alpha}),\ \forall\ t \geq 1\ , \tag{5.134}$$

where $g_t := \max_{\boldsymbol{\theta} \in \mathcal{C}} \langle \nabla F(\bar{\boldsymbol{\theta}}_t), \bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta} \rangle \geq 0$. The relation (5.65) is then derived by summing the above inequality from $t = T/2 + 1$ to $t = T$.

6 Consensus-based Alternating Optimization

This chapter continues with our study on algorithms that run on networks. We focus on extensions of the popular *alternating optimization* (AO) algorithm and design its consensus-based counterpart for multi-agent optimization. In particular, we develop two consensus-based algorithms with applications to signal estimation and dictionary learning.

6.1 Context and Background

We consider the following multi-agent optimization problem:

$$\min_{\mathbf{x}, \{\mathbf{y}_i\}_{i=1}^N} \; F(\mathbf{x}, \mathbf{y}) := \sum_{i=1}^{N} \big(f_i(\mathbf{x}, \mathbf{y}_i) + h_i(\mathbf{y}_i) \big) \; , \tag{6.1}$$

over a connected network of N agents, where $\mathbf{y} := (\mathbf{y}_i)_{i=1}^N$. We shall work with the following settings regarding Problem (6.1):

- The function $h_i(\mathbf{y}_i)$ is convex and can possibly be non-smooth.
- The function $f_i(\mathbf{x}, \mathbf{y}_i)$ is continuously differentiable (possibly non-convex) with respect to (w.r.t.) both $\mathbf{x}$ and $\mathbf{y}_i$.

As we are interested in the multi-agent optimization setting, here both $h_i(\cdot), f_i(\cdot, \cdot)$ are *private functions* that are known to agent i only. In particular, the optimization variables $\mathbf{x} \in \mathbb{R}^m$ and $\mathbf{y}_i \in \mathbb{R}^n$ can be treated as the common variable and the private variable, respectively. Applications of the formulation (6.1) include the popular matrix factorization problems, and we shall discuss the applications in latter part of this chapter when the specific algorithms are developed.

With a slight modification such as constraining $\mathbf{x}, \mathbf{y}_i$ to a compact set (e.g., a ball with a sufficiently large radius), Problem (6.1) can be viewed as a special case of (5.1) studied in the preceding chapter. Naturally, one can apply the DeFW or DPG algorithm to tackle the problem. However, as (6.1) is *non-convex* in general, a more commonly used technique is to apply the *alternating optimization* (AO) algorithm. The AO algorithm is motivated by the *separability* of the objective function in (6.1). In particular, the basic idea is to *alternate* between the optimization of the two variables $\mathbf{x}$ and $\{\boldsymbol{y}_i\}_{i=1}^N$ during the successive iterations, *i.e.,* let $t \in \mathbb{N}$ be the iteration number, the AO algorithm updates as follows:

$$\mathbf{x}^{t+1} = \arg\min_{\mathbf{x}} \; F(\mathbf{x}, \mathbf{y}^t), \qquad \mathbf{y}^{t+1} = \arg\min_{\mathbf{y}} \; F(\mathbf{x}^{t+1}, \mathbf{y}) \; . \tag{6.2}$$

The preceding procedure is attractive when the two sub-problems above can be solved easily, e.g., when $F(\cdot, \cdot)$ is bi-convex but not convex in both of the variables. Notice that the exact minimization in (6.2) is often replaced by an inexact optimization step such as gradient step or Newton step; see [Nesterov(2012), Yuan *et al.*(2012)]. Importantly, as studied in [Grippo and Sciandrone(2000), Razaviyayn *et al.*(2013), Hong *et al.*(2017)], under the *centralized* setting, the AO algorithm has often nice convergence properties (*i.e.,* convergence to a stationary point of (6.1)) and fast practical convergence speed. For example, when applied to some non-convex matrix factorization problems, the AO algorithm finds the *globally optimal* solution when the initial point for the algorithm is chosen judiciously [Arora *et al.*(2015)].

To this end, we observe that the '$\mathbf{y}$-update' in (6.2) can be solved in a decentralized manner since the objective function is decomposable with respect to the $\mathbf{y}$ variables, *i.e.,* we can obtain $\mathbf{y}^{t+1}$ by tackling $\mathbf{y}_i^{t+1} = \arg\min_{\mathbf{y}_i} \big(f_i(\mathbf{x}^{t+1}, \mathbf{y}_i) + h_i(\mathbf{y}_i) \big)$ independently at the ith agent. Instead, our challenge lies on the '$\mathbf{x}$-update' in (6.2), which deals with a common variable that every agent over the network has to agree on (or reach a consensus about). In the rest of this chapter, we propose two algorithms for tackling the $\mathbf{x}$ update in a decentralized manner while ensuring convergence of the resultant AO algorithm. Specifically, Section 6.2 proposes a consensus-based AO algorithm for least square problems (C-AOLS) to obtain an inexact solution on the $\mathbf{x}$ update using average consensus, and Section 6.3 proposes an EXTRA-AO algorithm that applies a decentralized gradient method on $\mathbf{x}$. We also discuss their motivating applications. Lastly, Section 6.4 presents result of numerical experiments to verify the efficacy of the proposed algorithms.

Notations. We use the notations defined in Section 2.1 on networks with additional conditions as follows. In particular, the network which we run our algorithms on is *undirected.* It has N nodes and the weighted adjacency matrix is non-negative, doubly stochastic, *i.e.,* such that $\boldsymbol{A}^\top \mathbf{1} = \boldsymbol{A}\mathbf{1} = \mathbf{1}$. Its second largest singular value, $\sigma_2(\boldsymbol{A})$, is strictly less than one.

6.2 Consensus-based AO Algorithm for Least Square Problems

Our first consensus-based AO algorithm is motivated by the least square estimation problem with nuisance parameters. We consider objective functions that take the form:

$$f_i(\mathbf{x}, \mathbf{y}_i) = \|\boldsymbol{g}_i(\mathbf{x}, \mathbf{y}_i)\|_2^2, \quad h_i(\mathbf{y}_i) = \mathcal{I}_{\mathcal{B}_i}(\mathbf{y}_i) \ , \tag{6.3}$$

where $\boldsymbol{g}_i : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^{p_i}$ is an affine function in the first argument $\mathbf{x}$, given as:

$$\boldsymbol{g}_i(\mathbf{x}, \mathbf{y}_i) = \mathbf{H}_i(\mathbf{y}_i)\mathbf{x} - \boldsymbol{\zeta}_i(\mathbf{y}_i) \ , \tag{6.4}$$

and $\mathcal{I}_{\mathcal{B}_i}(\cdot)$ is an indicator function for the convex set $\mathcal{B}_i$ such that $\mathcal{I}_{\mathcal{B}_i}(\mathbf{y}_i) = 0$ if $\mathbf{y}_i \in \mathcal{B}_i$ and it is ∞ for otherwise. Observe that when $\mathbf{y}_i$ are fixed for all i, the optimization problem is *convex* in $\mathbf{x}$ and it admits a closed form solution:

$$\mathbf{x}^\star(\mathbf{y}) = \Big(\sum_{i=1}^N \mathbf{H}_i(\mathbf{y}_i)^\top \mathbf{H}_i(\mathbf{y}_i) \Big)^{-1} \Big(\sum_{i=1}^N \mathbf{H}_i(\mathbf{y}_i)^\top \boldsymbol{\zeta}_i(\mathbf{y}_i) \Big) \ . \tag{6.5}$$

Importantly, we observe that for all i, the matrix-matrix and matrix-vector products:

$$\boldsymbol{\mathcal{H}}_i(\mathbf{y}_i) := \mathbf{H}_i(\mathbf{y}_i)^\top \mathbf{H}_i(\mathbf{y}_i), \quad \boldsymbol{\theta}_i(\mathbf{y}_i) := \mathbf{H}_i(\mathbf{y}_i)^\top \boldsymbol{\zeta}_i(\mathbf{y}_i) \ , \tag{6.6}$$

can be computed locally at agent i using his/her local variables. Moreover, $\mathbf{x}^\star(\mathbf{y})$ can be computed using *averages* of the quantities above, *i.e.,*

$$\mathbf{x}^\star(\mathbf{y}) = \Big(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\mathcal{H}}_i(\mathbf{y}_i)\Big)^{-1}\Big(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\theta}_i(\mathbf{y}_i)\Big) = \overline{\boldsymbol{\mathcal{H}}}(\mathbf{y})^{-1}\bar{\boldsymbol{\theta}}(\mathbf{y}) \;, \tag{6.7}$$

where we have defined $\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}) := (1/N)\sum_{i=1}^{N} \boldsymbol{\mathcal{H}}_i(\mathbf{y}_i)$ and $\bar{\boldsymbol{\theta}}(\mathbf{y}) := (1/N)\sum_{i=1}^{N} \boldsymbol{\theta}_i(\mathbf{y}_i)$ as the wanted averages.

The observation above motivates us to perform the required '$\mathbf{x}$-update' in a decentralized manner by applying average consensus to approximate $\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y})$ and $\bar{\boldsymbol{\theta}}(\mathbf{y})$. In particular, the proposed C-AOLS algorithm can be summarized in Algorithm 6.1. As seen, Line 3 to 6 in the psuedo code performs an S-steps average consensus on the given network to compute the averages $\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)$ and $\bar{\boldsymbol{\theta}}(\mathbf{y}^t)$, while the remaining steps simply perform the standard AO steps based on the approximate averages. Notice that we have applied a projected gradient update for the '$\mathbf{y}$-update' stage in the AO algorithm, since a closed form solution for the respective update may not be available.

6.2.1 Convergence Analysis

To analyze the convergence of the C-AOLS algorithm, we first show the following result regarding the consensus step in the algorithm:

Proposition 6.1 *Let $\mathcal{B} := \mathcal{B}_1 \times \cdots \mathcal{B}_N$ and suppose that*

$$\max_{\mathbf{y}\in\mathcal{B}} \Big\| \sum_{i=1}^{N} \mathbf{H}_i(\mathbf{y}_i)^\top \mathbf{H}_i(\mathbf{y}_i) \Big\| \cdot \max_{\mathbf{y}\in\mathcal{B}} \Big\| \Big(\sum_{i=1}^{N} \mathbf{H}_i(\mathbf{y}_i)^\top \mathbf{H}_i(\mathbf{y}_i) \Big)^{-1} \Big\| \cdot \sigma_2(\boldsymbol{A})^S < 1 \;, \tag{6.12}$$

then the iterates computed in Algorithm 6.1 satisfy:

$$\textstyle\sum_{i=1}^{N} \|\mathbf{x}^\star(\mathbf{y}^t) - \mathbf{x}_i^{t+1}\| \leq \tilde{C}_0 \cdot \sigma_2(\boldsymbol{A})^S \;, \tag{6.13}$$

$$\textstyle\sum_{i=1}^{N} \|\mathbf{x}^\star(\mathbf{y}^t) - (1/N)\sum_{j=1}^{N} \mathbf{x}_j^{t+1}\| \leq \tilde{C}_1 \cdot \sigma_2(\boldsymbol{A})^S \;, \tag{6.14}$$

Algorithm 6.1 Consensus-based AO algorithm for Least Square Problems (C-AOLS).

```
Initialize: $\{\mathbf{x}_i^0\}_{i=1}^N$, $\{\mathbf{y}_i^0\}_{i=1}^N$, and parameter $S$;
for $t = 1, 2, ...$ do
    Consensus step: we set
        $\overline{\boldsymbol{\mathcal{H}}}_i^{0,t} = \mathbf{H}_i(\mathbf{y}_i^t)^\top \mathbf{H}_i(\mathbf{y}_i^t)$ and $\boldsymbol{\theta}_i^{0,t} = \mathbf{H}_i(\mathbf{y}_i^t)^\top \boldsymbol{\zeta}_i(\mathbf{y}_i^t),\ \forall\, i \in [N]\,.$   (6.8)
    for $\ell = 0, 1, ..., S-1$ do
        $\overline{\boldsymbol{\mathcal{H}}}_i^{\ell+1,t} = \sum_{j=1}^N A_{ij} \cdot \overline{\boldsymbol{\mathcal{H}}}_j^{\ell,t}$ and $\bar{\boldsymbol{\theta}}_i^{\ell+1,t} = \sum_{j=1}^N A_{ij} \cdot \bar{\boldsymbol{\theta}}_j^{\ell,t},\ \forall\, i \in [N]\,.$   (6.9)
    end for
    AO step: for all $i = 1, ..., N$, agent $i$ updates its own copies of $\mathbf{x}$ and $\mathbf{y}_i$:
        $\mathbf{x}_i^{t+1} = \left(\overline{\boldsymbol{\mathcal{H}}}_i^{S,t}\right)^{-1} \bar{\boldsymbol{\theta}}_i^{S,t}\,,$   (6.10)
        $\mathbf{y}_i^{t+1} = \mathcal{P}_{\mathcal{B}_i}\Big(\mathbf{y}_i^t - \beta \cdot \nabla_{\mathbf{y}_i} f_i(\mathbf{x}_i^{t+1}, \mathbf{y}_i^t)\Big)\,,$   (6.11)
    where $\mathcal{P}_{\mathcal{B}_i}(\cdot)$ is the projection operator onto $\mathcal{B}_i$ and $\beta > 0$ is a step size.
end for
Return: an approximate solution to (6.1) — $\{\mathbf{x}_i^t\}_{i=1}^N$, $\{\mathbf{y}_i^t\}_{i=1}^N$.
```

$$\textstyle\sum_{i=1}^N \|\mathbf{x}_i^{t+1} - (1/N)\sum_{j=1}^N \mathbf{x}_j^{t+1}\| \leq \tilde{C}_2 \cdot \sigma_2(\boldsymbol{A})^S\,, \tag{6.15}$$

where $\mathbf{x}^\star(\mathbf{y}^t)$ *was defined in* (6.7)*, and* $\tilde{C}_0, \tilde{C}_1, \tilde{C}_2$ *are some constants that depend on the left hand side of* (6.12)*.*

The results above show that the approximation errors decay exponentially with S, the number of consensus exchanges used per iteration. In fact, the analysis simply follows from standard convergence result for the average consensus protocol, as we present in Appendix 6.A. However, Proposition 6.1 is crucial to establishing the asymptotic convergence of Algorithm 6.1.

Theorem 6.1 *Let* $(\mathbf{x}^\star, \mathbf{y}^\star)$ *be a local minimum to* (6.1)*. Suppose that the global function*

$F(\cdot,\cdot)$ is m_o-strongly convex and M_o-smooth in the neighborhood $\mathcal{N}_{R^\star}(\mathbf{x}^\star,\mathbf{y}^\star)$ with radius $R^\star$. Moreover, each of $f_i(\cdot,\cdot)$ is Lipschitz continuous with the constant L_o. Suppose $\beta < 1/M_o$, $B := \max_t \|(\overline{\mathbf{x}}^t,\mathbf{y}^t) - (\mathbf{x}^\star,\mathbf{y}^\star)\| \leq R^\star$, then the C-AOLS algorithm generates iterates which satisfy:

$$\lim_{t\to\infty} \|(\overline{\mathbf{x}}^t,\mathbf{y}^t) - (\mathbf{x}^\star,\mathbf{y}^\star)\|^2 \leq \rho\big(\sigma_2(\boldsymbol{A})\big) \ , \tag{6.16}$$

where $\overline{\mathbf{x}}^k := (1/N)\sum_{i=1}^N \mathbf{x}_i^t$ and

$$\begin{aligned}&\rho\big(\sigma_2(\boldsymbol{A})\big) := \\ &\frac{2}{m_o}\left((L_o\tilde{C}_1 + BM_o\tilde{C}_0)\sigma_2(\boldsymbol{A})^S + \sqrt{\frac{(\tilde{C}_0+\tilde{C}_2)\sigma_2(\boldsymbol{A})^S}{1/(18B^2L_oM_o)}}\right) \ ,\end{aligned} \tag{6.17}$$

where the constants $\tilde{C}_0, \tilde{C}_1, \tilde{C}_2$ are the same constants used in Proposition 6.1.

Note that the upper bound satisfies $\rho(\sigma_2(\boldsymbol{A})) = \mathcal{O}(\sigma_2(\boldsymbol{A})^{S/2})$. The proof is provided in Appendix 6.B, which is based on studying the error dynamics of the C-AOLS algorithm as a second order dynamical system. In fact, Theorem 6.1 implies that if the iterates of Algorithm 6.1 stay close enough to a local minimum, then the iterates converge to an approximate of that local minimum, where the approximation accuracy improves exponentially with S, *i.e.,* the number of average consensus used per iteration of the algorithm.

Notice that Theorem 6.1 does not assume F to be convex. However, the strong convexity assumption on $F(\cdot,\cdot)$ around a local minimum may appear restrictive at first. However, our numerical results indicate that Theorem 6.1 can accurately predict the performance of the algorithm applied to the problem (6.1).

6.2.2 State Estimation with Asynchronous Measurements

As a motivating example, we observe that Problem (6.1) with the above setting encompasses a practical problem pertaining to state estimation with asynchronous measurements. Our scenario of interest is a sensor network that captures the continuous-time sensor field $\boldsymbol{x}_c(t) \in \mathbb{R}^n$ using N sensors. Assume that the signal is band-limited by $1/(2T_s)$ Hz. We focus on the case when the sensors are taking memoryless, asynchronous and sub-Nyquist measurements on $\boldsymbol{x}_c(t)$. Specifically, the kth sample recorded at the ith sensor is:

$$\boldsymbol{\zeta}_i[k] = \mathbf{H}_i \boldsymbol{x}_c((kA_i - b_i)T_s) + \boldsymbol{w}_i[k] \ , \tag{6.18}$$

where $A_i \geq 1, A_i \in \mathbb{Z}$ is the down-sampling factor of the ith sensor and $b_i \in \mathbb{R}$ is the time offset in sampling, $\boldsymbol{w}_i[k] \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ is an additive noise and $\mathbf{H}_i \in \mathbb{R}^{m \times n}$ represents the measurement matrix for the pth sensor. Examples of sensing systems that can be modeled by (6.18) include wide area measurement systems (WAMS) for power system state estimation (PSSE) [Li and Scaglione(2013)]. As $\boldsymbol{x}_c(t)$ is bandlimited, it suffices to estimate the Nyquist-rate samples of $\boldsymbol{x}_c(t)$, *i.e.*, $\boldsymbol{x}[k] := \boldsymbol{x}_c(kT_s)$, for our task. We assume that the down-sampling factor A_i is known, while the time offset b_i is an unknown nuisance parameter, and we set $b_1 = 0$ without loss of generality.

Our next endeavor is to show that the regression problem under model (6.18) with asynchronous and sub-Nyquist sampling can be cast into a special case of the least square problem with nuisance parameter. Note that from (6.18), it is impossible to infer a single sample $\boldsymbol{x}[k]$ by taking one snapshot of the measured signal $\{\boldsymbol{\zeta}_i[k]\}_{i=1}^N$, as the latter also depends on the other samples $\{\boldsymbol{x}[j]\}_{j \neq k}$. We resort to an *offline processing* scheme that performs a batch regression of the Fourier series of a data stream. In particular, we observe that the frequency-domain equivalent model to (6.18) admits the following representation:

Observation 6.1 *Let $\boldsymbol{x}_c(t)$ be bandlimited by $1/(2T_s)$ Hz and define the frequency map:*

$$\Omega_{A_i}^a(\omega) := \big(\frac{\omega}{A_i} - \frac{a}{A_i} 2\pi\big) \bmod (-\pi, \pi] \ . \tag{6.19}$$

The measurement model (6.18) *is equivalent to the following:*

$$\mathbf{Z}_i(e^{j\omega}) = \frac{1}{A_i} \sum_{a=0}^{A_i - 1} e^{-jb_i \Omega_{A_i}^a(\omega)} \mathbf{H}_i \mathbf{X}\left(e^{j\Omega_{A_i}^a(\omega)}\right) + \mathbf{V}_i(e^{j\omega}) \ , \tag{6.20}$$

where $\mathbf{Z}_i(e^{j\omega})$, $\mathbf{V}_i(e^{j\omega})$ *and* $\mathbf{X}(e^{j\omega})$ *are the discrete-time Fourier transform (DTFT) of* $\boldsymbol{\zeta}_i[k]$, $\boldsymbol{v}_i[k]$ *and* $\boldsymbol{x}[k]$*, respectively.*

A key to verifying the above observation is to decompose $\boldsymbol{x}[k]$ into its polyphase components and study the spectrum of the down-sampled signal; see [Vaidyanathan(1993)].

Observation 6.1 shows that the observation on $\mathbf{X}(e^{j\omega})$ is subjected to linear transformation, linear phase shift and aliasing. These effects, when combined, can be viewed as a linear transformation on $\mathbf{X}(e^{j\omega})$ with given time offset b_i. To see this, we need to ensure that the measured samples are obtained at the same sampling rate. It can be achieved by generating the following samples from $\boldsymbol{\zeta}_i[k]$:

$$\boldsymbol{\zeta}_i^q[k] = \boldsymbol{\zeta}_i[Q_i k - q],\ q = 0, 1, ..., Q_i - 1\ , \tag{6.21}$$

where $Q_i := A/A_i$ and $A := \mathsf{LCM}\{A_1, ..., A_N\}$. These samples are equivalent to those sampled at $1/A$ of the Nyquist rate. Next, by noting that $\Omega_A^a((-\pi, \pi])$ is disjoint with $\Omega_A^b((-\pi, \pi])$ for $a \neq b$, we can define the following extended spectrum:

$$\tilde{\mathbf{X}}(e^{j\omega}) = \big[\mathbf{X}\big(e^{j\Omega_A^0(\omega)}\big)^\top\ \ldots\ \mathbf{X}\big(e^{j\Omega_A^{A-1}(\omega)}\big)^\top\big]\ , \tag{6.22}$$

where $A := \mathsf{LCM}\{A_1, ..., A_N\}$; and the extended matrix:

$$\tilde{\mathbf{H}}_i(b_i, e^{j\omega}) = \frac{1}{A}\big[e^{-jb_i\Omega_A^0(\omega)}\ \ldots\ e^{-jb_i\Omega_A^{A-1}(\omega)}\big] \otimes \mathbf{H}_i\ . \tag{6.23}$$

Using Eq. (6.22) and (6.23), the observed spectrum in (6.20) can be simplified as:

$$\mathbf{Z}_i^q(e^{j\omega}) = \tilde{\mathbf{H}}_i(b_i - qQ_i, e^{j\omega})\tilde{\mathbf{X}}(e^{j\omega}) + \mathbf{V}_i^q(e^{j\omega}), \tag{6.24}$$

which is an affine function in $\tilde{\mathbf{X}}(e^{j\omega})$.

We discuss how the spectrum $\mathbf{Z}_i^q(e^{j\omega})$ can be obtained from a finite number of measurements. Here, we accrue $Q_i T$ samples at sensor i and we can approximate $\mathbf{Z}_i^q(e^{j\omega})$ by the following F-point discrete Fourier transform (DFT) spectrum:

$$\mathbf{Z}_i^q[f] = \sum_{m=0}^{T-1} \boldsymbol{\zeta}_i^q[m] e^{-j\omega_f m},\ f = 0, ..., F-1\ , \tag{6.25}$$

where $\omega_f \triangleq 2\pi(f - F + 1)/F$ and $F \geq T$ is required. In this way, the sequence $\{\boldsymbol{x}[k]\}_{k=0}^{AT-1}$ can be inferred from the collection of spectrum $\{\mathbf{Z}_i^q[f]\}_{i,f}$. Finally, from (6.24) we can derive the following regression problem for estimating the desired sequence $\{\boldsymbol{x}[k]\}_{k=0}^{AT-1}$:

$$\min_{\{\mathbf{x}[k]\}_{k=0}^{AT-1}, \{b_i\}_{i=2}^{N}} \ \sum_{i=1}^{N} f_i(\{\boldsymbol{x}[k]\}_{k=0}^{AT-1}, b_i) \ \text{ s.t. } \ b_i \in \mathcal{B}_i,\ \forall\ i \in [N]\ , \tag{6.26}$$

where

$$f_i(\{\boldsymbol{x}[k]\}_{k=0}^{AT-1}, b_i) := \sum_{q=0}^{Q_i-1} \sum_{f=0}^{F-1} \left\| \mathbf{Z}_i^q[f] - \tilde{\mathbf{H}}_i(b_i - qQ_i, e^{j\omega_f}) \tilde{\mathbf{X}}(e^{j\omega_f}) \right\|_2^2 , \tag{6.27}$$

and $\tilde{\mathbf{X}}(e^{j\omega_f})$ is defined in (6.22), where $\mathbf{X}(e^{j\omega_f}) = \sum_{m=0}^{AT-1} \boldsymbol{x}[m] e^{-j\omega_f m}$ is a linear function of $\{\boldsymbol{x}[k]\}_{k=0}^{AT-1}$. We thus observe that the regression problem (6.26) is a special case of the problem (6.1) considered in this section, and we can apply Algorithm 6.1. Note, however, that the model is exact only in the limit when $F \to \infty$, thus there is a trade-off between the complexity of solving (6.26) and the quality of the approximation, that improves when the number of DFT points F is large.

6.3 EXTRA-AO Algorithm

The C-AOLS algorithm studied in the last chapter is specialized to the least square problems with local nuisance parameters. This section considers a consensus-based AO algorithm suitable for problems given in the general form of (6.1). Our main idea is to tackle the '$\mathbf{x}$-update' using an iterative scheme similar to the DPG algorithm.

However, we observe that combining the plain DPG algorithm with AO may not lead to a converging algorithm. To see this, let us consider the following algorithm:

$$\mathbf{x}_i^t = \sum_{j=1}^N A_{ij}\mathbf{x}_j^{t-1} - \alpha_t \nabla_{\mathbf{x}} f_i\Big(\textstyle\sum_{j=1}^N A_{ij}\mathbf{x}_j^{t-1}, \mathbf{y}_i^{t-1}\Big),\ \forall\, i \in [N]\ , \tag{6.28}$$

$$\mathbf{y}_i^t = \mathbf{prox}_{\beta_t h_i(\cdot)}\big(\mathbf{y}_i^{t-1} - \beta_t \nabla_{\mathbf{y}} f_i(\mathbf{x}_i^t, \mathbf{y}_i^{t-1})\big),\ \forall\, i \in [N]\ , \tag{6.29}$$

where $\alpha_t, \beta_t > 0$ are the step sizes to be specified later. In (6.29), the proximal operator is defined as:

$$\mathbf{prox}_{\beta_t h_i(\cdot)}(\mathbf{y}) := \arg\min_{\mathbf{z}}\ \frac{1}{2}\|\mathbf{y} - \mathbf{z}\|_2^2 + \beta_t h_i(\mathbf{z})\ . \tag{6.30}$$

A popular example is $h_i(\mathbf{y}_i) = \rho \cdot \|\mathbf{y}_i\|_1$ with $\rho > 0$. In this case, the proximal operator is equivalent to the soft thresholding operator [Beck and Teboulle(2009)] which can be given in closed form. Note that the algorithm is simply derived by combining the DPG algorithm with a proximal gradient update for '**y**-update'. It is known in the literature as the (adapt-then-combine) 'ATC-AO' algorithm and was studied in [Chainais and Richard(2013)].

However, the recursions (6.28) and (6.29) are not guaranteed to converge to a stationary point of (6.1) in general. In particular, under a fixed step size rule, *i.e.,* $\alpha_t = \alpha$ for all t, it can be shown that the recursion (6.28) and (6.29) may converge to a solution such that $\mathbf{x}_i^t \neq \mathbf{x}_j^t$, *i.e.,* consensus is not reached; on the other hand, when the step size α_t is diminishing, our numerical experiments suggest that the algorithm may not converge to a stationary point of (6.1) at all. An example of lack of convergence is the numerical simulations shown in Figure 6.4 of Section 6.4.2, which illustrates that the ATC-AO with a fixed step size does not lead to a solution that reaches consensus.

For this reason we leverage an alternative idea proposed in [Shi *et al.*(2015)], called the exact first order algorithm (EXTRA). The algorithm is described by (6.31) together with the pseudo code of EXTRA-AO in Algorithm 6.2. An important feature of the EXTRA update is that a fixed step size is used throughout the algorithm, leading to a distributed algorithm with linear convergence rate. As seen in [Shi *et al.*(2015)], this strategy achieves consensus and optimality simultaneously when applied to convex optimization problems.

Algorithm 6.2 EXTRA-AO algorithm for (6.1).

```
**Initialize:** $\{\mathbf{x}_i^0\}_{i=1}^N$, $\{\mathbf{y}_i^0\}_{i=1}^N$;
**for** $t = 1, 2, ...$ **do**
  **for** $i = 1, 2, ..., N$ **do**
    Agent $i$ computes the following EXTRA update for $\mathbf{x}_i$:
      $$\mathbf{x}_i^t = \begin{cases} \sum_{j=1}^{N} A_{ij}\mathbf{x}_j^{t-1} - \alpha \nabla_{\mathbf{x}} f_i(\mathbf{x}_i^{t-1}, \mathbf{y}_i^{t-1}), & \text{if } t = 1 \text{ ,} \\ \mathbf{x}_i^{t-1} + \sum_{j=1}^{N} A_{ij}\mathbf{x}_j^{t-1} - \alpha \nabla_{\mathbf{x}} f_i(\mathbf{x}_i^{t-1}, \mathbf{y}_i^{t-1}) & \\ \qquad - \sum_{j=1}^{N} \tilde{A}_{ij}\mathbf{x}_j^{t-2} + \alpha \nabla_{\mathbf{x}} f_i(\mathbf{x}_i^{t-2}, \mathbf{y}_i^{t-2}), & \text{if } t > 1 \text{ ,} \end{cases} \tag{6.31}$$
    where $\alpha > 0$ is a fixed step size and $\tilde{\boldsymbol{A}} = (\boldsymbol{I} + \boldsymbol{A})/2$. Notice that $\tilde{\boldsymbol{A}}$ can also take a different form with more relaxed conditions, see [Shi *et al.*(2015)].
    Agent $i$ computes the following update for $\mathbf{y}_i$:
      $$\mathbf{y}_i^t = \mathbf{prox}_{\beta h_i(\cdot)}\Big(\mathbf{y}_i^{t-1} - \beta \nabla_{\mathbf{y}} f_i(\mathbf{x}_i^t, \mathbf{y}_i^{t-1})\Big) \text{ .} \tag{6.32}$$
  **end for**
**end for**
**Return:** approximate stationary solution to (6.1) — $\{\mathbf{x}_i^k\}_{i=1}^N$, $\{\mathbf{y}_i^k\}_{i=1}^N$.
```

Note that, in (6.31), $\tilde{\boldsymbol{A}} = (\boldsymbol{I} + \boldsymbol{A})/2$ has the same sparsity as $\boldsymbol{A}$, therefore similar to the ATC strategy described previously, the EXTRA update can also be computed via local computations and information exchange with the neighboring agents.

We now discuss about the EXTRA update on the $\mathbf{x}$'s side. The EXTRA step (6.31) combines both consensus and gradient descent in a single step, where the optimization variables from the *previous two iterations* are required. In fact, the latter features a similar *gradient tracking* idea used in the consensus based DeFW algorithm [cf. (5.28)] of the previous chapter; see [Nedić *et al.*(2016)] for details about the said interpretation.

6.3.1 Convergence Analysis

Next, we analyze the convergence of the EXTRA-AO algorithm. To facilitate our discussion, let us introduce the following variables/functions:

$$\boldsymbol{x}^t := [\mathbf{x}_1^t \ \mathbf{x}_2^t \ \cdots \ \mathbf{x}_N^t]^\top, \quad \boldsymbol{y}^t := [\mathbf{y}_1^t \ \mathbf{y}_2^t \ \cdots \ \mathbf{y}_N^t]^\top, \tag{6.33}$$

$$\mathbf{f}(\boldsymbol{x}, \boldsymbol{y}) := [f_1(\mathbf{x}_1, \mathbf{y}_1) \ \cdots \ f_N(\mathbf{x}_N, \mathbf{y}_N)]^\top \ , \tag{6.34}$$

$$\nabla_{\boldsymbol{x}} \mathbf{f}(\boldsymbol{x}, \boldsymbol{y}) := [\nabla_{\mathbf{x}} f_1(\mathbf{x}_1, \mathbf{y}_1) \ \cdots \ \nabla_{\mathbf{x}} f_N(\mathbf{x}_N, \mathbf{y}_N)]^\top \ , \tag{6.35}$$

$$F(\boldsymbol{x}, \boldsymbol{y}) := \textstyle\sum_{i=1}^N f_i(\mathbf{x}_i, \mathbf{y}_i) = \mathbf{1}^\top \mathbf{f}(\boldsymbol{x}, \boldsymbol{y}) \ . \tag{6.36}$$

Notice that both $\mathbf{x}_i$ and $\mathbf{y}_i$ are given as column vectors, therefore $\boldsymbol{x}^t \in \mathbb{R}^{N \times m}$ and $\boldsymbol{y}^t \in \mathbb{R}^{N \times n}$ for all t. To this end, a sufficient condition for EXTRA-AO to reach a stationary point of (6.1) is given as follows:

Proposition 6.2 *Let* $\text{null}\{\boldsymbol{I} - \boldsymbol{A}\} = \text{span}\{\mathbf{1}\}$. *Suppose that the sequence* $\{(\boldsymbol{x}^t, \boldsymbol{y}^t)\}_{t \geq 1}$ *generated by EXTRA-AO converges to a point* $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$*, then* $(\overline{\mathbf{x}}^\infty, \boldsymbol{y}^\infty)$ *is a stationary point to problem* (6.1)*, where* $\overline{\mathbf{x}}^t := (1/N)\mathbf{1}^\top \boldsymbol{x}^t$*, and* $\lim_{t \to \infty} \mathbf{x}_i^t = \overline{\mathbf{x}}^\infty$ *for all* $i \in [N]$.

Importantly, the proposition above shows that as long as the sequence, $\{(\boldsymbol{x}^t, \boldsymbol{y}^t)\}_{t=1}^\infty$, produced by the EXTRA-AO algorithm is convergent, the limit point is both *consensual* and a stationary point to (6.1). The proof can be found in Appendix 6.C.

However, verifying that the sequence converges to a *unique* limit is non-trivial as the algorithm uses a constant step size. A partial analysis is presented below which provides insights regarding the choice of step size required. We have:

Proposition 6.3 *Suppose that each of the functions* $f_i(\mathbf{x}_i, \mathbf{y}_i)$ *has a Lipschitz continuous gradient with constants* L_x*,* L_y *with respect to* $\mathbf{x}_i$*,* $\mathbf{y}_i$*, respectively, for all* $i \in [N]$*. If the step sizes* α, β *in the EXTRA-AO algorithm satisfy*

$$0 < \alpha < (2\lambda_{min}(\tilde{\boldsymbol{A}})/L_x), \ 0 < \beta < (1/L_y) \ , \tag{6.37}$$

then the following inequalities hold at each iteration:

$$F(\boldsymbol{x}^{t+1}, \boldsymbol{y}^t) - F(\boldsymbol{x}^t, \boldsymbol{y}^t) \leq -\delta \|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_F^2 - \frac{1}{\alpha} \left\langle (\tilde{\boldsymbol{A}} - \boldsymbol{A}) \sum_{\ell=0}^{t+1} \boldsymbol{x}^\ell, \boldsymbol{x}^{t+1} - \boldsymbol{x}^t \right\rangle , \tag{6.38}$$

$$F(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1}) - F(\boldsymbol{x}^{t+1}, \boldsymbol{y}^t) \leq -\frac{1}{2} \|\boldsymbol{y}^t - \boldsymbol{y}^{t+1}\|_F^2 , \tag{6.39}$$

where we have defined the constant $\delta := (\lambda_{min}(\tilde{\boldsymbol{A}})/\alpha - L_x/2) > 0$.

Proposition 6.3 provides a guideline for choosing the step size for EXTRA-AO. Moreover, suppose that we are in a situation where the inner product of (6.38) is non-negative or vanishing. Then the objective values along the iterates of the EXTRA-AO algorithm are non-increasing, *i.e.,*

$$\cdots \leq F(\boldsymbol{x}^{t+1}, \boldsymbol{y}^{t+1}) \leq F(\boldsymbol{x}^{t+1}, \boldsymbol{y}^t) \leq F(\boldsymbol{x}^t, \boldsymbol{y}^t) \leq \cdots \tag{6.40}$$

Consequently, if the optimal objective value to (6.1) is bounded below, $F(\boldsymbol{x}^t, \boldsymbol{y}^t)$ converges to a unique value as $t \to \infty$ and we have

$$\lim_{t\to\infty} \|\boldsymbol{x}^t - \boldsymbol{x}^{t+1}\| = \lim_{t\to\infty} \|\boldsymbol{y}^t - \boldsymbol{y}^{t+1}\| = 0 , \tag{6.41}$$

and the sufficient conditions required by Proposition 6.2 are satisfied. Verifying the requirement above that the latter inner product in (6.38) is non-negative or vanishing is an open problem.

6.3.2 Decentralized Dictionary Learning

Our study on the EXTRA-AO algorithm is motivated by the dictionary learning (DL) problem:

$$\min_{\mathbf{X},\mathbf{Y}} \frac{1}{2} \|\mathbf{S} - \mathbf{X}\mathbf{Y}\|_F^2 + \lambda \|\mathbf{Y}\|_1 + \sum_{\ell=1}^{M} \gamma_\ell \|\mathbf{X}_{:,\ell}\|_2^2 , \tag{6.42}$$

where $[\mathbf{X}]_{:,\ell}$ denotes the ℓth column in matrix $\mathbf{X}$. The regularization terms $\|\mathbf{Y}\|_1$ and $\|\mathbf{X}_{:,\ell}\|_2^2$ are introduced respectively to promote sparsity and to ensure that the solution

$\mathbf{X}$ is bounded. In the data model underlying (6.42), the matrix $\mathbf{S} \in \mathbb{R}^{m\times M}$ contains M columns of training data, in which each of them is assumed to be a linear combination of the column vectors in the dictionary $\mathbf{X} \in \mathbb{R}^{m\times n}$, with the coefficients contained in the columns of $\mathbf{Y} \in \mathbb{R}^{n\times M}$. It is assumed that $\mathbf{S}$ is sparse such that each column of $\mathbf{Y}$ is formed by combining only a few columns from the dictionary $\mathbf{X}$, *i.e.,* we have $\mathbf{S} = \mathbf{X}\mathbf{Y}$.

As an application for the EXTRA-AO algorithm developed, we consider a decentralized implementation of (6.42). In particular, each of the N agents collects training data $\mathbf{S}_i$ independently while all agents want to learn a common dictionary $\mathbf{X}$. We write $\mathbf{S} = [\mathbf{S}_1 \; \mathbf{S}_2 \; \cdots \; \mathbf{S}_N]$ and $\mathbf{Y} = [\mathbf{Y}_1 \; \mathbf{Y}_2 \; \cdots \; \mathbf{Y}_N]$. We can rewrite (6.42) as follows:

$$\min_{\mathbf{X},\{\mathbf{Y}_i\}_{i=1}^N} \frac{1}{2}\sum_{i=1}^N \Big(\|\mathbf{S}_i - \mathbf{X}\mathbf{Y}_i\|_F^2 + \lambda \|\mathrm{vec}(\mathbf{Y}_i)\|_1 \Big) + \sum_{\ell=1}^n \gamma_\ell \|\mathbf{X}_{:,\ell}\|_2^2. \tag{6.43}$$

This is a special case of (6.1) with $f_i(\mathbf{X},\mathbf{Y}_i) = (1/2)\|\mathbf{S}_i - \mathbf{X}\mathbf{Y}_i\|_F^2 + (1/N)\sum_{\ell=1}^n \gamma_\ell \|\mathbf{X}_{:,\ell}\|_2^2$ and $h_i(\mathbf{Y}_i) = \lambda\|\mathrm{vec}(\mathbf{Y}_i)\|_1$. In particular, the EXTRA-AO update for it can be implemented by observing that:

$$\nabla_{\mathbf{X}} f_i(\mathbf{X}_i^t, \mathbf{Y}_i^t) = (\mathbf{X}_i^t \mathbf{Y}_i^t - \mathbf{S}_i)(\mathbf{Y}_i^t)^\top + (\gamma/N)\mathbf{X}_i^t \;, \tag{6.44}$$

$$\nabla_{\mathbf{Y}} f_i(\mathbf{X}_i^t, \mathbf{Y}_i^t) = (\mathbf{X}_i^t)^\top (\mathbf{X}_i^t \mathbf{Y}_i^t - \mathbf{S}_i) \;. \tag{6.45}$$

Moreover, Eq. (6.44) can be computed efficiently as $\mathbf{Y}_i$ is sparse. As for the proximal operation required in (6.32), since $h_i(\mathbf{Y}_i) = \|\mathrm{vec}(\mathbf{Y}_i)\|_1$, it can be replaced by the low complexity soft shrinkage operator [Beck and Teboulle(2009)].

6.4 Numerical Experiments

We divide this section into two parts — the first part focuses on the consensus based AO algorithm for least square problems, where we demonstrate its efficacy in handling a decentralized state estimation problem with asynchronous measurements; the second part focuses on the EXTRA-AO algorithm and we apply the algorithm to tackle a decentralized

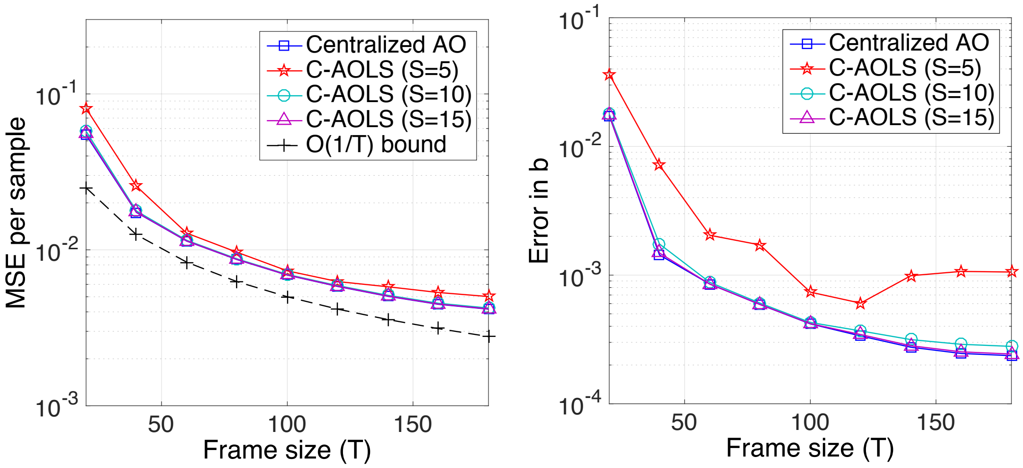

Figure 6.1: Comparing the MSE performance against the frame size T. (Left) On estimating $\{\boldsymbol{x}[k]\}_k$. (Right) On estimating $\{b_i\}_i$.

dictionary learning problem.

6.4.1 Decentralized State Estimation

In the following example, the network G is generated as an Erdos-Renyi (ER) graph with connectivity 0.5 and $N = 12$ agents. The weights on the adjacency matrix $\boldsymbol{A}$ are found with the Metropolis-Hastings rule in [Xiao and Boyd(2004)]. We consider tackling the asynchronous state estimation problem (6.26) under 'Rayleigh fading' with synthetic data. In particular, the states $\mathbf{x}[n]$ and measurement matrices $\mathbf{H}_i$ are generated as random vectors/matrices with unit variance i.i.d. complex Gaussian random entries. The DFT size is set as $F = 192$ and $\sigma_w^2 = 10^{-2}$ is the noise variance. As a benchmark, we also compare the performance of applying a centralized AO for the problem (6.26).

Our first example considers a system with sub-Nyquist and unknown asynchronous sampling, *i.e.,* we set $A_i = 2$ for all i. The system dimensions are set as $m = 4, n = 8$ and we have $N = 12$ agents. The time offsets are uniformly drawn from $\mathcal{B} = [-0.5, 0.5]$. Notice that under sub-Nyquist sampling, without exploiting the time offsets between the sensors, it is impossible to estimate the state vector $\mathbf{x}[k]$ for all k. Therefore, as a benchmark, we provide the mean square error (MSE) evaluated by comparing $\{\mathbf{x}[k]\}_{k=1}^{T}$ with an interpolated state

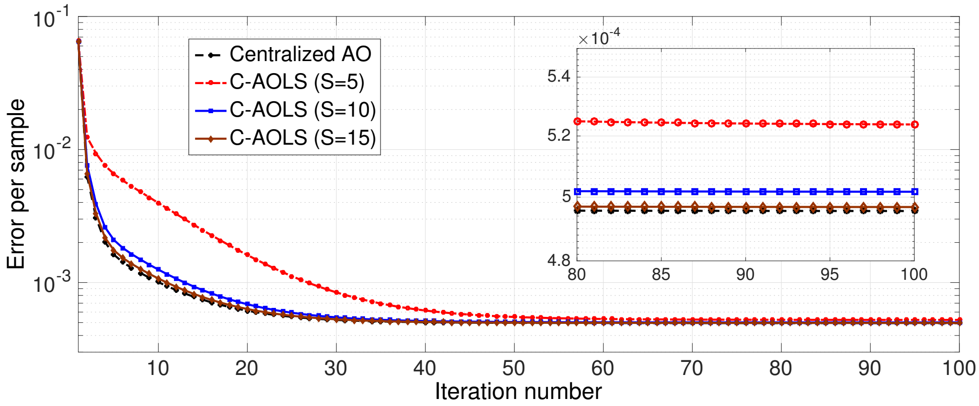

Figure 6.2: Comparing the state estimation error against iteration number of the C-AOLS algorithm.

sequence estimated from the sub-Nyquist measurements. The simulation result is shown in Figure 6.1, where we compare the MSE in state and in $\{b_i\}_{i=1}^N$ against the frame size T. From the figure, we see that the error metrics of the proposed algorithm decrease as T increases, which is due to the improved approximation to the true DTFT spectrum. In fact, the MSE in state decays as $\mathcal{O}(T^{-1})$. On the other hand, the C-AOLS algorithm achieves a similar performance to its centralized counterpart. Especially, as S increases, the performance of the former approaches that of the latter. This observation is in line with the results in Theorem 6.1.

The next example, shown in Figure 6.2, examines the convergence rate of the C-AOLS algorithm, for which we track the state estimation error as C-AOLS algorithm proceeds. In this example, we set $m = 4, n = 8, T = 120$ and consider having $N = 12$ agents. We consider solving a randomly generated instance of (6.26) and compare the MSE against iteration number. We observe that the error is gradually decreasing as the algorithm progresses and converges in about 30-40 iterations. In particular, the converged MSE improves as S increases, as predicted by Theorem 6.1.

Lastly, we describe an application of the C-AOLS algorithm to power system state estimation (PSSE) utilizing asynchronous measurements from PMU devices. Note that the power system's states are complex voltages on buses and the PMU devices provide linear

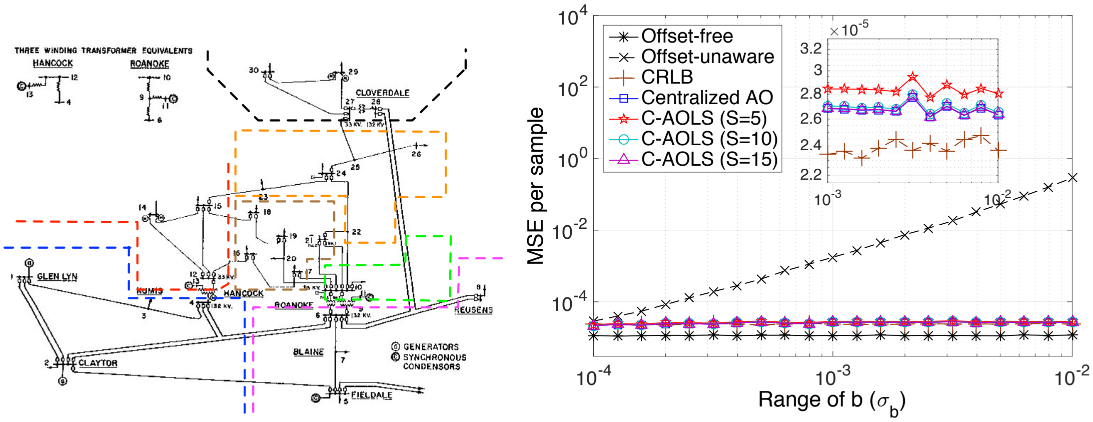

Figure 6.3: Application of C-AOLS algorithm on PSSE. (Left) Set up of the IEEE 30 bus system with $N = 7$ sensing sites (or agents), marked by the colored dashed lines. (Right) MSE per sample with asynchronous PMUs against the variance in sampling offset in b_i.

measurements of the states. To this end, we can formulate a regression problem similar to (6.26) and apply the C-AOLS algorithm. The setup and the estimation performance for performing PSSE on an IEEE-30 bus system is illustrated in Fig. 6.3; and the details are provided in [J4 of Section 1.3]. We observe that the C-AOLS algorithm achieves performance that is close to the Cramer-Rao Lower Bound (CRLB) and is comparable to the centralized AO algorithm.

6.4.2 Decentralized Dictionary Learning

This subsection presents numerical results to demonstrate the efficacy of the proposed EXTRA-AO algorithm for DL. To prepare the training data, we have randomly extracted 300 overlapping patches, each with size 16×16, from the 512×512 image of `barbara.png`, as shown in Figure 6.7. Each of the extracted patch is vectorized, thereby giving $\mathbf{S}$ a size of 256×300. We assume that there are $n = 64$ atoms, thereby giving a compression ratio of $1/4$. The size of the common dictionary $\mathbf{X}$ is 256×64. Notice that our algorithm is scalable to handle problems of larger scale.

For the decentralized DL problem (6.43), we set $\lambda = 0.03$ and $\gamma = \gamma_\ell = 0.1$ as the regularization parameters. The columns of the training data matrix $\mathbf{Y}$ is divided into

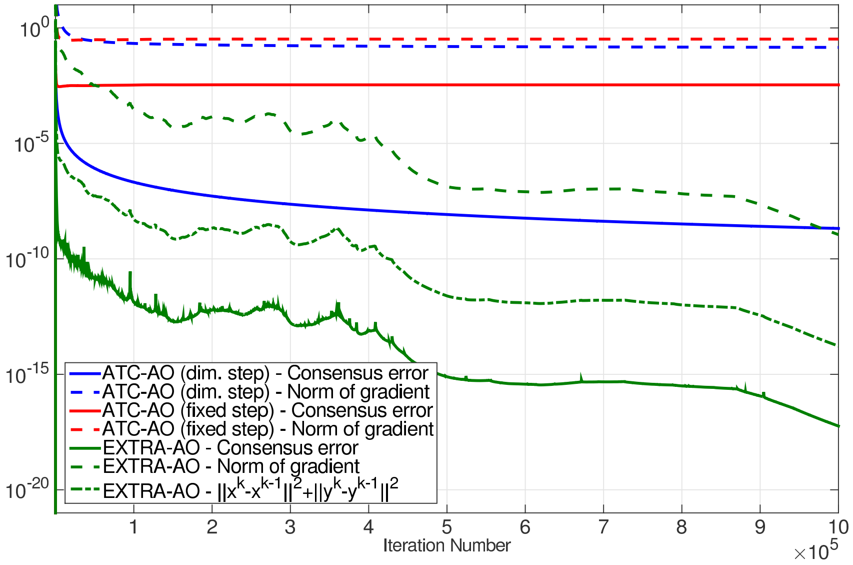

Figure 6.4: Convergence behavior of the algorithms for decentralized DL (6.43). We compare the 'consensus error', $(1/N)\sum_{i=1}^{N}\|\mathbf{x}_i^t - (1/N)\sum_{j=1}^{N}\mathbf{x}_j^t\|_2^2$, and the 'norm of gradient' w.r.t. $\mathbf{x}$, $\|\sum_{i=1}^{N}\nabla_{\mathbf{x}} f_i(\mathbf{x}_i^t, \mathbf{y}_i^t)\|_2^2$, against the iteration number t. In addition, the 'norm of difference' for EXTRA-AO denotes $\|\boldsymbol{x}^t - \boldsymbol{x}^{t-1}\|_F^2 + \|\boldsymbol{y}^t - \boldsymbol{y}^{t-1}\|_F^2$.

$N = 10$ equally sized partitions $\mathbf{S}_i \in \mathbb{R}^{256\times 30}$. It corresponds to the scenario when 10 sensors are taking samples from the image for dictionary learning. In addition to learning the dictionary $\mathbf{X}$ distributively, each agent is responsible for computing the sparse matrix $\mathbf{Y}_i$ of size 64×30 only. The network G is generated as an ER graph with $N = 10$ agents, together with connection probability of 0.6. The matrix $\boldsymbol{A}$ is constructed using the Metropolis-Hastings rule [Xiao and Boyd(2004)].

The performance of EXTRA-AO is compared to the ATC-AO method in [Chainais and Richard(2013)] and the Method of Optimal Directions (MOD) in [Engan *et al.*(1999)], where the latter is a centralized algorithm for DL. We initialized the algorithms with $\mathbf{X}$ set as the 2D discrete cosine transform (DCT) matrix. For EXTRA-AO and ATC-AO with fixed step size, we set $\alpha = 0.03$ and $\beta = 0.02$, ATC-AO with diminishing step size is set with $\alpha_t = \beta_t = 0.02 \cdot \frac{10}{(t/100)+10}$.

We first verify if the EXTRA-AO algorithm achieves convergence to a stationary point of (6.43). As shown in Figure 6.4, the norm of difference between successive iterations

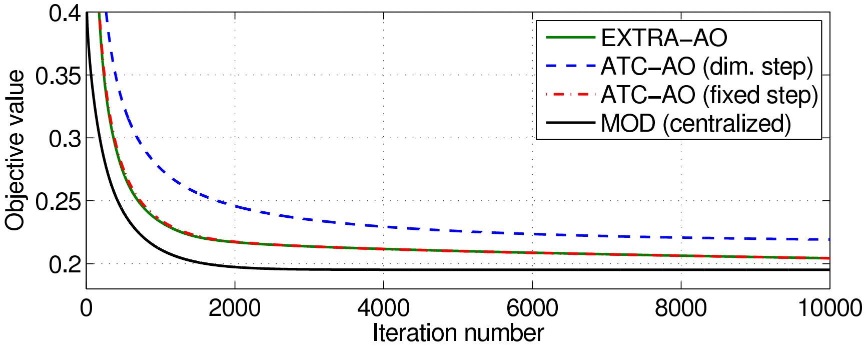

Figure 6.5: Objective value against the iteration number for the decentralized DL algorithms. The same step size selection as in Figure 6.4 is used. Notice that the solution of ATC-AO with fixed step size does not reach consensus and is infeasible to (6.43).

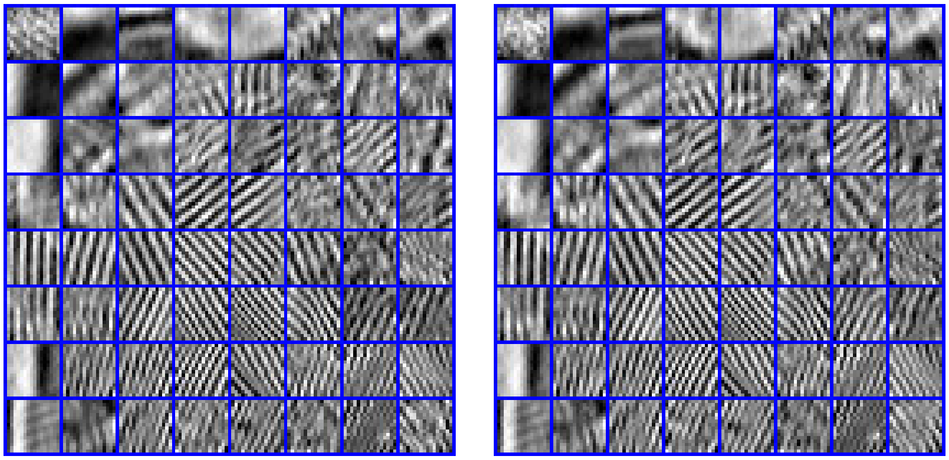

Figure 6.6: The dictionary learnt from the image `babara`: (Left) using the ATC-AO algorithm after 2×10^4 iterations. (Right) using the EXTRA-AO algorithm after 2×10^4 iterations. Each 8×8 patch represents an atom in the dictionary.

decreases to 0 as $t \to \infty$, thereby satisfying the sufficient condition in Proposition 6.2. We also note that the solution $\mathbf{X}_i$ obtained at ATC-AO (with fixed step size) does not achieve consensus and is thus infeasible to (6.1). In Figure 6.5, we compare the objective value against the number of iteration. As seen, EXTRA-AO achieves a comparable objective value to the centralized MOD algorithm. The estimated dictionary is depicted in Figure 6.6, which shows a combination of rotated tiles that correspond to the common features found in `barbara.png`. Upon careful observation, we also found that some of the atoms (e.g., the $(3, 4)$th atom) learnt by EXTRA-AO shows clearer edges than ATC-AO.

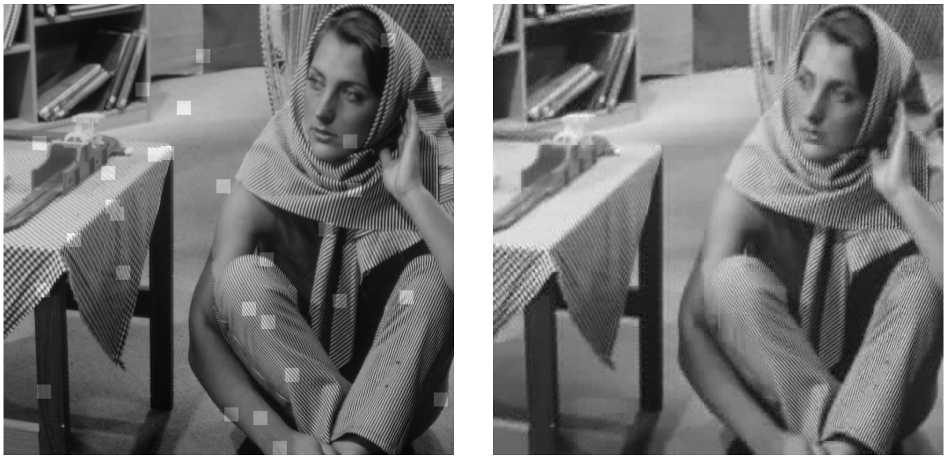

Figure 6.7: Image reconstruction result using the dictionaries learnt: (Left) the original image with the 30 shaded masks representing the training samples taken by one agent. (Right) by sparse coding using the dictionary learnt in EXTRA-AO after 2×10^4 iterations.

Lastly, Figure 6.7 shows the reconstruction result after sparse decoding using the dictionary learnt before. To promote sparsity, the sparse decoding is performed with $\lambda = 0.1$, the resulting sparse code contains only 32.12% of non-zero entries, *i.e.,* there are about 21 non-zero coefficients out of 64 for every 16×16 patch. As seen, the image reconstructed shows only a reasonable amount of artifacts compared to the original image.

6.5 Chapter Summary

This chapter focuses on using the alternating optimization (AO) technique in multi-agent optimization. In particular, we focus on tackling non-convex optimization problems with a *common variable* and a *private variable*, where it is easy to obtain solution that optimizes one of them but not both. In many practical applications, the AO technique is demonstrated to show superior performance compared to general optimization primitives such as gradient method or the Frank-Wolfe method studied in the previous chapter.

Specifically, we proposed two consensus-based AO algorithms. The first algorithm, named C-AOLS, is suitable for least square problems with nuisance parameters. Here, we proposed a consensus-based algorithm to replace the closed form solution used in the optimization of the common variable. The second algorithm, named EXTRA-AO, uses an accelerated consensus-based gradient method to optimize the common variable. For both algorithms, we provide conditions under which a stationary point of the optimization problem can be found. Moreover, we provide numerical results to verify their efficacy in applications such as signal estimation with asynchronous measurements and decentralized dictionary learning.

Appendix

6.A Proof of Proposition 6.1

To facilitate our analysis, let us define that $C_0 := \max_{\mathbf{y}} \| \sum_{i=1}^N \mathbf{H}_i(\mathbf{y}_i)^\top \mathbf{H}_i(\mathbf{y}_i)\|$ and $C_1 := \max_{\mathbf{y}} \|(\sum_{i=1}^N \mathbf{H}_i(\mathbf{y}_i)^\top \mathbf{H}_i(\mathbf{y}_i))^{-1}\|$. To begin our proof, we observe that we can rewrite the variables as:

$$\boldsymbol{\mathcal{H}}_i^{S,t} = \overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t) + \delta \boldsymbol{\mathcal{H}}_i^{S,t},\ \boldsymbol{\theta}_i^{S,t} = \bar{\boldsymbol{\theta}}(\mathbf{y}^t) + \delta \boldsymbol{\theta}_i^{S,t} \tag{6.46}$$

with the error terms satisfying

$$\|\delta \boldsymbol{\mathcal{H}}_i^{S,t}\| \leq C_0 \cdot \sigma_2(\boldsymbol{A})^S, \quad \|\delta \boldsymbol{\theta}_i^{S,t}\| \leq C_z \cdot \sigma_2(\boldsymbol{A})^S\ , \tag{6.47}$$

for some $C_z < \infty$. The bounds above are obtained as a consequence of the S rounds of average consensus applied [cf. Fact 2.1]. Under assumption (6.12), the matrix inverse admits a series expansion [Horn and Johnson(1986)]:

$$(\boldsymbol{\mathcal{H}}_i^{S,t})^{-1} = \big(\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t) + \delta \boldsymbol{\mathcal{H}}_i^{S,t}\big)^{-1} = \sum_{q=0}^{\infty} (-1)^q \Big(\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)^{-1} \delta \boldsymbol{\mathcal{H}}_i^{S,t}\Big)^q \overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)^{-1}\ . \tag{6.48}$$

Consequently, we observe the following chain of inequalities:

$$\begin{aligned}
& \|(\boldsymbol{\mathcal{H}}_i^{S,t})^{-1} \boldsymbol{\theta}_i^{S,t} - (\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t))^{-1} \bar{\boldsymbol{\theta}}^t(\mathbf{y}^t)\| \\
& = \Big\| \overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)^{-1} \delta \boldsymbol{\theta}_i^{S,t} + \sum_{q=1}^{\infty} (-1)^q \Big(\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)^{-1} \delta \boldsymbol{\mathcal{H}}_i^{S,t}\Big)^q \overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)^{-1} (\bar{\boldsymbol{\theta}}(\mathbf{y}^t) + \delta \boldsymbol{\theta}_i^{S,t}) \Big\| \\
& \leq \|\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)^{-1}\| \cdot \Big(C_z \cdot \sigma_2(\boldsymbol{A})^S + \sum_{q=1}^{\infty} \Big\| \Big(\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t)^{-1} \delta \boldsymbol{\mathcal{H}}_i^{S,t}\Big)^q \Big\| \, \|\bar{\boldsymbol{\theta}}(\mathbf{y}^t) + \delta \boldsymbol{\theta}_i^{S,t}\| \Big) \\
& \leq C_1 \cdot \Big(C_z \cdot \sigma_2(\boldsymbol{A})^S + C_Z \cdot \sum_{q=1}^{\infty} \big(C_0 C_1 \sigma_2(\boldsymbol{A}) \big)^q \Big) \\
& \leq C_1 \Big(C_z \cdot \sigma_2(\boldsymbol{A})^S + \frac{C_Z C_0 C_1}{1 - C_0 C_1 \sigma_2(\boldsymbol{A})^S} \cdot \sigma_2(\boldsymbol{A})^S \Big) \leq \frac{1}{N} \tilde{C}_0 \cdot \sigma_2(\boldsymbol{A})^S\ .
\end{aligned} \tag{6.49}$$

In the above, we have assumed that $\|\bar{\boldsymbol{\theta}}(\mathbf{y}^t) + \delta\boldsymbol{\theta}_i^{S,t}\| \leq C_Z$ in the second last inequality.

Note that $\mathbf{x}_i^{t+1} = (\boldsymbol{\mathcal{H}}_i^{S,t})^{-1}\boldsymbol{\theta}_i^{S,t}$ and $\mathbf{x}^\star(\mathbf{y}^t) = (\overline{\boldsymbol{\mathcal{H}}}(\mathbf{y}^t))^{-1}\bar{\boldsymbol{\theta}}^t(\mathbf{y}^t)$, we have

$$\textstyle\sum_{i=1}^N \|\mathbf{x}_i^{t+1} - \mathbf{x}^\star(\mathbf{y}^t)\| \leq \tilde{C}_0\sigma_2(\boldsymbol{A})^S \ , \tag{6.50}$$

i.e., the desired inequality (6.13). The remaining inequalities in the proposition, (6.14) and (6.15), can be established in a similar fashion.

6.B Proof of Theorem 6.1

Under the assumptions made in Theorem 6.1, for all t, $(\overline{\mathbf{x}}^t, \mathbf{y}^t)$ stays in the neighborhood $\mathcal{N}_{R^\star}(\boldsymbol{x}^\star, \mathbf{b}^\star)$ where the function $F(\cdot,\cdot)$ is strongly convex with modulus m_o. Our idea is to study the dynamics of the following non-negative scalar:

$$\Delta^t = F(\overline{\mathbf{x}}^t, \mathbf{y}^t) - F(\mathbf{x}^\star, \mathbf{y}^\star) \ . \tag{6.51}$$

With a slight abuse of the notations, we shall define $F(\boldsymbol{x}^t, \mathbf{y}^t) := \sum_{i=1}^N f_i(\mathbf{x}_i^t, \mathbf{y}_i^t)$ such that $\boldsymbol{x}^t$ denotes the concatenation of $(\mathbf{x}_i^t)_{i=1}^N$. We observe that:

$$\begin{aligned}
\Delta^t - \Delta^{t-1} &= F(\overline{\mathbf{x}}^t, \mathbf{y}^t) - F(\overline{\mathbf{x}}^{t-1}, \mathbf{y}^{t-1}) \\
&= F(\boldsymbol{x}^t, \mathbf{y}^t) - F(\boldsymbol{x}^t, \mathbf{y}^t) + F(\overline{\mathbf{x}}^t, \mathbf{y}^t) - F(\overline{\mathbf{x}}^{t-1}, \mathbf{y}^{t-1}) \\
&\leq L_o\tilde{C}_2\sigma_2(\mathbf{A})^S + F(\boldsymbol{x}^t, \mathbf{y}^t) - F(\overline{\mathbf{x}}^{t-1}, \mathbf{y}^{t-1}) \\
&\leq L_o\tilde{C}_2\sigma_2(\mathbf{A})^S + F(\boldsymbol{x}^t, \mathbf{y}^t) - F(\overline{\mathbf{x}}^{t-1}, \mathbf{y}^{t-1}) \\
&\qquad\quad + F(\boldsymbol{x}^t, \mathbf{y}^{t-1}) - F(\boldsymbol{x}^t, \mathbf{y}^{t-1}) \\
&\leq L_o\psi(\lambda_{\overline{W}}^{\ell_t}) + F(\boldsymbol{x}^t, \mathbf{y}^{t-1}) - F(\overline{\mathbf{x}}^{t-1}, \mathbf{y}^{t-1}) - \frac{M_o}{2}\|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2^2 \ ,
\end{aligned} \tag{6.52}$$

where the first inequality is due to Lipschitz continuity of each of $f_i(\cdot,\cdot)$ and Proposition 6.1; the second inequality is due to the descent lemma [Bertsekas(1999)] applied on the difference of function values $F(\boldsymbol{x}^t, \mathbf{y}^t) - F(\boldsymbol{x}^t, \mathbf{y}^{t-1})$ and the choice of our step size β. Moreover, we have:

$$\begin{aligned} F(\boldsymbol{x}^t, \mathbf{y}^{t-1}) &\leq F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^{t-1}) + L_o \tilde{C}_0 \sigma_2(\boldsymbol{A})^S \\ &\leq F(\overline{\boldsymbol{x}}^{t-1}, \mathbf{y}^{(t-1)}) + L_o \tilde{C}_0 \sigma_2(\boldsymbol{A})^S, \end{aligned} \tag{6.53}$$

where the first inequality is again due to the Lipschitz continuity of f_i and the second inequality is due to the optimality of $\mathbf{x}^\star(\mathbf{y}^{t-1})$ with $\mathbf{y}^{t-1}$ fixed [cf. (6.7)]. Therefore,

$$\Delta^t - \Delta^{t-1} \leq L_o \big(\tilde{C}_0 + \tilde{C}_2\big) \cdot \sigma_2(\boldsymbol{A})^S - \frac{M_o}{2} \|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2^2. \tag{6.54}$$

Our next task is to provide a lower bound for $\|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2^2$. To this end, we proceed by:

$$\begin{aligned} \Delta^t &= F(\overline{\mathbf{x}}^t, \mathbf{y}^t) - F(\mathbf{x}^\star, \mathbf{y}^\star) \\ &= F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t) - F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t) + F(\overline{\mathbf{x}}^t, \mathbf{y}^t) - F(\mathbf{x}^\star, \mathbf{y}^\star) \\ &\leq L_o \tilde{C}_1 \sigma_2(\boldsymbol{A})^S + \langle \nabla_{\mathbf{y}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t), \mathbf{y}^t - \mathbf{y}^\star \rangle + B M_o \|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2 \ , \end{aligned} \tag{6.55}$$

where in the last inequality, we have used i) f is Lipschitz continuous, ii) f is locally convex and iii)

$$\begin{aligned} &\langle \nabla_{\mathbf{x}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t), \mathbf{x}^\star(\mathbf{y}^{t-1}) - \mathbf{x}^\star \rangle \\ &= \langle \nabla_{\mathbf{x}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t) - \nabla_{\mathbf{x}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^{t-1}), \mathbf{x}^\star(\mathbf{y}^{t-1}) - \mathbf{x}^\star \rangle \\ &\leq B M_o \|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2. \end{aligned} \tag{6.56}$$

The equality is due to $\nabla_{\mathbf{x}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^{t-1}) = \mathbf{0}$.

Our next endeavor is to upper bound $\langle \nabla_{\mathbf{y}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t), \mathbf{y}^t - \mathbf{y}^\star \rangle$. To this end, we observe

$$\begin{aligned} \nabla_{\mathbf{y}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t) = \nabla_{\mathbf{y}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t) - \nabla_{\mathbf{y}} F(\boldsymbol{x}^t, \mathbf{y}^{t-1}) + \\ \frac{1}{\beta}\big((\mathbf{y}^{t-1} - \mathbf{y}^t) + \mathbf{y}^t - (\mathbf{y}^{t-1} - \beta \nabla_{\mathbf{y}} F(\boldsymbol{x}^t, \mathbf{y}^{t-1})\big) \ , \end{aligned} \tag{6.57}$$

together with the following inequality:

$$\begin{aligned}\langle \nabla_{\mathbf{y}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t) - \nabla_{\mathbf{y}} F(\boldsymbol{x}^t, \mathbf{y}^{t-1}), \mathbf{y}^t - \mathbf{y}^\star \rangle \leq \\ BM_o\big(\tilde{C}_0 \cdot \sigma_2(\boldsymbol{A}) + \|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2\big),\end{aligned} \tag{6.58}$$

which is a consequence of Cauchy-Schwarz and Proposition 6.1. Moreover, we have

$$\langle \mathbf{y}^t - (\mathbf{y}^{t-1} - \beta \nabla_{\mathbf{y}} F(\boldsymbol{x}^t, \mathbf{y}^{t-1}), \mathbf{y}^t - \mathbf{y}^\star \rangle \leq 0, \tag{6.59}$$

since $\mathbf{y}^t$ is the projection of $\mathbf{y}^{t-1} - \beta \nabla_{\mathbf{y}} F(\boldsymbol{x}^t, \mathbf{y}^{t-1})$ onto the set $\mathcal{B} := \mathcal{B}_1 \times \cdots \mathcal{B}_N$ and $\mathbf{y}^\star \in \mathcal{B}$.

Consequently,

$$\begin{aligned}\langle \nabla_{\mathbf{y}} F(\mathbf{x}^\star(\mathbf{y}^{t-1}), \mathbf{y}^t), \mathbf{y}^t - \mathbf{y}^\star \rangle \leq BM_o\big(\tilde{C}_0 \cdot \sigma_2(\boldsymbol{A}) + \|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2\big) \\ + \frac{B}{\beta} \cdot \|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2\end{aligned} \tag{6.60}$$

and we obtain a lower bound for $\|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2$ as follows:

$$\Delta^t \leq L_o \tilde{C}_1 \sigma_2(\boldsymbol{A})^S + BM_o \tilde{C}_0 \sigma_2(\boldsymbol{A})^S + 3BM_o \|\mathbf{y}^t - \mathbf{y}^{t-1}\|_2 \tag{6.61}$$

Plugging the above results back in (6.54) yields the following:

$$\begin{aligned}\Delta^t - \Delta^{t-1} \leq L_o(\tilde{C}_0 + \tilde{C}_2) \cdot \sigma_2(\boldsymbol{A})^S - \\ \frac{1}{18B^2 M_o}\Big(\max\{0, \Delta^t - (L_o \tilde{C}_1 \sigma_2(\boldsymbol{A})^S + BM_o \tilde{C}_0 \sigma_2(\boldsymbol{A})^S)\}\Big)^2 .\end{aligned} \tag{6.62}$$

Since Δ^t is non-negative, we can simplify (6.62) by considering the upper bound ξ^t such that $\Delta^t \leq \xi^t$ for all t:

$$\begin{aligned}\xi^t - \xi^{t-1} = L_o\big(\tilde{C}_0 + \tilde{C}_2\big) \cdot \sigma_2(\boldsymbol{A})^S - \\ \frac{1}{18B^2 M_o}\Big(\max\{0, \xi^t - (L_o \tilde{C}_1 \sigma_2(\boldsymbol{A})^S + BM_o \tilde{C}_0 \sigma_2(\boldsymbol{A})^S)\}\Big)^2\end{aligned} \tag{6.63}$$

A fixed point $\bar{\xi}$ to the preceding dynamic system must satisfy:

$$L_o\big(\tilde{C}_0 + \tilde{C}_2\big)\sigma_2(\boldsymbol{A})^S = \frac{1}{18B^2M_o}\Big(\max\{0, \bar{\xi} - (L_o\tilde{C}_1 + BM_o\tilde{C}_0)\cdot\sigma_2(\boldsymbol{A})^S\}\Big)^2 , \tag{6.64}$$

which implies

$$\begin{aligned}\bar{\xi} &= (L_o\tilde{C}_1 + BM_o\tilde{C}_0)\cdot\sigma_2(\boldsymbol{A})^S + \sqrt{18B^2M_oL_o\big(\tilde{C}_0+\tilde{C}_2\big)\sigma_2(\boldsymbol{A})^S} \\ &= \frac{m_o}{2}\rho\big(\sigma_2(\boldsymbol{A})\big) = \sqrt{\mathcal{O}(\sigma_2(\boldsymbol{A})^S)} .\end{aligned} \tag{6.65}$$

It can be verified that the above fixed point is stable. In fact, it is the only fixed point for the system (6.63). Finally, from (6.65) and the local strong convexity of $F(\cdot,\cdot)$, we have:

$$\lim_{t\to\infty}\|(\overline{\mathbf{x}}^{(t)}, \mathbf{y}^t) - (\mathbf{x}^\star, \mathbf{y}^\star)\|^2 \leq \frac{2}{m_o}\lim_{t\to\infty}\Delta^t = \sqrt{\mathcal{O}(\sigma_2(\boldsymbol{A})^S)} , \tag{6.66}$$

which completes the proof.

6.C Proof of Proposition 6.2

We observe that the update equation (6.31) can be rewritten as follows. For example, at $k = 2$, we have

$$\begin{aligned}\boldsymbol{x}^2 &= \boldsymbol{A}\boldsymbol{x}^1 - \alpha\nabla_{\boldsymbol{x}}\mathbf{f}(\boldsymbol{x}^1,\boldsymbol{y}^1) + \boldsymbol{x}^1 - (\tilde{\boldsymbol{A}}\boldsymbol{x}^0 - \alpha\nabla_{\boldsymbol{x}}\mathbf{f}(\boldsymbol{x}^0,\boldsymbol{y}^0)) \\ &= \boldsymbol{A}\boldsymbol{x}^1 - \alpha\nabla_{\boldsymbol{x}}\mathbf{f}(\boldsymbol{x}^1,\boldsymbol{y}^1) + (\boldsymbol{A} - \tilde{\boldsymbol{A}})\boldsymbol{x}^0,\end{aligned} \tag{6.67}$$

where the second equality is due to (6.31) with $t = 1$. By induction on $t = 3, 4, ...$, we obtain:

$$\boldsymbol{x}^{t+1} = \boldsymbol{A}\boldsymbol{x}^t - \alpha\nabla_{\boldsymbol{x}}\mathbf{f}(\boldsymbol{x}^t,\boldsymbol{y}^t) + (\boldsymbol{A} - \tilde{\boldsymbol{A}})\sum_{\ell=0}^{t-1}\boldsymbol{x}^\ell . \tag{6.68}$$

Since $\mathbf{1}^\top(\boldsymbol{A} - \tilde{\boldsymbol{A}}) = \mathbf{0}$, multiplying $(1/N)\mathbf{1}^\top$ from the left of both side in (6.68) yields

$$\overline{\mathbf{x}}^{t+1} = \overline{\mathbf{x}}^t - (1/N)\mathbf{1}^\top\nabla_{\boldsymbol{x}}\mathbf{f}(\boldsymbol{x}^t,\boldsymbol{y}^t) , \tag{6.69}$$

Note that (6.69) is similar to applying a centralized gradient method. Now, if the sequence $\{\boldsymbol{x}^t, \boldsymbol{y}^t\}_t$ converges to a unique limit point $(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty)$ as $t \to \infty$, at the limit the EXTRA update in (6.31) yields:

$$(\boldsymbol{A} - \tilde{\boldsymbol{A}})\boldsymbol{x}^\infty = \mathbf{0} \ . \tag{6.70}$$

As $\text{null}\{\boldsymbol{A} - \tilde{\boldsymbol{A}}\} = \text{null}\{\boldsymbol{A} - \mathbf{I}\} = \text{span}\{\mathbf{1}\}$, the above implies $\mathbf{x}_i^\infty = \overline{\mathbf{x}}^\infty$ for all i, *i.e.,* consensus is achieved as $k \to \infty$.

To obtain the convergence to a stationary point, applying the above to (6.69) with $t \to \infty$ gives:

$$\mathbf{0} = (1/N)\mathbf{1}^\top \nabla_{\boldsymbol{x}} \mathbf{f}(\boldsymbol{x}^\infty, \boldsymbol{y}^\infty) = (1/N)\nabla_{\mathbf{x}} F(\overline{\mathbf{x}}^\infty, \boldsymbol{y}^\infty) \ , \tag{6.71}$$

where we have used the fact that $\mathbf{x}_i^\infty = \overline{\mathbf{x}}^\infty$. This implies that $\overline{\mathbf{x}}^\infty$ is a stationary point of (6.1), given $\boldsymbol{y}^\infty$.

On the other hand, as $\boldsymbol{y}^t$ is convergent, its limit $\boldsymbol{y}^\infty$ is a fixed point to Eq. (6.32), given $\mathbf{x}_i^\infty$, *i.e.,*

$$\mathbf{y}_i^\infty = \mathbf{prox}_{\beta h_i(\cdot)}\big(\mathbf{y}_i^\infty - \beta \nabla_{\mathbf{y}} f_i(\mathbf{x}_i^\infty, \mathbf{y}_i^\infty)\big), \ \forall \ i \in [N] \ . \tag{6.72}$$

As $\beta > 0$, the above guarantees that $\mathbf{y}_i^\infty$ is a stationary point of (6.1) given $\mathbf{x}_i^\infty = \overline{\mathbf{x}}^\infty$. Combining this observation with (6.71) shows that $(\overline{\mathbf{x}}^\infty, \boldsymbol{y}^\infty)$ is a stationary point to (6.1).

6.D Proof of Proposition 6.3

We first prove the corresponding inequality pertaining to the '$\boldsymbol{x}$-update'. In particular, Eq. (6.38) follows by considering the following inequality for functions with Lipschitz continuous gradient:

$$F(\boldsymbol{x}^{t+1}, \boldsymbol{y}^t) - F(\boldsymbol{x}^t, \boldsymbol{y}^t) \leq \langle \nabla_{\boldsymbol{x}} \mathbf{f}(\boldsymbol{x}^t, \boldsymbol{y}^t), \boldsymbol{x}^{t+1} - \boldsymbol{x}^t \rangle + \frac{L_x}{2}\|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_F^2 \ , \tag{6.73}$$

where the Frobenius norm is taken as the inequality is obtained by summing up the function differences $f_i(\mathbf{x}_i^{t+1}, \mathbf{y}_i^t) - f_i(\mathbf{x}_i^t, \mathbf{y}_i^t)$ for all $i \in [N]$.

Using the result from (6.68) and the fact that $\tilde{\boldsymbol{A}} = (\boldsymbol{I} + \boldsymbol{A})/2$. We can rewrite the matrix $\nabla_{\boldsymbol{x}} \mathbf{f}(\boldsymbol{x}^t, \boldsymbol{y}^t)$ as:

$$\begin{aligned}
\nabla_{\boldsymbol{x}} \mathbf{f}(\boldsymbol{x}^t, \boldsymbol{y}^t) &= \frac{1}{\alpha}\Big(\boldsymbol{A}\boldsymbol{x}^t - \boldsymbol{x}^{t+1} + (\boldsymbol{A} - \tilde{\boldsymbol{A}})\sum_{\ell=0}^{t-1} \boldsymbol{x}^\ell\Big) \\
&= \frac{1}{\alpha}\Big(\tilde{\boldsymbol{A}}(\boldsymbol{x}^t - \boldsymbol{x}^{t+1}) - \tilde{\boldsymbol{A}}(\boldsymbol{x}^t - \boldsymbol{x}^{t+1}) + \boldsymbol{A}\boldsymbol{x}^t - \boldsymbol{x}^{t+1} + (\boldsymbol{A} - \tilde{\boldsymbol{A}})\sum_{\ell=0}^{t-1} \boldsymbol{x}^\ell\Big) \\
&= \frac{1}{\alpha}\Big(\tilde{\boldsymbol{A}}(\boldsymbol{x}^t - \boldsymbol{x}^{t+1}) + (\boldsymbol{A} - \tilde{\boldsymbol{A}})\sum_{\ell=0}^{t+1} \boldsymbol{x}^\ell\Big)\Big)
\end{aligned} \tag{6.74}$$

Plugging this back into the equation (6.73) yields:

$$\begin{aligned}
F(\boldsymbol{x}^{t+1}, \boldsymbol{y}^t) - F(\boldsymbol{x}^t, \boldsymbol{y}^t) &\le \frac{1}{\alpha}\Big\langle (\boldsymbol{A} - \tilde{\boldsymbol{A}})\sum_{\ell=0}^{t+1} \boldsymbol{x}^\ell, \boldsymbol{x}^{t+1} - \boldsymbol{x}^t \Big\rangle \\
&\qquad - \Big\langle \boldsymbol{x}^{t+1} - \boldsymbol{x}^t, \Big(\frac{1}{\alpha}\tilde{\boldsymbol{A}} - \frac{L_x}{2}\boldsymbol{I}\Big)(\boldsymbol{x}^{t+1} - \boldsymbol{x}^t) \Big\rangle \\
&\le -\delta \, \|\boldsymbol{x}^{t+1} - \boldsymbol{x}^t\|_F^2 \; + \frac{1}{\alpha}\Big\langle (\boldsymbol{A} - \tilde{\boldsymbol{A}})\sum_{\ell=0}^{t+1} \boldsymbol{x}^\ell, \boldsymbol{x}^{t+1} - \boldsymbol{x}^t \Big\rangle \; ,
\end{aligned} \tag{6.75}$$

where we have used the fact that:

$$\frac{1}{\alpha}\tilde{\boldsymbol{A}} - \frac{L_x}{2}\boldsymbol{I} \succeq \Big(\frac{\lambda_{\mathsf{min}}(\tilde{\boldsymbol{A}})}{\alpha} - \frac{L_x}{2}\Big)\boldsymbol{I} = \delta \boldsymbol{I} \; , \tag{6.76}$$

in the last inequality. On the other hand, for the '$\boldsymbol{y}$-update', the inequality follows from the classical proximal gradient analysis in Lemma 2.1.

7 Conclusions and Research Plans

As networks continue to infiltrate our everyday life, it is necessary to further our understanding of the networks, such as social networks and biological networks, around us and utilize them for our own good. Concentrated along this line of network science research, this dissertation has presented new and provable results on the longstanding problem of network identification and proposed new optimization algorithms to harvest the computation power distributed across the network.

In this dissertation, we have presented new results on the modeling and identification of network dynamics. Specifically, we have described some general models for opinion dynamics in social networks and gene dynamics in gene regulatory networks. Utilizing these dynamical system models, we proposed new formulations of network identification for the social networks and gene regulatory networks. Given the size of these networks and the limits in the amount of data one could collect, thus resulting in a set of *low rank* observed data. We noticed that the network identification problem corresponds to solving an undetermined linear system. Importantly, we analyzed a novel recovery conditions, proven using mathematical tools from graph theories and sparse recovery. This gives the theoretical guarantees that allow one to recover a large-scale networks using the proposed identification problem formulations from a few observations.

We have also developed three new decentralized algorithms for solving optimization problems over the network. In the first algorithm, we emphasize on the importance of the projection-free paradigm as it can greatly reduce the per-iteration complexity of the conventional projection-based algorithms. The new algorithm, named DeFW, was shown analytically to exhibit low iteration complexities for both convex and non-convex optimization problems. As the examples applications, we demonstrated how the new algorithm can be used to efficiently tackle a robust matrix completion problem and the classical sparse recovery problem, while requiring low communication complexity. We have also studied the alternating optimization (AO) technique for structured problems in a decentralized setting. We proposed two related algorithms. The first algorithm, called C-AOLS, is specialized to least square problems with local nuisance parameters. The second algorithm, called EXTRA-AO, is designed for tackling general non-convex optimization with a smooth objective function. We have shown the convergence of these algorithms and tested them on applications of asynchronous signal estimation and decentralized dictionary learning.

7.1 Future Research

This dissertation has laid foundations for advancing a few important problems in network science research. Future research that can be built upon these result and are listed below.

7.1.1 Network Dynamics Modeling and Identification

This dissertation has made advancements towards understanding the theoretical limits of network identification based on network dynamics. An important challenge that we have dealt with is the limitations with low-rank observation on the network, which has not been considered in most of the network identification literature. However, such a feature is observed in empirical data that may pertain to networks and have been exploited in various machine learning formulations, e.g., the low rank matrix completion problem studied in Chapter 5. In particular, general models for explaining the low rank behavior have been studied in [Udell and Townsend(2017)] at the time of finishing the writing of this dissertation. To this end, several directions on uncovering network structure from low rank observed data will be of interest:

Network identifiability condition with laxer restrictions. The theoretical findings in this dissertation, so far, are limited to the linear opinion dynamics and certain types of nonlinear gene dynamics, both taking specific forms that are relevant to their respective applications. There are two possible sub-directions — firstly and most obviously, we would like to relax the assumptions on the dynamics such as extending the network identifiability result nonlinear opinion dynamics; secondly, the notion of network identifiability may also be relaxed, *i.e.,* in some applications, identifying some high-level description of the network will be sufficient for the task, in lieu of identifying the entire network. The community detection method suggested in Section 3.5 is an example towards this sub-direction.

Applications on empirical network data. Besides social networks and gene regulatory networks, our method can also be applied to identifying networks where *perturbation data* are prevalent. An interesting example for future exploration is the inter-bank loaning network where the amount of total debts and loans of banks are dependent on the network structure when *shocks* happen in the financial market; see the systematic risk model in [Acemoglu *et al.*(2015)].

7.1.2 Large-scale Optimization Utilizing Networks

On the computation aspects utilizing networks, this dissertation has proposed several consensus-based algorithms for tackling large scale optimization problems. As our goal is to develop better algorithms for the growing demand of solving optimization problems, e.g., those arising from machine learning applications, there are a number of open problems and future research topics worth investigating.

Asynchronous DeFW algorithm. The first extension is to develop an asynchronous version of the DeFW algorithm where the agents can communicate on a time varying graph. This is an important step towards developing a fully decentralized algorithm where the agents can operate at greater degree of autonomous. In particular, our preliminary numerical result shows that we can employ an asynchronous scheme for choosing step size to achieve faster convergence.

Projection-free Primal-dual Optimization. We are investigating if the projection-free advantages of the Frank Wolfe algorithm can be incorporated into the class of primal-dual algorithms, e.g. [Chambolle and Pock(2015)], which allows one to efficiently handle constraints of complicated structures with a divide-and-conquer approach. To this end, we notice that related research along this direction has appeared at the time of submitting the dissertation [Udell and Townsend(2017)].

Convergence Analysis of Consensus-based AO Algorithms. Thus far, we have only provided a partial analysis on the convergence of the EXTRA-AO algorithm for general non-convex problems. An obvious next step is to complete the analysis and preferably obtain the convergence rate of the algorithm. We notice that a few provably convergent AO type decentralized algorithms have been proposed [Hong(2016),Lorenzo and Scutari(2016),Zhao *et al.*(2016)]. That said, it was found that EXTRA-AO still enjoys the fastest convergence empirically.

Curvature-aided Decentralized Optimization. Utilizing the incremental method's architecture for handling large-scale optimization, our recent result [cf. C6 in Section 1.3] have shown that curvature information can effectively improve the convergence rate of big data optimization. Here, the idea is to exploit the local Hessian to provide better approximation of the erroneous gradient information. To this end, an interesting direction is to study the impact of such technique in decentralized optimization.

7.2 Final Remarks

The problems addressed in this dissertation constitute only a small portion in the field of network science research. As network science is an interdisciplinary field involving engineers, mathematicians, physicists, computer scientists, biologists, sociologists, economists, etc., there are numerous gaps to be closed by combining the knowledge from different domains, as well as defining new problems that can be leveraged to solve real life issues. These are all exciting and impactful new problems to be tackled in the future.

BIBLIOGRAPHY

[Acemoglu *et al.*(2013)] Acemoglu, D., G. Como, F. Fagnani and A. Ozdaglar, "Opinion Fluctuations and Disagreement in Social Networks", Mathematics of Operations Research 38, 1, 1–27 (2013).

[Acemoglu *et al.*(2010)] Acemoglu, D., A. Ozdaglar and A. ParandehGheibi, "Spread of (mis) information in social networks", Games and Economic Behavior 70, 2, 194–227 (2010).

[Acemoglu *et al.*(2015)] Acemoglu, D., A. Ozdaglar and A. Tahbaz-Salehi, "Systemic risk and stability in financial networks", The american economic review 105, 2, 564–608 (2015).

[Albert(2005)] Albert, R., "Scale-free networks in cell biology", Journal of Cell Science 118, 21, 4947, URL http://jcs.biologists.org/content/118/21/4947.abstract (2005).

[Ansari *et al.*(2000)] Ansari, A., S. Essegaier and R. Kohli, "Internet recommendation systems", Journal of Marketing research 37, 3, 363–375 (2000).

[Arora *et al.*(2015)] Arora, S., R. Ge, T. Ma and A. Moitra, "Simple, efficient, and neural algorithms for sparse coding", in "Conference on Learning Theory", pp. 113–149 (2015).

[Asch(1955)] Asch, S. E., "Opinions and social pressure", Readings about the social animal pp. 17–26 (1955).

[Attiya and Welch(2004)] Attiya, H. and J. Welch, *Distributed Computing: Fundamentals, Simulations, and Advanced Topics* (Wiley, 2004).

[Aybat and Hamedani(2016)] Aybat, N. S. and E. Y. Hamedani, "A primal-dual method for conic constrained distributed optimization problems", in "NIPS", (2016).

[Aysal *et al.*(2009)] Aysal, T. C., M. E. Yildiz, A. D. Sarwate and A. Scaglione, "Broadcast gossip algorithms for consensus", IEEE Trans. Signal Process. 57, 7, 2748–2761 (2009).

[Bako(2011)] Bako, L., "Identification of switched linear systems via sparse optimization", Automatica 47, 4, 668 – 677 (2011).

[Banerjee *et al.*(2008)] Banerjee, O., L. E. Ghaoui and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data", Journal of Machine learning research 9, Mar, 485–516 (2008).

[Barabasi and Oltvai(2004)] Barabasi, A.-L. and Z. N. Oltvai, "Network biology: understanding the cell's functional organization", Nat Rev Genet 5, 2, 101–113, URL http://dx.doi.org/10.1038/nrg1272 (2004).

[Barzel and Barabasi(2013)] Barzel, B. and A.-L. Barabasi, "Universality in network dynamics", Nat Physics 9, 673–681 (2013).

[Barzel and Biham(2009)] Barzel, B. and O. Biham, "Quantifying the connectivity of a network: The network correlation function method", Physical Review E 80, 4, 046104–, URL http://link.aps.org/doi/10.1103/PhysRevE.80.046104 (2009).

[Bastian *et al.*(2009)] Bastian, M., S. Heymann and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks", URL http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154 (2009).

[Beck and Teboulle(2009)] Beck, A. and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems", SIAM journal on imaging sciences 2, 1, 183–202 (2009).

[Bellet *et al.*(2015)] Bellet, A., Y. Liang, A. B. Garakani, M.-F. Balcan and F. Sha, "A distributed frank-wolfe algorithm for communication-efficient sparse learning", in "Proceedings of the 2015 SIAM International Conference on Data Mining", pp. 478–486 (SIAM, 2015).

[Ben-Tal and Nemirovski(2001)] Ben-Tal, A. and A. Nemirovski, *Lectures on modern convex optimization: analysis, algorithms, and engineering applications* (SIAM, 2001).

[Berg *et al.*(2007)] Berg, E. v., M. P. Friedlander, G. Hennenfent, F. Herrmann, R. Saab and Ö. Yılmaz, "Sparco: A testing framework for sparse reconstruction", Tech. Rep. TR-2007-20, Dept. Computer Science, University of British Columbia, Vancouver (2007).

[Berinde *et al.*(2008)] Berinde, R., A. C. Gilbert, P. Indyk, H. Karloff and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery", in "46th Annual Allerton Conference on Communication, Control, and Computing", pp. 798–805 (2008).

[Bertsekas(1999)] Bertsekas, D. P., *Nonlinear programming* (Athena Scientific, 1999).

[Bianchi and Jakubowicz(2013)] Bianchi, P. and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization", IEEE Trans. Autom. Control 58, 2, 391–405 (2013).

[Blondel *et al.*(2005)] Blondel, V. D., J. M. Hendrickx, A. Olshevsky and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking", in "Proc CDC-ECC '05", vol. 2005, pp. 2996–3000 (2005).

[Blumensath and Davies(2008)] Blumensath, T. and M. E. Davies, "Iterative hard thresholding for compressed sensing", CoRR abs/0805.0510 (2008).

[Bonacich(1987)] Bonacich, P., "Power and centrality: A family of measures", American journal of sociology 92, 5, 1170–1182 (1987).

[Bonneau *et al.*(2006)] Bonneau, R., D. J. Reiss, P. Shannon, M. Facciotti, L. Hood, N. S.

Baliga and V. Thorsson, “The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo”, Genome Biology 7, 5, R36, URL http://dx.doi.org/10.1186/gb-2006-7-5-r36 (2006).

[Boutsidis *et al.*(2015)] Boutsidis, C., P. Kambadur and A. Gittens, “Spectral clustering via the power method-provably”, in “International Conference on Machine Learning”, pp. 40–48 (2015).

[Boyd *et al.*(2006)] Boyd, S., A. Ghosh, B. Prabhakar and D. Shah, “Randomized gossip algorithms”, IEEE Trans. Inf. Theory 52, 6, 2508–2530 (2006).

[Bresler(2015)] Bresler, G., “Efficiently learning ising models on arbitrary graphs”, in “Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing”, STOC ’15, pp. 771–782 (ACM, New York, NY, USA, 2015), URL http://doi.acm.org/10.1145/2746539.2746631.

[Brunet *et al.*(2004)] Brunet, J.-P., P. Tamayo, T. R. Golub and J. P. Mesirov, “Metagenes and molecular pattern discovery using matrix factorization”, Proceedings of the national academy of sciences 101, 12, 4164–4169 (2004).

[Buchanan(2010)] Buchanan, M., *Networks in Cell Biology* (Cambridge University Press, 2010), URL https://books.google.com/books?id=ojMhR2pq7qIC.

[Burke *et al.*(2005)] Burke, R., B. Mobasher and R. Bhaumik, “Limited knowledge shilling attacks in collaborative filtering systems”, in “Proceedings of 3rd International Workshop on Intelligent Techniques for Web Personalization (ITWP 2005), 19th International Joint Conference on Artificial Intelligence (IJCAI 2005)”, pp. 17–24 (2005).

[Candes and Tao(2005)] Candes, E. and T. Tao, “Decoding by Linear Programming”, IEEE Trans. Inf. Theory 51, 12, 4203–4215 (2005).

[Candès and Recht(2009)] Candès, E. J. and B. Recht, “Exact matrix completion via convex optimization”, Found. Comput. Math. 9, 6, 717–772 (2009).

[Candogan *et al.*(2012)] Candogan, O., K. Bimpikis and A. Ozdaglar, “Optimal pricing in networks with externalities”, Operations Research 60, 4, 883–905 (2012).

[Chainais and Richard(2013)] Chainais, P. and C. Richard, “Learning a common dictionary over a sensor network”, in “Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on”, pp. 133–136 (IEEE, 2013).

[Chambolle and Pock(2015)] Chambolle, A. and T. Pock, “On the ergodic convergence rates of a first-order primal–dual algorithm”, Mathematical Programming pp. 1–35 (2015).

[Chandrasekaran *et al.*(2011)] Chandrasekaran, V., S. Sanghavi, P. A. Parrilo and A. S. Willsky, “Rank-sparsity incoherence for matrix decomposition”, SIAM Journal on Optimization 21, 2, 572–596 (2011).

[Chang *et al.*(2014)] Chang, T.-H., A. Nedić and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method", IEEE Trans. Autom. Control 59, 6, 1524–1538 (2014).

[Chen(2012)] Chen, I.-A., *Fast Distributed First-Order Methods*, Master's thesis, MIT (2012).

[Ching *et al.*(2015)] Ching, E. S., P.-Y. Lai and C. Leung, "Reconstructing weighted networks from dynamics", Physical Review E 91, 3, 030801 (2015).

[Das *et al.*(2014)] Das, A., S. Gollapudi and K. Munagala, "Modeling opinion dynamics in social networks", in "Proc WSDM", pp. 403–412 (2014).

[d'Aspremont *et al.*(2008)] d'Aspremont, A., O. Banerjee and L. El Ghaoui, "First-order methods for sparse covariance selection", SIAM Journal on Matrix Analysis and Applications 30, 1, 56–66 (2008).

[Davis and Goadrich(2006)] Davis, J. and M. Goadrich, "The relationship between precision-recall and roc curves", in "Proceedings of the 23rd international conference on Machine learning", pp. 233–240 (ACM, 2006).

[De *et al.*(2014)] De, A., S. Bhattacharya, P. Bhattacharya, N. Ganguly and S. Chakrabarti, "Learning a linear influence model from transient opinion dynamics", Proc CIKM pp. 401–410 (2014).

[De Smet and Marchal(2010)] De Smet, R. and K. Marchal, "Advantages and limitations of current network inference methods", Nat Rev Micro 8, 10, 717–729, URL http://dx.doi.org/10.1038/nrmicro2419 (2010).

[Defazio *et al.*(2014)] Defazio, A., F. Bach and S. Lacoste-Julien, "SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives", in "NIPS", (2014).

[DeGroot(1974)] DeGroot, M., "Reaching a consensus", in "Journal of American Statistcal Association", vol. 69, pp. 118–121 (1974).

[Dietz *et al.*(2007)] Dietz, L., S. Bickel and T. Scheffer, "Unsupervised prediction of citation influences", in "ICML", (2007).

[DiMaggio *et al.*(1996)] DiMaggio, P., J. Evans and B. Bryson, "Have american's social attitudes become more polarized?", American journal of Sociology 102, 3, 690–755 (1996).

[Dimakis *et al.*(2010)] Dimakis, A. G., S. Kar, J. M. F. Moura, M. G. Rabbat and A. Scaglione, "Gossip Algorithms for Distributed Signal Processing", Proc. IEEE 98, 11, 1847–1864 (2010).

[Duchi *et al.*(2012)] Duchi, J., A. Agarwal and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling", IEEE Trans. Autom. Control 57, 3, 592–606 (2012).

[Duchi *et al.*(2008)] Duchi, J., S. Shalev-Shwartz, Y. Singer and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions", in "ICML", (2008).

[Eldar(2014)] Eldar, Y. C., *Sampling Theory: Beyond Bandlimited Systems* (Cambridge University Press, New York, NY, USA, 2014).

[Engan *et al.*(1999)] Engan, K., S. O. Aase and J. H. Husoy, "Method of optimal directions for frame design", in "Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on", vol. 5, pp. 2443–2446 (IEEE, 1999).

[Etesami *et al.*(2016)] Etesami, J., N. Kiyavash, K. Zhang and K. Singhal, "Learning network of multivariate hawkes processes: A time series approach", arXiv preprint arXiv:1603.04319 (2016).

[Faith *et al.*(2007)] Faith, J. J., B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins and T. S. Gardner, "Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles", PLOS Biology 5, 1, 1–13, URL http://dx.doi.org/10.1371%2Fjournal.pbio.0050008 (2007).

[Figueiredo *et al.*(2007)] Figueiredo, M. A., R. D. Nowak and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems", IEEE Journal of selected topics in signal processing 1, 4, 586–597 (2007).

[Fortunato(2010)] Fortunato, S., "Community detection in graphs", Physics reports 486, 3, 75–174 (2010).

[Foucart and Rauhut(2013)] Foucart, S. and H. Rauhut, *A mathematical introduction to compressive sensing*, vol. 1 (Birkhäuser Basel, 2013).

[Frank and Wolfe(1956)] Frank, M. and P. Wolfe, "An algorithm for quadratic programming", Naval Res. Logis. Quart. (1956).

[Friedkin and Johnsen(2011)] Friedkin, N. E. and E. C. Johnsen, *Social Influence Network Theory: A Sociological Examination of Small Group Dynamics* (Cambridge University Press, 2011).

[Friedman *et al.*(2008)] Friedman, J., T. Hastie and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso", Biostatistics 9, 3, 432–441 (2008).

[Galton(1907)] Galton, F., "Vox populi (the wisdom of crowds)", Nature 75, 7, 450–451 (1907).

[Ghosh and Lam(2015)] Ghosh, S. and H. Lam, "Computing worst-case input models in stochastic simulation", CoRR abs/1507.05609 (2015).

[Gilbert and Indyk(2010)] Gilbert, A. and P. Indyk, "Sparse recovery using sparse matrices", Proceedings of the IEEE 98, 6, 937–947 (2010).

[Golub and van Loan(1996)] Golub, G. H. and C. F. van Loan, *Matrix computations* (Johns Hopkins University Press, Baltimore, MD, 1996), third edn.

[Golub and van Loan(2013)] Golub, G. H. and C. F. van Loan, *Matrix computations* (Johns Hopkins University Press, Baltimore, MD, 2013), fourth edn.

[Grippo and Sciandrone(2000)] Grippo, L. and M. Sciandrone, "On the convergence of the block nonlinear gauss–seidel method under convex constraints", Operations research letters 26, 3, 127–136 (2000).

[Han *et al.*(2015)] Han, X., Z. Shen, W.-X. Wang and Z. Di, "Robust reconstruction of complex networks from sparse data", Phys. Rev. Lett. 114, 028701, URL http://link.aps.org/doi/10.1103/PhysRevLett.114.028701 (2015).

[Hardt and Price(2014)] Hardt, M. and E. Price, "The noisy power method: A meta algorithm with applications", in "NIPS", (2014).

[Harper and Konstan(2015)] Harper, F. M. and J. A. Konstan, "The movielens datasets: History and context", ACM TiiS (2015).

[Hartigan and Wong(1979)] Hartigan, J. A. and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm", Journal of the Royal Statistical Society. Series C (Applied Statistics) 28, 1, 100–108 (1979).

[Haury *et al.*(2012)] Haury, A.-C., F. Mordelet, P. Vera-Licona and J.-P. Vert, "Tigress: Trustful inference of gene regulation using stability selection", BMC Systems Biology 6, 1, 1–17, URL http://dx.doi.org/10.1186/1752-0509-6-145 (2012).

[He *et al.*(2015)] He, X., T. Rekatsinas, J. Foulds, L. Getoor and Y. Liu, "Hawkestopic: A joint model for network inference and topic modeling from text-based cascades", in "International Conference on Machine Learning", (2015).

[Hegselmann and Krause(2002)] Hegselmann, R. and U. Krause, "Opinion dynamics and bounded confidence models, analysis, and simulation", Journal of Artificial Societies and Social Simulation 5, 3 (2002).

[Holley and Liggett(1975)] Holley, R. A. and T. M. Liggett, "Ergodic theorems for weakly interacting infinite systems and the voter model", The annals of probability pp. 643–663 (1975).

[Hong(2016)] Hong, M., "Decomposing linearly constrained nonconvex problems by a proximal primal dual approach: Algorithms, convergence, and applications", CoRR abs/1604.00543 (2016).

[Hong *et al.*(2017)] Hong, M., X. Wang, M. Razaviyayn and Z.-Q. Luo, "Iteration complexity analysis of block coordinate descent methods", Mathematical Programming 163, 1-2, 85–114 (2017).

[Horn and Johnson(1986)] Horn, R. A. and C. R. Johnson, eds., *Matrix Analysis* (Cambridge University Press, 1986).

[Horn and Johnson(1994)] Horn, R. A. and C. R. Johnson, *Topics in matrix analysis* (Cambridge University Press, Cambridge, 1994), corrected reprint of the 1991 original.

[Hsieh *et al.*(2014)] Hsieh, C.-J., M. A. Sustik, I. S. Dhillon and P. Ravikumar, “Quic: quadratic approximation for sparse inverse covariance estimation.”, Journal of Machine Learning Research 15, 1, 2911–2947 (2014).

[Huynh-Thu *et al.*(2010)] Huynh-Thu, V. A., A. Irrthum, L. Wehenkel and P. Geurts, “Inferring regulatory networks from expression data using tree-based methods”, PLoS ONE 5, 9 (2010).

[Ideker and Sharan(2008)] Ideker, T. and R. Sharan, “Protein networks in disease”, Genome Research 18, 4, 644–652 (2008).

[Jaggi(2013)] Jaggi, M., “Revisiting Frank-Wolfe: Projection-free sparse convex optimization”, in “ICML”, (2013).

[Jaggi and Sulovsky(2010)] Jaggi, M. and M. Sulovsky, “A simple algorithm for nuclear norm regularized problems”, in “ICML”, (2010).

[Jakovetic *et al.*(2014)] Jakovetic, D., J. Xavier and J. M. F. Moura, “Fast distributed gradient methods”, IEEE Trans. Autom. Control 59, 5, 1131–1146 (2014).

[Johansson *et al.*(2008)] Johansson, B., T. Keviczky, M. Johansson and K. H. Johansson, “Subgradient methods and consensus algorithms for solving convex optimization problems”, in “Proc. CDC”, pp. 4185–4190 (2008).

[Kang *et al.*(2015)] Kang, T., R. Moore, Y. Li, E. Sontag and L. Bleris, “Discriminating direct and indirect connectivities in biological networks”, Proceedings of the National Academy of Sciences 112, 41, 12893–12898 (2015).

[Kempe *et al.*(2003)] Kempe, D., J. Kleinberg and É. Tardos, “Maximizing the spread of influence through a social network”, in “Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining”, pp. 137–146 (ACM, 2003).

[Khajehnejad *et al.*(2011)] Khajehnejad, M. A., A. G. Dimakis, W. Xu and B. Hassibi, “Sparse recovery of nonnegative signals with minimal expansion”, IEEE Trans. Signal Process. 59, 1, 196–208 (2011).

[Küffner *et al.*(2012)] Küffner, R., T. Petri, P. Tavakkolkhah, L. Windhager and R. Zimmer, “Inferring gene regulatory networks by anova”, Bioinformatics (2012).

[Kuhlman *et al.*(2012)] Kuhlman, C. J., V. S. A. Kumar and S. S. Ravi, “Controlling opinion bias in online social networks”, in “Proc. WebSci”, pp. 165–174 (2012).

[Lacoste-Julien(2016)] Lacoste-Julien, S., “Convergence rate of Frank-Wolfe for non-convex objectives”, CoRR abs/1607.00345 (2016).

[Lacoste-Julien and Jaggi(2015)] Lacoste-Julien, S. and M. Jaggi, “On the global linear convergence of Frank-Wolfe optimization variants”, in “NIPS”, (2015).

[Lam and Riedl(2004)] Lam, S. K. and J. Riedl, “Shilling recommender systems for fun and profit", in "Proceedings of the 13th international conference on World Wide Web", pp. 393–402 (ACM, 2004).

[LeCun *et al.*(2015)] LeCun, Y., Y. Bengio and G. Hinton, "Deep learning", Nature 521, 7553, 436–444 (2015).

[Li *et al.*(2011a)] Li, L., A. Scaglione and J. H. Manton, "Distributed principal subspace estimation in wireless sensor networks", IEEE Journal of Sel. Topics in Signal Process. 5, 4, 725–738 (2011a).

[Li *et al.*(2011b)] Li, L., A. Scaglione, A. Swami and Q. Zhao, "Trust, opinion diffusion and radicalization in social networks", in "Signals, systems and computers (ASILOMAR), 2011 conference record of the forty fifth asilomar conference on", pp. 691–695 (IEEE, 2011b).

[Li and Scaglione(2013)] Li, X. and A. Scaglione, "Convergence and Applications of a Gossip-Based Gauss-Newton Algorithm", IEEE Trans. Signal Process. 61, 21, 5231–5246 (2013).

[Ling *et al.*(2012)] Ling, Q., Y. Xu, W. Yin and Z. Wen, "Decentralized low-rank matrix completion", in "Proc ICASSP", (2012).

[Liu and Vandenberghe(2010)] Liu, Z. and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification", SIAM J. Matrix Anal. Appl. 31, 3, 1235–1256 (2010).

[Lorenzo and Scutari(2016)] Lorenzo, P. D. and G. Scutari, "Next: In-network nonconvex optimization", IEEE Trans. on Signal and Info. Process. over Networks (2016).

[Mackey *et al.*(2015)] Mackey, L., A. Talwalkar and M. I. Jordan, "Distributed matrix completion and robust factorization", Journal of Machine Learning Research 16, 913–960 (2015).

[Marbach *et al.*(2012)] Marbach, D., J. C. Costello, R. Kuffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, M. Kellis, J. J. Collins and G. Stolovitzky, "Wisdom of crowds for robust gene network inference", Nat Meth 9, 8, 796–804, URL http://dx.doi.org/10.1038/nmeth.2016 (2012).

[Massart(2003)] Massart, P., *Concentration Inequalities and Model Selection* (Springer, 2003).

[Menten and Michaelis(1913)] Menten, L. and M. Michaelis, "Die kinetik der invertinwirkung", Biochem Z 49, 333–369 (1913).

[Mirollo and Strogatz(1990)] Mirollo, R. E. and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators", SIAM Journal on Applied Mathematics 50, 6, 1645–1662 (1990).

[Mobilia(2003)] Mobilia, M., "Does a single zealot affect an infinite group of voters?", Physical review letters 91, 2, 028701 (2003).

[Mobilia *et al.*(2007)] Mobilia, M., A. Petersen and S. Redner, “On the role of zealotry in the voter model”, Journal of Statistical Mechanics: Theory and Experiment 2007, 08, P08029 (2007).

[Mohimani *et al.*(2007)] Mohimani, G. H., M. Babaie-Zadeh and C. Jutten, “Fast Sparse Representation Based on Smoothed L0 Norm”, in “ICA”, Lecture Notes in Computer Science, pp. 389–396 (Springer, 2007).

[Moussaïd *et al.*(2013)] Moussaïd, M., J. E. Kämmer, P. P. Analytis and H. Neth, “Social influence and the collective dynamics of opinion formation”, PloS one 8, 11, e78433 (2013).

[Moussad *et al.*(2013)] Moussad, M., J. E. Kammer, P. P. Analytis and H. Neth, “Social influence and the collective dynamics of opinion formation”, PLoS ONE 8, 11 (2013).

[Nedić *et al.*(2016)] Nedić, A., A. Olshevsky and W. Shi, “Achieving geometric convergence for distributed optimization over time-varying graphs”, CoRR abs/1607.03218 (2016).

[Nedić *et al.*(2010)] Nedić, A., A. Ozdaglar and P. Parrilo, “Constrained Consensus and Optimization in Multi-Agent Networks”, IEEE Trans. Autom. Control 55, 4, 922–938 (2010).

[Nesterov(2012)] Nesterov, Y., “Efficiency of coordinate descent methods on huge-scale optimization problems”, SIAM Journal on Optimization 22, 2, 341–362 (2012).

[Ng *et al.*(2002)] Ng, A. Y., M. I. Jordan and Y. Weiss, “On spectral clustering: Analysis and an algorithm”, in “Advances in neural information processing systems”, pp. 849–856 (2002).

[Nurminskii(1972)] Nurminskii, E. A., “Convergence conditions for nonlinear programming algorithms”, Cybernetics , 6, 79–81 (1972).

[Patterson *et al.*(2014)] Patterson, S., Y. C. Eldar and I. Keidar, “Distributed compressed sensing for static and time-varying networks”, IEEE Trans. on Signal Process. 62, 19, 4931–4946 (2014).

[Petralia *et al.*(2015)] Petralia, F., P. Wang, J. Yang and Z. Tu, “Integrative random forest for gene regulatory network inference”, Bioinformatics 31, 12, i197–i205 (2015).

[Polyak(1987)] Polyak, B. P., *Introduction to Optimization* (Optimization Software, Inc., 1987).

[Pouget-Abadie and Horel(2015)] Pouget-Abadie, J. and T. Horel, “Inferring graphs from cascades: A sparse recovery framework”, in “Proceedings of the 32nd International Conference on Machine Learning, (ICML 2015)”, (2015), URL http://econcs.seas.harvard.edu/files/econcs/files/pouget_icml15.pdf.

[Qu and Li(2016)] Qu, G. and N. Li, “Harnessing smoothness to accelerate distributed optimization”, CoRR abs/1605.07112 (2016).

[Ram *et al.*(2012)] Ram, S. S., A. Nedić and V. V. Veeravalli, "A new class of distributed optimization algorithms : application to regression of distributed data", Optimization Methods and Software , 1, 37–41 (2012).

[Ramos *et al.*(2015)] Ramos, M., J. Shao, S. D. S. Reis, C. Anteneodo, J. S. A. Jr, S. Havlin and H. A. Makse, "How does public opinion become extreme?", Sci. Rep. , 10032 (2015).

[Ravazzi *et al.*(2016)] Ravazzi, C., S. M. Fosson and E. Magli, "Randomized algorithms for distributed nonlinear optimization under sparsity constraints", IEEE Trans. on Signal Process. 64, 6, 1420–1434 (2016).

[Ravazzi *et al.*(2015)] Ravazzi, C., P. Frasca, R. Tempo and H. Ishii, "Ergodic randomized algorithms and dynamics over networks", IEEE transactions on control of network systems 2, 1, 78–87 (2015).

[Razaviyayn *et al.*(2013)] Razaviyayn, M., M. Hong and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization", SIAM Journal on Optimization 23, 2, 1126–1153 (2013).

[Recht and Ré(2013)] Recht, B. and C. Ré, "Parallel stochastic gradient algorithms for large-scale matrix completion", Mathematical Programming Computation 5, 2, 201–226 (2013).

[Reddi *et al.*(2016)] Reddi, S. J., S. Sra, B. Póczos and A. Smola, "Stochastic frank-wolfe methods for nonconvex optimization", in "Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on", pp. 1244–1251 (IEEE, 2016).

[Ronen *et al.*(2002)] Ronen, M., R. Rosenberg, B. I. Shraiman and U. Alon, "Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics", Proceedings of the National Academy of Sciences 99, 16, 10555–10560, URL http://www.pnas.org/content/99/16/10555.abstract (2002).

[Rosenbaum and Tsybakov(2010)] Rosenbaum, M. and A. Tsybakov, "Sparse recovery under matrix uncertainty", Annals of Statistics 38, 5, 2620–2651 (2010).

[Rudelson and Vershynin(2009)] Rudelson, M. and R. Vershynin, "Smallest singular value of a random rectangular matrix", Communications on Pure and Applied Mathematics 62, 12, 1707–1739 (2009).

[Sayed *et al.*(2013)] Sayed, A. H., S.-Y. Tu, J. Chen, X. Zhao and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: an examination of distributed strategies and network behavior", IEEE Signal Process. Mag. 30, 3, 155–171 (2013).

[Scaglione *et al.*(2008)] Scaglione, A., R. Pagliari and H. Krim, "The decentralized estimation of the sample covariance", in "Proc. Asilomar", pp. 1722–1726 (2008).

[Scheinberg *et al.*(2010)] Scheinberg, K., S. Ma and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods", in "Advances in neural information processing systems", pp. 2101–2109 (2010).

[Segarra *et al.*(2016)] Segarra, S., A. G. Marques, G. Mateos and A. Ribeiro, "Network topology inference from spectral templates", arXiv preprint arXiv:1608.03008 (2016).

[Shen *et al.*(2017)] Shen, Y., B. Baingana and G. B. Giannakis, "Kernel-based structural equation models for topology identification of directed networks", IEEE Transactions on Signal Processing 65, 10, 2503–2516 (2017).

[Shen *et al.*(2014)] Shen, Z., W.-X. Wang, Y. Fan, Z. Di and Y.-C. Lai, "Reconstructing propagation networks with natural diversity and identifying hidden sources", Nature communications 5 (2014).

[Shi *et al.*(2015)] Shi, W., Q. Ling, G. Wu and W. Yin, "A proximal gradient algorithm for decentralized composite optimization", IEEE Trans. on Signal Process. 63, 22, 6013–6023 (2015).

[Simonetto and Jamali-Rad(2016)] Simonetto, A. and H. Jamali-Rad, "Primal recovery from consensus-based dual decomposition for distributed convex optimization", JOTA 168, 1, 172–197 (2016).

[Singh and Vidyasagar(2016)] Singh, N. and M. Vidyasagar, "blars: An algorithm to infer gene regulatory networks", IEEE/ACM Transactions on Computational Biology and Bioinformatics 13, 2, 301–314 (2016).

[Sontag(2008)] Sontag, E. D., "Network reconstruction based on steady-state data", Essays in Biochemistry 45, 161–176 (2008).

[Tang *et al.*(2012a)] Tang, J., H. Gao, H. Liu and A. Das Sarma, "etrust: Understanding trust evolution in an online world", in "Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining", KDD '12, pp. 253–261 (ACM, New York, NY, USA, 2012a), URL http://doi.acm.org/10.1145/2339530.2339574.

[Tang *et al.*(2012b)] Tang, J., T. Lou and J. Kleinberg, "Inferring social ties across heterogeneous networks", in "In WSDM'12", pp. 743–752 (2012b).

[Tatarenko and Touri(2017)] Tatarenko, T. and B. Touri, "Non-convex distributed optimization", IEEE Transactions on Automatic Control (2017).

[Timme(2007)] Timme, M., "Revealing network connectivity from response dynamics", Physical Review Letters 98, 22, 1–4 (2007).

[Traud *et al.*(2012)] Traud, A. L., P. J. Mucha and M. A. Porter, "Social structure of Facebook networks", Physica A: Statistical Mechanics and its Applications 391, 16, 4165–4180 (2012).

[Tremblay *et al.*(2016)] Tremblay, N., G. Puy, R. Gribonval and P. Vandergheynst, "Compressive spectral clustering", in "International Conference on Machine Learning", pp. 1002–1011 (2016).

[Tsianos *et al.*(2012)] Tsianos, K. I., S. Lawlor and M. G. Rabbat, "Push-sum distributed dual averaging for convex optimization", in "Decision and Control (CDC), 2012 IEEE 51st Annual Conference on", pp. 5453–5458 (IEEE, 2012).

[Tsitsiklis(1984)] Tsitsiklis, J., *Problems in decentralized decision making and computation*, Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., Boston, MA (1984).

[Udell and Townsend(2017)] Udell, M. and A. Townsend, "Nice latent variable models have log-rank", arXiv preprint arXiv:1705.07474 (2017).

[Vaidyanathan(1993)] Vaidyanathan, P. P., *Multirate systems and filter banks* (Pearson Education India, 1993).

[Wainwright *et al.*(2008)] Wainwright, M. J., M. I. Jordan *et al.*, "Graphical models, exponential families, and variational inference", Foundations and Trends® in Machine Learning 1, 1–2, 1–305 (2008).

[Wang *et al.*(2011a)] Wang, M., W. Xu and A. Tang, "A unique "nonnegative" solution to an underdetermined system: From vectors to matrices", IEEE Trans. Signal Process. 59, 3, 1007–1016 (2011a).

[Wang *et al.*(2011b)] Wang, W.-X., Y.-C. Lai, C. Grebogi and J. Ye, "Network Reconstruction Based on Evolutionary-Game Data via Compressive Sensing", Physical Review X 1, 2, 1–7 (2011b).

[Wei and Ozdaglar(2013)] Wei, E. and A. Ozdaglar, "On the o(1/k) convergence of asynchronous distributed alternating direction method of multipliers", CoRR abs/1307.8254 (2013).

[West(2000)] West, D. B., *Introduction to Graph Theory* (Prentice Hall, 2000), 2 edn.

[Williams *et al.*(2007)] Williams, C. A., B. Mobasher and R. Burke, "Defending recommender systems: detection of profile injection attacks", Service Oriented Computing and Applications 1, 3, 157–170 (2007).

[Wu *et al.*(2016)] Wu, J., X. Zhao, Z. Lin and Z. Shao, "Large scale gene regulatory network inference with a multi-level strategy", Molecular BioSystems 12, 2, 588–597, URL http://dx.doi.org/10.1039/C5MB00560D (2016).

[Xiang *et al.*(2010)] Xiang, R., J. Neville and M. Rogati, "Modeling relationship strength in online social networks", in "NIPS", (2010).

[Xiao and Boyd(2004)] Xiao, L. and S. Boyd, "Fast linear iterations for distributed averaging", Systems & Control Letters 53, 1, 65–78 (2004).

[Yang *et al.*(2014)] Yang, Y., G. Scutari, D. P. Palomar and M. Pesavento, "A parallel stochastic approximation method for nonconvex multi-agent optimization problems", CoRR abs/1410.5076 (2014).

[Yildiz *et al.*(2013)] Yildiz, M. E., A. Ozdaglar, D. Acemoglu, A. Saberi and A. Scaglione, "Binary opinion dynamics with stubborn agents", ACM Trans. Econ. Comput. 1, 4, 19 (2013).

[Yildiz and Scaglione(2008)] Yildiz, M. E. and A. Scaglione, "Coding with side information for rate-constrained consensus", IEEE Trans. on Signal Process. 56, 8, 3753–3764 (2008).

[Yildiz and Scaglione(2010)] Yildiz, M. E. and A. Scaglione, "Computing along routes via gossiping", IEEE Trans. on Signal Process. 58, 6, 3313–3327 (2010).

[Yip *et al.*(2010)] Yip, K. Y., R. P. Alexander, K.-K. Yan and M. Gerstein, "Improved reconstruction of in silico gene regulatory networks by integrating knockout and perturbation data", PLoS ONE 5, 1, e8121–, URL http://dx.doi.org/10.1371%2Fjournal.pone.0008121 (2010).

[Yu *et al.*(2012)] Yu, H.-F., C.-J. Hsieh, S. Si and I. Dhillon, "Scalable coordinate descent approaches to parallel matrix factorization for recommender systems", in "ICDM", pp. 765–774 (IEEE, 2012).

[Yu *et al.*(2014)] Yu, Y., X. Zhang and D. Schuurmans, "Generalized conditional gradient for sparse estimation", CoRR (2014).

[Yuan *et al.*(2012)] Yuan, G.-X., C.-H. Ho and C.-J. Lin, "An improved glmnet for l1-regularized logistic regression", Journal of Machine Learning Research 13, Jun, 1999–2030 (2012).

[Yun *et al.*(2011)] Yun, S., P. Tseng and K.-C. Toh, "A block coordinate gradient descent method for regularized convex separable optimization and covariance selection", Mathematical programming 129, 2, 331–355 (2011).

[Zhao *et al.*(2016)] Zhao, M.-M., Q. Shi and M. Hong, "A distributed algorithm for dictionary learning over networks", in "Signal and Information Processing (GlobalSIP), 2016 IEEE Global Conference on", pp. 505–509 (IEEE, 2016).