Statistical Signal Processing on Dynamic Graphs with Applications in Social Networks

by

Maziyar Hamdi

M.A.Sc., The University of British Columbia, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

December 2015

© Maziyar Hamdi 2015

Abstract

Due to the proliferation of social networks and their significant effects on our day-to-day activities, there has been a growing interest in modeling and analyzing behavior of agents in social networks over the past decade. The unifying theme of this thesis is to develop a set of mathematical theories and algorithmic tools for different estimation and sensing problems over graphs with applications to social networks.

The first part of this dissertation is devoted to multi-agent Bayesian estimation and learning problem in social networks. We consider a set of agents that interact over a network to estimate an unknown parameter called state of nature. As a result of the recursive nature of Bayesian models and the correlation introduced by the structure of the underlying communication graph, information collected by one agent can be mistakenly considered independent, that is, *mis-information propagation*, also known as *data incest* arises. This part presents data incest removal algorithms that ensure complete mitigation of the mis-information associated with the estimates of agents in two different information exchange patterns: First, a scenario where beliefs (posterior distribution of state of nature) are transmitted over the network. Second, a social learning context where agents map their local beliefs into a finite set of actions and broadcast their actions to other agents. We also present a necessary and sufficient condition on the structure of information flow graph to mitigate mis-information propagation.

The second part of the thesis considers a Markov-modulated duplication-deletion random graph where at each time instant, one node can either join or leave the network; the probabilities of joining or leaving evolve according to the realization of a finite state Markov chain. This part presents two results. First, motivated by social network applications, the asymptotic behavior of the degree distribution is analyzed. Second, a stochastic approximation algorithm is presented to track empirical degree distribution as it evolves over time. The tracking performance of the algorithm is analyzed in terms of mean square error and a functional central limit theorem is presented for the asymptotic tracking error.

Preface

The work presented in this thesis is based on the research and development conducted in the Statistical Signal Processing Laboratory at the University of British Columbia (Vancouver). The research work presented in the chapters of this dissertation is performed by the author with feedback and assistance provided by Prof. Vikram Krishnamurthy. The author is responsible for writeup, problem formulation, research development, data analyses and numerical studies presented in this dissertation with frequent suggestions, technical and editorial feedback from Prof. Vikram Krishnamurthy. Hilbert space analysis of Chapter 4 are due in part to Prof. George Yin. The dataset for the psychology experiment presented in Chapter 3 is obtained from Prof. Alan Kingstone and Dr. Grayden Solman. For the psychology experiment, informed consent was obtained from all participants, and all experimental procedures and protocols were reviewed and approved by the University of British Columbia Behavioral Research Ethics Board (UBC BREB). The UBC BREB approval number is: H10-00527. The work presented in different chapters of this thesis has been appeared in several publications which are listed below. In these publications, all co-authors contributed to the editing of the manuscript.

- The work of Chapter 2 has been presented in the following publications:
 - [Journal Paper] V. Krishnamurthy, M. Hamdi, *Mis-information Removal Algorithms* on Social Networks: Constrained Estimation on Random Graphs, IEEE Journal of Selected Topics in Signal Processing, vol:7, issue: 2, pp.333–346.
 - [Conference Paper] V. Krishnamurthy and M. Hamdi, *Data Fusion and Mis-information Removal in Social Networks*, IEEE conference on Information Fusion, 2012, Singapore, July 2012.
- Materials in Chapter 3 have been appeared in the following publications and preprints for possible publications:
 - [Book] V. Krishnamurthy, O. N. Gharehshiran, and M. Hamdi. *Interactive Sensing and Decision Making in Social Networks*. Now Publishers Inc., Hanover, MA, 2014.

- [Journal Paper] M. Hamdi, V. Krishnamurthy Removal of data incest in multi-agent social learning in social networks, preprint: arXiv:1309.6687.
- [Report] M Hamdi, G Solman, A Kingstone, V Krishnamurthy Social learning in a human society: An experimental study, preprint: arXiv:1408.5378.
- The work of Chapter 4 has been presented in the following publications:
 - [Book] V. Krishnamurthy, O. N. Gharehshiran, and M. Hamdi. *Interactive Sensing and Decision Making in Social Networks*. Now Publishers Inc., Hanover, MA, 2014.
 - [Journal Paper] M. Hamdi, V. Krishnamurthy, G. Yin *Tracking a Markov-Modulated* Stationary Degree Distribution of a Dynamic Random Graph, IEEE Transaction on Information Theory, vol 60, issue 10, pp. 6609–6625.
 - [Conference Paper] M. Hamdi and V. Krishnamurthy, A Novel Use of Stochastic Approximation Algorithms for Estimating Degree of Each Node in Social Networks, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'12), Kyoto, Japan, March 2012.
 - [Conference Paper] M.Hamdi, V. Krishnamurthy, *The Asymptotic of Duplication-deletion Random Graphs, accepted for publication in 44th Asilomar Conference on Signals, Systems and Computers. Nov. 2010.*
- Although not being presented in this thesis, the discussion and results about social networks that has appeared in the following journal article was inspired by the work presented in this thesis:
 - [Journal Paper] A. Leshem, M. Hamdi, V. Krishnamurthy, Boundary Value Problems in Consensus Networks, Submitted to IEEE Transactions on Signal Processing, arXiv preprint arXiv:1407.7170.

| Al | ostrac | t | i | | | | | | | | |
|----|-------------------|----------|---|--|--|--|--|--|--|--|--|
| Pr | reface | | ii | | | | | | | | |
| Ta | Table of Contents | | | | | | | | | | |
| Li | st of 7 | fables . | ix | | | | | | | | |
| Li | st of H | figures | х | | | | | | | | |
| Ac | cknow | ledgme | nts | | | | | | | | |
| De | edicati | ion | xv | | | | | | | | |
| 1 | Intro | oduction | n | | | | | | | | |
| | 1.1 | Overvi | ew | | | | | | | | |
| | | 1.1.1 | Bayesian Estimation over Social Networks | | | | | | | | |
| | | 1.1.2 | Interactive Social Learning over Networks | | | | | | | | |
| | | 1.1.3 | Tracking Degree Distribution of Social Networks | | | | | | | | |
| | 1.2 | Main C | Contributions | | | | | | | | |
| | | 1.2.1 | Bayesian Estimation over Social Networks | | | | | | | | |
| | | 1.2.2 | Interactive Social Learning over Networks | | | | | | | | |
| | | 1.2.3 | Tracking Degree Distribution of Social Networks | | | | | | | | |
| | 1.3 | Related | 1 Works | | | | | | | | |
| | | 1.3.1 | Bayesian Estimation over Social Networks | | | | | | | | |
| | | 1.3.2 | Interactive Social Learning over Networks 12 | | | | | | | | |
| | | 1.3.3 | Tracking Degree Distribution of Social Networks | | | | | | | | |
| | 1.4 | Thesis | Outline | | | | | | | | |
| Pa | art I | Estim | ation and Learning Over Directed Acyclic Graphs | | | | | | | | |
| 2 | Con | strained | l Estimation Over Random Graphs | | | | | | | | |

| | 2.1 | Introdu | uction | 19 |
|---|-----|---------|---|----|
| | | 2.1.1 | Chapter Goals | 21 |
| | | 2.1.2 | Main Results and Organization of Chapter | 22 |
| | 2.2 | Model | ing Information Flow in Social Networks | 22 |
| | | 2.2.1 | Constrained Information Flow Protocol | 23 |
| | | 2.2.2 | Benchmark Full Information Flow Protocol | 24 |
| | | 2.2.3 | Modeling Time Evolution of the Information Flow | 25 |
| | 2.3 | Optim | al Mis-information Propagation Removal Algorithm | 25 |
| | | 2.3.1 | Optimal Combination Scheme in Constrained Information Flow Protocol . | 26 |
| | 2.4 | Sub-op | ptimal Mis-information Removal Algorithm Without Complete Knowledge of | |
| | | Inform | nation Flow Graph | 29 |
| | | 2.4.1 | Sub-optimal Combination Scheme | 30 |
| | 2.5 | Numer | rical Examples | 30 |
| | | 2.5.1 | Example and Intuition on Theorems 2.2.1 and 2.3.2 | 31 |
| | | 2.5.2 | Numerical Examples Illustrating Alg. \mathscr{A} in Estimation Problem (2.5) | 32 |
| | | 2.5.3 | Numerical Examples Illustrating Alg. \mathscr{B} in Estimation Problem (2.14) | 34 |
| | 2.6 | Closin | g Remarks | 38 |
| | 2.7 | Proof | of Theorems | 39 |
| | | 2.7.1 | Proof of Theorem 2.2.1 | 40 |
| | | 2.7.2 | Proof of Theorem 2.3.1 | 40 |
| | | 2.7.3 | Proof of Theorem 2.3.2 | 41 |
| 3 | Mis | -inform | ation Management Problem in Social Learning Over Directed Graphs . | 42 |
| | 3.1 | Introdu | uction | 42 |
| | | 3.1.1 | Social Learning Protocol on Network | 43 |
| | | 3.1.2 | Chapter Goals | 45 |
| | | 3.1.3 | Main Results and Organization of Chapter | 47 |
| | 3.2 | Social | Learning Over Social Networks | 48 |
| | | 3.2.1 | Constrained Social Learning in Social Networks | 49 |
| | 3.3 | Data I | ncest Removal Algorithm | 51 |
| | | 3.3.1 | The Idealized Benchmark for Data Incest Free Social Learning in Social | |
| | | | Networks | 51 |
| | | 3.3.2 | The Data Incest Free Belief in the Idealized Social Learning Protocol 2 | 52 |
| | | 3.3.3 | Data Incest Removal Algorithm for Problem (3.9) With Constrained Social | |
| | | | Learning Protocol 1 | 53 |
| | | 3.3.4 | Discussion of Data Incest Removal in Social Learning | 57 |
| | 3.4 | Numer | rical Examples | 58 |

| | 3.5 | Psycho | logy Experiment | 65 |
|----|-------|----------|---|-----|
| | | 3.5.1 | Experiment Setup | 65 |
| | | 3.5.2 | Experimental Results | 68 |
| | 3.6 | Closing | g Remarks | 71 |
| | 3.7 | Proof c | of Results | 73 |
| | | 3.7.1 | Proof of Lemma 3.2.1 | 73 |
| | | 3.7.2 | Proof of Theorem 3.3.1 | 75 |
| | | 3.7.3 | Proof of Theorem 3.3.2 | 76 |
| Pa | rt II | Tracl | king Degree Distribution in Dynamic Social Networks | 78 |
| 4 | Trac | king a I | Markov Modulated Degree Distribution | 79 |
| | 4.1 | Introdu | ction | 79 |
| | | 4.1.1 | Chapter Goals | 79 |
| | | 4.1.2 | Main Results and Organization of Chapter | 81 |
| | 4.2 | Markov | v-modulated Dynamic Random Graph of Duplication-deletion Type | 82 |
| | 4.3 | Asymp | totic Degree Distribution Analysis for Non-Markov Modulated case | 84 |
| | | 4.3.1 | Fixed Size Random Graph | 84 |
| | | 4.3.2 | Power Law Exponent for Infinite Duplication-deletion Random Graph | 87 |
| | 4.4 | Trackir | ng the Degree Distribution of the Fixed Size Markov-modulated Random | |
| | | Graph | | 88 |
| | | 4.4.1 | Tracking Error of the Stochastic Approximation Algorithm | 91 |
| | | 4.4.2 | Limit System of Regime-Switching Ordinary Differential Equations | 92 |
| | | 4.4.3 | Scaled Tracking Error | 92 |
| | 4.5 | Estima | ting the Degree Distribution of Infinite Duplication-deletion Random Graphs | 93 |
| | | 4.5.1 | Infinite Random Graphs without Markovian Dynamics | 93 |
| | | 4.5.2 | Markov-modulated Probability Mass Functions with Denumerable Support | 96 |
| | 4.6 | Numer | ical Examples | 98 |
| | 4.7 | Closing | g Remarks | 102 |
| | 4.8 | Proof o | of Results | 104 |
| | | 4.8.1 | Proof of Theorem 4.3.1 | 104 |
| | | 4.8.2 | Proof of Theorem 4.3.2 | 109 |
| | | 4.8.3 | Proof of Lemma 4.8.1 | 112 |
| | | 4.8.4 | Proof of Lemma 4.8.2 | 112 |
| | | 4.8.5 | Proof of Theorem 4.4.1 | 116 |
| | | 4.8.6 | Sketch of the Proof of Theorem 4.4.2 | 118 |
| | | 4.8.7 | Sketch of the Proof of Theorem 4.4.3 | 119 |

| | | 4.8.8 | Proof of Theorem 4.5.2 | | | | | | | ••• | • | ••• | 120 |
|-----|--------|-----------|---------------------------|----------|-------|------|---|---------|------|---------|---------|-----|-----|
| | | 4.8.9 | Proof of Theorem 4.5.3 | | | | | | | | • | | 122 |
| | | 4.8.10 | Proof of Theorem 4.5.4 | | | | | | | ••• | • • | ••• | 125 |
| 5 | Con | clusions | | | | | | | | | | | 127 |
| | 5.1 | Summa | ry of Findings in Part I | | | | | | | • | | | 127 |
| | 5.2 | Summa | ry of Findings in Part II | | | | | | | | | | 129 |
| | 5.3 | Direction | ons for Future Research a | nd Deve | elopr | nent | • | • • | | ••• | • | ••• | 130 |
| Bil | bliogr | aphy . | | | | | | | | | • | ••• | 133 |
| A | Som | e Graph | 1 Theoretic Definitions | | | | | | | | • | ••• | 145 |
| Ap | pend | ices | | | | | | | | | | ••• | 145 |
| B | A No | ote on D | egree-based Graph Con | structio | on. | | | | | | | | 147 |

List of Tables

| dergraduate students of the University of British Columbia along with the s of the time required by participants (of both types) to make their judgments | 316 un- | |
|---|-----------|----|
| of the time required by participants (of both types) to make their judgments | tatistics | |
| | in mil- | |
| liseconds. | | 71 |

List of Figures

| 1.1 | Example of a network of six agents (social sensors) aims to estimate a parameter (state of nature) interactively: each edge depicts a communication link between two | |
|-----|--|---------|
| | sensors. | 4 |
| 1.2 | Tracking the underlying state of nature using a Markov-modulated random graph as | |
| 1.3 | Main results and organization of thesis. | 0 16 |
| 2.1 | Example of constrained information flow network, $S = 2$ and $K = 3$. Circles repre- | |
| | sent a social group at a specific time indexed by (2.4) in the social network and each | |
| | edge depicts a communication link between two nodes | 31 |
| 2.2 | The conditional mean of the state of nature given the observation in estimation with | |
| | optimal mis-information removal algorithm compared to the full information net- | |
| | work | 33 |
| 2.3 | Comparison of the mean squared errors of the estimates obtained by optimal | |
| | mis-information removal algorithm, Bayesian estimator in full information flow | |
| | network (free of mis-information), and standard Bayesian estimator in constrained | |
| | information flow network (with mis-information propagation) | 35 |
| 2.4 | Comparison of the conditional mean of the state of nature x given the observa- | |
| | tions obtained by sub-optimal mis-information removal algorithm, optimal mis- | |
| | information removal algorithm (knowing the exact information flow graph), Bayesian | |
| | estimator in full information flow network (free of mis-information), and standard | |
| | Bayesian estimator in constrained information flow network (with mis-information | |
| | propagation) in "accurate estimation" scenario ($\beta = 0.2$). | 36 |
| 2.5 | Comparison of the mean squared errors of the estimates obtained by sub-optimal | |
| | mis-information removal algorithm, optimal mis-information removal algorithm (know | - |
| | ing the exact information flow graph), Bayesian estimator in full information flow | |
| | network (free of mis-information), and standard Bayesian estimator in constrained | |
| | information flow network (with mis-information propagation) in "accurate estima- | |
| | tion" scenario ($\beta = 0.2$). | 37 |
| | | |

| 2.6 | Comparison of the conditional mean of the state of nature x given the observa- | |
|-----|--|----|
| | tions obtained by sub-optimal mis-information removal algorithm, optimal mis- | |
| | information removal algorithm (knowing the exact information flow graph), Bayesian | |
| | estimator in full information flow network (free of mis-information), and standard | |
| | Bayesian estimator in constrained information flow network (with mis-information | |
| | propagation) in "inaccurate estimation" scenario ($\beta = 0.8$). | 38 |
| 2.7 | Comparison of the the mean squared errors of the estimates obtained by sub-optimal | |
| | mis-information removal algorithm, optimal mis-information removal algorithm (know- | - |
| | ing the exact information flow graph), Bayesian estimator in full information flow | |
| | network (free of mis-information), and standard Bayesian estimator in constrained | |
| | information flow network (with mis-information propagation) "inaccurate estima- | |
| | tion" scenario ($\beta = 0.8$) | 39 |
| 0.1 | | |
| 3.1 | Two examples of multi-agent social learning in social networks: (1) target localiza- | 16 |
| 2.2 | tion, and (ii) online rating and review systems. | 46 |
| 3.2 | Example of communication graph, with two agents ($S = 2$) and over three event | |
| | epochs ($K = 3$). The arrows represent exchange of information regarding actions | 47 |
| | taken by agents. | 47 |
| 3.3 | Protocol 1: Constrained social learning in social networks described in Section 3.1.1. | 40 |
| | As a result of random (unknown) communication delays, data incest arises. | 49 |
| 3.4 | Protocol 2: Idealized benchmark social learning in social networks. In this protocol, | |
| | the complete history of actions chosen by agents and the communication graph are | |
| | known. Hence, data incest does not arise. This benchmark protocol will be used to | |
| ~ - | design the data incest removal protocol. | 52 |
| 3.5 | Two examples of networks: (a) satisfies the topological constraint, and (b) does not | |
| | satisfy the topological constraint. | 57 |
| 3.6 | Data incest removal algorithm employed by network administrator in the state es- | |
| | timation problem over social network. The underlying state of nature could be ge- | |
| | ographical coordinates of an event (target localization problem) or reputation of a | |
| | social unit (online rating and review systems). | 58 |
| 3.7 | Three different communication topologies: (a) the communication graph with 41 | |
| | nodes, (b) agents interact on a fully interconnected graph and the information from | |
| | one agent reach other agents after a delay chosen randomly from $\{1,2\}$ with the | |
| | same probabilities, (c) star-shaped communication topology with random delay cho- | |
| | sen from $\{1,2\}$ | 59 |
| 3.8 | Actions of agents obtained with social learning over social networks in three differ- | |
| | ent scenarios described in Section 3.4 with communication graph depicted in Fig.3.7a. | 61 |

List of Figures

| 3.9 | Mean of the estimated state of nature in the state estimation problem with social | |
|------|---|----|
| | learning over social networks in three different scenarios described in Section 3.4 | |
| | with communication graph depicted in Fig.3.7a | 61 |
| 3.10 | Actions of agents obtained with social learning over social networks in three differ- | |
| | ent scenarios described in Section 3.4 with communication graph depicted in Fig.3.7c. | 62 |
| 3.11 | Mean of the estimated state of nature in the state estimation with social learning | |
| | over social networks in three different scenarios described in Section 3.4 with com- | |
| | munication graph depicted in Fig.3.7c. | 62 |
| 3.12 | Actions of agents obtained with social learning over social networks in three differ- | |
| | ent scenarios described in Section 3.4 with communication graph depicted in Fig.3.7b. | 63 |
| 3.13 | Mean of the estimated state of nature in the state estimation problem with social | |
| | learning over social networks in three different scenarios described in Section 3.4 | |
| | with communication graph depicted in Fig.3.7b | 63 |
| 3.14 | Actions of agents obtained with social learning over social networks in three differ- | |
| | ent scenarios described in Section 3.4 with arbitrary communication graph | 64 |
| 3.15 | Mean of the estimated state of nature in the state estimation problem with social | |
| | learning over social networks in three different scenarios described in Section 3.4 | |
| | with arbitrary communication graph | 64 |
| 3.16 | Mean squared error of estimates (of state of nature) obtained with social learning | |
| | with communication graph depicted in Fig.3.7a | 66 |
| 3.17 | Mean squared error of estimates (of state of nature) obtained with social learning | |
| | with communication graph depicted in Fig.3.7b (complete fully interconnected graph). | 66 |
| 3.18 | Mean squared error of estimates (of state of nature) obtained with social learning | |
| | with communication graph depicted in Fig.3.7c (star-shaped communication graph). | 67 |
| 3.19 | Mean squared error of estimates (of state of nature) obtained with social learning | |
| | with arbitrary communication graph | 67 |
| 3.20 | Two arrays of circles were given to each pair of participants on a screen. Their | |
| | task is to interactively determine which side (either left or right) had the larger | |
| | average diameter. The partner's previous decision was displayed on screen prior | |
| | to the stimulus. | 69 |
| 3.21 | Actions of two participants in a group at different trials in one experiment | 70 |
| 3.22 | Two scenarios where data incest arose in our experimental studies | 71 |
| 3.23 | Social learning with data incest that is exercised by groups of students who were | |
| | asked to perform a conceptual task in our experimental study | 72 |
| 3.24 | Actions of two participants in a group at different epochs. Participant 1 can be | |
| | considered as an internal node and Participant 2 can be viewed as a boundary node. | 73 |

List of Figures

| 3.25 | Optimal weights (which depends on the topology of the communication graph) and set of available public belief are computed in separate units. The user, then, can compute the most updated data-incest free network belief. | 74 |
|------|---|-----|
| 4.1 | The power law component for the non-Markovian random graph generated accord- | |
| | ing to Procedure 4.5 obtained by (4.11) for different values of p and q in Proce- | |
| | dure 4.5 | 89 |
| 4.2 | Degree distribution of the duplication-deletion random graph satisfies a power law. | |
| | The parameters are specified in Example 4.6.1 of Section 4.6. | 100 |
| 4.3 | Degree distribution of the fixed size duplication-deletion random graph. The pa- | |
| | rameters are specified in Example 4.6.2 of Section 4.6. | 100 |
| 4.4 | Degree distribution of the fixed size duplication-deletion random graph satisfies a | |
| | power law when N_0 is sufficiently large. The parameters are specified in Exam- | |
| | ple 4.6.2 of Section 4.6. | 101 |
| 4.5 | The estimates obtained by SA algorithm (4.14) follows the expected PMF precisely | |
| | with no knowledge of the Markovian dynamics. The parameters are specified in | |
| | Example 4.6.3 | 101 |
| 4.6 | The average degree of nodes (as a measure of connectivity) of the fixed size Markov- | |
| | modulated duplication-deletion random graph obtained by Procedure 4.5 for differ- | |
| | ent values of the probability of connection, p , in Algorithm 4.5. The parameters are | |
| | specified in Example 4.6.4 of Section 4.6. | 102 |
| 4.7 | Trace of the covariance matrix of scaled tracking error, trace $(\Sigma(\theta))$, versus the av- | |
| | erage degree of nodes as a measure of connectivity of the network. The parameters | |
| | are specified in Example 4.6.3 of Sec.4.6. | 103 |
| 4.8 | Trace of the covariance matrix of the scaled tracking error, trace $(\Sigma(\theta))$, versus the | |
| | order of delay in the searching problem as a measure of searchability of the network. | |
| | The parameters are specified in Example 4.6.3 of Sec.4.6. | 103 |

Acknowledgments

I owe my sincere gratitude to my academic advisor Prof. Vikram Krishnamurthy. I am deeply indebted to him for every bit of guidance, encouragement and support that I received during the completion of this work. This feat was not possible without his consistent support, fruitful discussions, valuable feedbacks, and constructive suggestions. I hope that someday I could be as enthusiastic, and energetic as him and be able to advise an audience as well as he can.

I would like to express my thanks to Prof. George Yin for his time, effort, suggestions and helpful comments during our collaboration in the past years which resulted in much improvement in our work. I would also like to extend my thanks to my friends and colleagues in the Statistical Signal Processing Lab for creating a friendly and productive environment.

Furthermore, heartfelt and deep thanks go to my family, and particularly my parents, for all the love, patience, sacrifices, assistance, and support. They have always encouraged me and given their unconditional support throughout my studies at the University of British Columbia. Without their support I could not have completed this thesis and it is to them that I dedicate this thesis.

Maziyar Hamdi UBC, Vancouver October 2015

Dedication

To my beloved family: Mohammadreza, Haydeh, and Mazda.

Introduction

1.1 Overview

Social networks are crucial to the modern society: they permeate our social, personal, and economic lives. They have changed the way that people connect and communicate, vote, select items to purchase, pick hotels to stay in, and adapt a new technology or behavior. For example, in the new *"social media"* era, Angry Birds¹ required only 35 days to obtain 50 million users. For comparison purposes and to get a better idea, radio reached the same milestone in 38 years, telephone in 75 years [5]. Social networks facilitate transmission of information, communication, and interaction among people and have grown steadily in size, complexity, and importance [28, 80]. One of the reasons behind this huge growth of social networks, both in science and society, is the concept of a *"network"* as a structural pattern of interactions among social actors. Such a social structure appears on a wide range of contexts in sociology, biology, economics, computer science and electrical engineering, whenever a set of dyadic ties between objects are observed. In all these topics, an ability to model and analyze those–often huge and complex–social structures, is fundamental for the computational purposes and it answers questions about the behavior of the social actors–individually as an agent or collectively as a team–in a data-driven manner.

As a result of the enormous effects and applications of social networks in community and in science, there has been a growing interest in modeling and analysis of social networks over the past decade. Due to the constraints imposed by the structure of networks and the nature of humans (as interacting sensors), research in social networks requires borrowing techniques from complex networks (dynamics of random graphs) and social analysis (mostly used in the areas of economics and sociology)[80, 146]. Statistical inference using social networks is an area that has witnessed marvelous progress recently. Such systems comprising of humans acting as sensors, also called participatory sensing, has received wide attention recently in the research communities of computer science, economics, marketing, social sciences and electrical engineering. The proliferation of social media such as real-time microblogging services (Twitter²) and online rating and review systems (Yelp) make real time sensing of social activities, social patterns, and behavior easier. In

¹https://www.angrybirds.com.

²About 6 thousand tweets are sent on average in a second. The tweet-per-second number on US Presidential election day in 2012, was 15 thousand tweets per second resulting in 500 million tweets in the day. Twitter can be considered as a real-time human-based sensor of social situations.

signal processing community, the term *social sensor* is used to denote an agent (or a group of agents) that provides information about its environment (state of nature) to others (possibly via social media channel) after interaction with other agents in a social network. Examples of such social media channels include Twitter, Facebook, online rating and review systems like Yelp and Tripadvisor, e-commerce platforms such as Amazon. The ability of social sensors in sensing social activities, patterns and behaviors is beyond physical sensors. For example, level of satisfaction of attendees of a concert revealed by sentiment analysis on Facebook statuses and tweets are impossible to predict using physical sensors, or as opposed to physical sensors, social sensors can be used to find out how the quality of food is in a restaurant from the reviews on online rating system such as Yelp [91, 92].

Statistical signal processing in social networks (inference using social sensors) appears in enormous range of applications across different industry sectors including marketing and advertisement, health and medicine, and financial technology. For example, [38, 101] used content of tweets to provide Geo-location services which are useful in targeting and event advertising. Other examples include detecting influential users with applications in marketing [144], localization of natural disasters [132], and predicting stock markets [26]. It is shown in [11] that a simple model built from the rate of tweets casted about particular topics can outperform market-based predictors.

With the above applications of social sensors, there is a strong motivation to develop a set of mathematical models and algorithmic tools to understand the effects of interactions among social sensors on estimation problem (interactive sensing). The majority of this thesis is devoted to development of algorithms and procedures that are aimed at multi-agent estimation, tracking, and decision making where agents acts as social sensors of an underlying state of nature in the presence of uncertainty. Such problems are non-standard in two ways: First, in social networks, agents interact with and influence other agents. For example, ratings posted on online rating and review systems strongly influence the behavior of individuals³. Such interactions can result in non-standard information patterns as a results of constraints imposed and correlations introduced by the structure of the underlying social network (communication topology among social sensors). Second, due to privacy reasons and time constraints, social sensors typically do not reveal raw observations of the underlying state of nature. Instead, they reveal their beliefs (opinions), or actions (tweet/re-tweet of common trends, thumbs-up on rating and review systems, purchase an item on e-commerce platforms) which can be viewed as a quantized version of their knowledge about state of nature formed by raw measurements and interactions with other social sensors. These together with the uncertainty involved in observations of state of nature, result in non-standard estimation and tracking problem which are the major topics of this thesis. The unifying theme of this thesis is to develop a set of theory and methods for statistical signal processing on graphs (social sensors) which involves adaptive filtering and stochastic approximation, dynamics of random graphs, multi-agent Bayesian estima-

³It is reported in [79] that 81% of hotel managers regularly check Tripadvisor reviews. It is reported in [109] that a one-star increase in the Yelp rating maps to 5-9 % revenue increase.

tion, and social learning to understand how sensors interact. We employ *social learning* [16, 24, 35], *graph theoretic tools*, and *stochastic approximation* [99, 157] as useful mathematical abstractions for modeling the interaction of social sensors. The rest of this section is devoted to an overview of these topics along with motivation and research goals that have been addressed in this thesis.

1.1.1 Bayesian Estimation over Social Networks

Bayesian filtering, which is a recursive form of the famous Bayes' rule, has been used extensively in the traditional signal processing literature for logical inference on uncertain parameters from a prior knowledge and new observations[21]. What statisticians and mathematicians call "Bayesian theory" was originally developed by Thomas Bayes in a famous paper which was presented at a meeting of the Royal Society of London [18, 138]. The well-known Bayes theorem describes the fundamental probability law in order to perform logical inference. He states[138]

"If there be two subsequent events, the probability of the second $\frac{b}{N}$ and the probability of both together $\frac{P}{N}$, and it being first discovered that the second event has also happened, from hence I guess that the first event has also happened, the probability I am right is $\frac{P}{b}$."

Bayesian inference is devoted to applying Bayes' rule to statistical inference in order to update the probability of a hypothesis as new observations are obtained [21, 125, 130]. Bayesian models for inference have been applied in decision theory, detection and estimation, communications theory, pattern recognition, machine learning and artificial intelligence, and filtering and parameter estimation [39, 64, 75, 105, 112].

Chapter 2 of this dissertation considers a multi-agent sensing problem where agents (social sensors) interact over a random graph and evaluate their belief (opinion) about an economic or a social parameter, namely state of nature. This state of nature can be, for example, quality of food in a restaurant, occurrence of earthquake, or Geo-location of a target. The evaluation of belief about state of nature can be made using Bayesian [1, 64] or non-Bayesian models [57, 81]. In this work, we focus on Bayesian inference in social networks. In the setup considered in Chapter 2, each agent acts a social sensor: (i) she records her observation about state of nature in uncertainty (by sampling from a conditional probability distribution), (ii) interact with other agents and receives estimates from other agents, (iii) updates her belief using recursive Baysian models and transmits her updated belief over the network, and this repeats on.

As a result of the recursive nature of the Bayesian models and the correlation introduced by the structure of the communication graph between agents, *"mis-information propagation"* also known as *data incest*⁴ arises in such networks. In a general sense, mis-information propagation involves in-

⁴These terms are used interchangeably throughout this dissertation.

advertent re-use of identical information (which are naively considered to be independent) in formation of belief about state of nature [71, 89]. The following example illustrates how mis-information can propagate and affect the estimates of social sensors in the multi-agent state estimation over graphs.

Example to Illustrate Mis-information Propagation:

Consider six social sensors that aim to estimate an underlying state of nature interactively with their communication topology illustrated in Figure 1.1. The graph in Figure 1.1 shows the information exchange protocol among social sensors, for example, the link between node 1 and node 3 shows that the estimation of node 1 is available at node 3. Each node records its own observation and combines it with the available estimates received from other nodes in the network (which are available due to the network structure) in order to form its estimates of state of nature. Then, the updated estimates are broadcasted over the network.



Figure 1.1: Example of a network of six agents (social sensors) aims to estimate a parameter (state of nature) interactively; each edge depicts a communication link between two sensors.

To understand what can go wrong with the above example, note that the estimate of node 1 is used at nodes 3 and 4, therefore, the estimates of these nodes are both functions of the estimate of node 1. Thus, if node 5 naively combines the information received from node 3 and 4, it would have double counted the estimate of node 1; this results in mis-information propagation.

Research Goals

As illustrated with the above example, mis-information propagation (also called data incest) arises in interactive sensing over networks due to the recursive nature of Bayesian estimators and the correlation imposed by the structure of the underlying communication graph among agents (social sensors). Mis-information propagation results in a bias in estimates of agents and over-confidence phenomena, i.e., the variance is underestimated. Therefore, in the presence of mis-information propagation, Bayesian estimators require careful design. Chapter 2 is devoted to design and development of algorithms that mitigate the effect of mis-information propagation in multi-agent estimation problem over the networks.

Remark 1.1.1. Our implicit assumption throughout Chapter 2 is that, agents do not directly broadcast their raw observations over the network. First, (private) raw observations of the other individuals are not typically available because of the privacy concerns or time constraints. For example, let $x \in (\{1,2,...,5\} \times \{1,2,...,5\})$ denote quality and affordability of a restaurant (x is a two dimensional vector). Assume that an individual in social network (social sensor) checks out the restaurant and based on her observations, she estimates the quality and cost of that restaurant (the estimated value of x). But at the time of her future social interactions, she usually does not provide (or even does not remember) details of the "raw" observations and she only shares her belief with others. Second, the dimension of observations, is typically much larger than the dimension of the state of nature. In the example of restaurant, the observation vector can include many elements such as quality of food, music, lighting, price of food, staff, neighborhood, cost of beverages, etc. Instead of broadcasting all these raw observations, it is more common in social networks to share only the beliefs about quality and cost of that restaurants.

1.1.2 Interactive Social Learning over Networks

Social learning which has been applied to understand, model, and predict the behavior of agents in economics, financial markets, political sciences, and social networks [16, 35, 36, 100] seeks to answer the following question: *How do actions (decisions) of other agents affect actions of subsequent agents?* Social learning model comprises of a set of agents seeking to estimate the underlying state of nature not only from their private observations, but also from the actions of previous agents. All agents know the structure of the model, they know that the action of each agent is a rational response to its private observations, and thus, convey information about state of nature. Social learning can be considered as the diffusion of (private) information about state of nature to all agents through the intercommunication of actions in a set of (finite, or infinite) agents [35].

When a human learns form another person's behavior, decision, or action, social learning occurs. In social learning, the actions of one agent affects the behavior of others, since they know that those actions are motivated by some type of information that other agents have about the state of nature. To better understand this, let's borrow the umbrella example from [35]. When someone sees other people going out with an umbrella, she also takes an umbrella without checking the weather forecast. That happens because people know that the actions or behavior of others have some information about the state of nature; this results in rational herding. Social learning is a useful approximation to ordinal human behavior. Classical social learning is used to model the behavior of expected cost minimizer agents. Also, social learning can be generalized to of risk averse minimizers, see [95, 96].

Chapter 3 considers a social learning model comprising of a set of agents (social sensors) that are interacting over a network to estimate an underlying state of nature. As opposed to the classical social learning model (where agents act once in a pre-determined order), in the social learning model considered in Chapter 3, the structure of social network dictates who interacts with whom.

We use social learning as a mathematical abstraction to model interactions between agents (social sensors) in interactive sensing problem. From a statistical signal processing point of view, the interactive sensing using social learning models–estimating state of nature via social sensors–is non-standard in two ways: First, agents are influenced by the rating of other agents, this can result in interesting phenomenon where rational agents can all end up making the same decision (herding and information cascades; [35]). Second, (and this effect is more complex), an agent might be influenced by his own rating leading to data incest (mis-information propagation). In the following, the main results of Chapter 3 is described briefly.

Research Goals

As is apparent from the above discussion, mis-information can propagate in interactive sensing with social learning due to the correlations imposed by the structure of the underlying social networks (existence of multiple paths between agents in the graph which represents the underlying social network) and the recursive nature of Bayesian inference, see Section 1.1.1. To give an example, suppose an agent wrote a poor rating of the restaurant on a rating and review site. Another agent is influenced by this rating and also gives the restaurant a poor rating. The first agent visits the rating and review again and finds out that another agent has also given the restaurant a poor rating—this double confirms his rating and he enters another poor rating. In a fair system, the first agent should have been aware that the rating of the second agent was influenced by his rating—so that first agent has effectively double counted his first rating by casting the second poor rating. Mis-information propagation cause the over-confidence problem and results in a bias in the estimate of state of nature.

Motivated by online rating and review systems, our aim is to manage mis-information propagation problems associated with interactive sensing using social learning. In particular, our goal is to design and develop a protocol for the administrator of a rating and review system such that it can automatically maintain a fair (data incest free) rating and review system which results in a system with a higher trust rating.

1.1.3 Tracking Degree Distribution of Social Networks

Tracking a time-varying parameter that evolves according to a finite-state Markov chain (state of nature) is a problem of much interest in signal processing [20, 63, 153]. In the context of social networks, this parameter can be, for example, the level of happiness in a community, the tendency of individuals to expand their networks, the strength of social links between individuals, or searchability of network which cannot be sensed by physical sensors. In such cases, social sensors can go beyond physical sensors and can be used to track those parameters of social networks. A social network with a large number of individuals can be viewed as an interactive sensing tool to obtain information about individuals or state of nature; this is a social sensor. Motivated by social network

applications, a social sensor based framework is presented in Chapter 4 to track the degree distribution of Markov-modulated dynamic networks whose dynamics evolve over time according to a finite-state Markov chain.

The question that may arise here is: "What is the importance of degree distribution of a social network?" The most important parameter of a network that characterizes its structure is the degree distribution. It yields useful information about the connectivity of the random graph [10, 86, 116]. For example, if a majority of nodes in the random graph have relatively high degrees, the graph is highly connected and a message can be transferred between two arbitrary nodes with shorter paths. However, if a majority of nodes have smaller degrees then for transmitting a message throughout the network, longer paths are needed, see [80]. The degree distribution can further be used to investigate the diffusion of information or disease through social networks [108, 146]. The existence of a "giant component"⁵ in complex networks can be studied using the degree distribution of the graph that models that social network. The size and existence of a giant component has important implications in social networks in terms of modeling information propagation and spread of human disease, see [62, 115, 118]. The degree distribution is also used to analyze the "*searchability*" of a network. The "search" problem arises when a specific node in a network faces a problem (request) whose solution is at other node, namely, destination (e.g., delivering a letter to a specific person, or finding a web page with specific information) [4, 146]. The searchability of a social network [146] is the average number of nodes that need to be accessed to reach the destination. Degree distribution is also used to investigate the robustness and vulnerability of a network in terms of the network response to attacks on its nodes or links [33, 76]. The papers [148, 149] further use degree-dependent tools for classification of social networks.

Chapter 4 considers a dynamic social network where the interactions between nodes evolve over time according to a Markov process that undergoes infrequent jumps (the state of nature). An example of such social networks is the friendship network among residents of a city, where the dynamics of the network change in the event of a large festival. In this chapter, we propose *Markov-modulated random graphs* to mimic social networks where the interactions among nodes evolve over time due to the underlying dynamics (state of nature). For example, state of nature can be level of happiness in the society which is impossible to measure using participatory sensors. Here, social networks can be used as social sensors for tracking the underlying state of nature. That is, using noisy measurements of the degree distribution of the network, the jumps in the underlying state of nature can be tracked.

⁵A giant component is a connected component with size O(n), where *n* is the total number of vertices in the graph. If the average degree of a random graph is strictly greater than one, then there exists a unique giant component with probability one [41], and the size of this component can be computed from the expected degree sequence.



Figure 1.2: Tracking the underlying state of nature using a Markov-modulated random graph as a social sensor.

Research Goals

Chapter 4 considers a Markov-modulated duplication-deletion random graph where, at each time instant, one node can either join or leave the network with probabilities that evolve according to the realization of a finite state Markov chain (state of nature). This chapter deals with the following questions: *How can one estimate the state of nature using noisy observations of nodes' degrees in a social network?* and *How good are these estimates?* By tracking the degree distribution of a Markov-modulated random graph, we can design a social sensor to track the underlying state of nature using the noisy measurements of nodes' connectivity; see Figure 3.20.

Chapter 4 comprises of two results. First, motivated by social network applications, we analyze the asymptotic behavior of the degree distribution of the Markov-modulated random graph. From this degree distribution analysis, we can study the connectivity of the network, the size and the existence of a large connected component, the delay in searching such graphs, etc. [62, 80, 115, 118]. Second, a stochastic approximation algorithm is presented to track the empirical degree distribution as it evolves over time. We further show that the stationary degree distribution of Markov-modulated duplication-deletion random graphs depends on the dynamics of such graphs and, thus, on the state of nature. This means that, by tracking the empirical degree distribution, the social network can be viewed as a social sensor to track the state of nature. The tracking performance of the proposed stochastic approximation algorithm is analyzed in terms of mean square error. A functional central limit theorem is further presented for the asymptotic tracking error.

1.2 Main Contributions

In the subsequent sections, a brief summary of major novel contributions of the chapters that constitute this thesis is provided in order that they appear in the thesis. More detailed description of the contributions and findings of each chapter is provided in individual chapters.

1.2.1 Bayesian Estimation over Social Networks

As briefly described in Section 1.1.1, Chapter 2 considers multi-agent Bayesian estimation problem with constraints imposed by the structure of the underlying social network. In such problems, as a result of the recursive nature of Bayesian estimators and the correlation imposed by the communication topology of social sensors, mis-information propagation arises.

The main contributions of Chapter 2 are summarized below:

- 1. Mis-information propagation problem in interactive-sensing problem over social networkswhere agents transmit their beliefs about state of nature instead of raw observations over the network-is mathematically formulated using a family of directed acyclic graphs.
- 2. A necessary and sufficient condition on the information flow graph⁶ is presented for the exact mis-information removal problem. It is shown in Section 2.3 that under Constraint 2.3.1, mis-information associated with estimates of agents can be completely removed.
- 3. An optimal information aggregation algorithm is proposed for multi-agent estimation problem over networks which mitigates the mis-information associated with estimates of agents (social sensors) when the information flow graph is known.
- 4. A sub-optimal mis-information removal algorithm is presented for scenarios where the the information flow graph is not completely known.

1.2.2 Interactive Social Learning over Networks

Motivated by online rating and review systems, Chapter 3 considers social learning as a mathematical abstraction to model the interactions among agents (social sensors) in state estimation problem using interactive sensing. Agents record their own private observations, then, update their private beliefs about the state of nature using Bayes' rule. Based on their belief each agent, then, chooses an action (rating) from a finite set and transmits this action over the social network. An important consequence of such social learning over a network is the ruinous multiple re-use of information

⁶Information flow graph is a directed acyclic graph that models the flow of information among social sensors, for example, a directed edge from Sensor 1 to Sensor 2 means that the information– or beliefs in the context of Chapter 2–of Sensor 1 is available at Sensor 2, see Section 2.2 for more details.

known as data incest (or mis-information propagation). In this chapter, the data incest management problem in social learning context is formulated on a family directed acyclic graphs.

The main contributions of Chapter 3 are summarized below:

- 1. A social learning model is presented to mimic the behavior of agents in online rating and review systems that aim to estimate a state of nature (for example quality of a restaurant on Yelp).
- A fair rating and review protocol is presented and the criterion for achieving a fair rating is defined. This protocol is used as a benchmark in the data incest management problem in social learning over social networks.
- 3. An automated incest removal protocol is developed for the administrator of online rating and review system, to deploy such that the system maintains a fair rating of its entities. Such algorithm can easily be applied to any interactive-sensing system that involves transmission of actions and Bayesian inference.
- 4. Necessary and sufficient conditions on the graph topology of social interactions between social sensors are presented to eliminate data incest.

1.2.3 Tracking Degree Distribution of Social Networks

Chapter 4 considers dynamical random graphs. The most important measure that characterizes the structure of a network (specially when the size of the network is large and the connections— adjacency matrix of the underlying graph—are not given) is the degree distribution of that network. The degree of a node in a network (also known as the connectivity) is the number of connections the node has in that network. In this chapter, motivated by social network applications, we consider a class of stochastic approximation algorithms to track a time-varying probability mass function that evolves according to a finite-state Markov chain whose transition matrix is close to identity. In the context of social network analysis, the time-varying probability mass function which we aim to track is the expected degree distribution of a dynamic random graph.

The main contributions of Chapter 4 are summarized below:

- A family of Markov-modulated duplication-deletion random graphs are introduced in Chapter 4 that mimic social networks where interaction among agents are varying over time according to realization of a finite-state Markov chain. We consider two categories of such graphs: (i) fixed size duplication-deletion random graph, and (ii) infinite duplication-deletion random graph.
- 2. An asymptotic degree distribution analysis is presented for the fixed size Markov-modulated random graph. In particular, it is shown that the expected degree distribution of such graphs

at each time can be computed in terms of the expected degree distribution at the previous time and the dynamics of the graph via a recursive equation.

- 3. We extend the degree distribution analysis to infinite random graphs and prove that the degree distribution of such graphs satisfy a *power law* with the exponent depending on the dynamics of the graph. An expression is presented to compute the power law exponent in terms of the dynamics of the duplication-deletion model.
- 4. Chapter 4, further, considers the adaptive estimation problem of degree distribution for a fixed size Markov-modulated duplication-deletion random graph given noisy observations. A stochastic approximation algorithm is presented for tracking the degree distribution as it evolves over time. In the following, the results related to the tracking performance of stochastic approximation algorithm are presented.
 - *Mean square error analysis:* Using error bounds on two-time scale Markov chains and perturbed Lyapunov function methods, the asymptotic mean square error between the expected degree distribution and the estimates obtained via the stochastic approximation algorithm is computed.
 - *Weak convergence analysis:* We show that the asymptotic behavior of the stochastic approximation algorithm converges weakly to the solution of a switched Markovian ordinary differential equation.
 - *Functional central limit theorem for scaled tracking error:* Finally, Chapter 4 investigates the asymptotic behavior of the scaled tracking error of stochastic approximation algorithm. Similar to [94], it is shown that the interpolated scaled tracking error converges weakly to the solution of a switching diffusion process.
- 5. Chapter 4, further, investigates infinite (denumerable) duplication-deletion random graphs where the number of nodes in the graph (and so the support of degree distribution) is no longer fixed and increases over time. A Hilbert-space-valued stochastic approximation algorithm is proposed to track the degree distribution of the infinite graph with support on the set of non-negative integers. To study the tracking performance of such a Hilbert-space-valued stochastic approximation algorithm, limit system characterization and asymptotic analysis of scaled tracking error are provided.

1.3 Related Works

This section is devoted to the literature review of topics and advances in the fields related to this dissertation.

1.3.1 Bayesian Estimation over Social Networks

Bayesian inference deals with the problem of inferring knowledge about unknown parameters using Bayes' rule [21, 125, 130]. Bayesian theory has many application in decision theory, detection and estimation, communications theory, pattern recognition, machine learning and artificial intelligence, and filtering and parameter estimation [39, 58, 64, 75, 77, 105, 112]. For a comprehensive survey on Bayesian inference and estimation theory, we refer the interested reader to books [130, 133, 141]. Bayesian networks and different inference methods in Bayesian networks are investigated in [120]. A model of Bayesian social learning where agents receive a private information about state of nature and observe the actions of their neighbors is investigated in [83]. They proposed an algorithm to compute actions of agents on tree-based social networks and analyzed their algorithm in terms of efficiency and convergence.

There are several papers discussing the spread of information in social networks, see [7, 31, 36] for a comprehensive survey and tutorial on different methods for diffusion of information in social networks. Applications of gossip algorithms, which is a protocol based on communication of agents with their local neighbors, in signal processing are studied in [65]. A type of mis-information propagation in social networks caused by "influential" or "forceful" agents is investigated in [3]. Viral propagation of faulty information (for example, mis-information of swine flu) through social media channels is studied in [30, 119]. For the motivation of the mis-information problem addressed in this work, we refer to [29, 49, 78, 106, 128] in sensor networks. Data incest in sensor network context happens in distributed tracking systems where sensors locally integrate the estimates received from other sensors through a (possibly loopy) communication graph with random delays. The key requirement is to fuse estimates that share a common information set. An optimal solution for the case of connected tree networks by combining a decentralized information filter and a channel filter is presented in [49].

In this thesis, we consider mis-information propagation through a social network with arbitrary network topologies. Each agent records its observation of state of nature in presence of noise. We used a combination of graph theoretic tools and Bayesian estimation to remove the mis-information removal generated by different delays in links.

1.3.2 Interactive Social Learning over Networks

Social learning theory is used to investigate the learning behavior of agents in social and economic networks [2]. There are several papers in the literature discussing Bayesian models [1, 43, 92, 120] and non-Bayesian models [14, 48, 56, 57, 81] for social learning. Different models for diffusion of beliefs in social networks are presented in [36]. For a comprehensive survey on herding and information cascade in social learning, see [37]. Stochastic control with social learning for sequential change detection problems is considered in [87].

Mis-information in the signal processing literature refers to faulty or inaccurate information which is broadcasted unintentionally. A brief summary of works related to misinformation propagation and removal is presented in Section 1.3.1. Mis-information in the context of this chapter is motivated by sensor networks where the term "data incest" is used [89]. In multi-agent social learning in networks, data incest occurs when information (action) of one agent is double-counted by other agents (due to the lack of information about the topology of the communication graph); this yields to overconfidence. The overconfidence phenomena (caused by data incest) also arises in Belief Propagation (BP) algorithms [113, 123] which are used in various fields such as graphical models for learning, computer vision, and error-correcting coding theory. The aim of BP algorithms is to solve inference problems over graphical models such as Bayesian networks (where nodes represent random variables and edges depict dependencies among them) by computing a marginal distribution. BP algorithms require passing local messages over the graph (Bayesian network) at each iteration. These algorithms converge to the exact marginal distribution when the factor graph is a tree (loop free). But for graphical models with loops, BP algorithms are only approximate due to the over-counting of local messages [113, 152] (which is similar to data incest in multi-agent social learning)⁷.

In Chapter 2 and papers [89, 90], data incest is considered in a network where agents exchange their private belief states—that is, no social learning is considered. In a social network, agents rarely exchange private beliefs, they typically broadcast actions (votes) over the network. Motivated by trustable online rating and review systems, we consider data incest in a social learning context with social network structure where actions (or equivalently public belief of the social learning) are transmitted over the network. This is quite different from private belief propagation in social networks. Simpler versions of this information exchange process and estimation were investigated by Aumann [12] and Geanakoplosand and Polemarchakis [66]. The results derived in this chapter extend theirs.

Finally, the methodology of Chapter 3 can be interpreted in terms of the recent Time magazine article [145] which provides interesting rules for online rating and review systems. These include: (i) review the reviewers, (ii) censor fake (malicious) reviewers. The data incest removal algorithm proposed in this chapter can be viewed as "reviewing the reviews" of other agents to see if they are associated with data incest or not.

⁷There exists some similarities between BP and social learning in the sense that they are both systematic structures to perform Bayesian inference over graphs. However, they are not related in principle. While graphs represent social interactions among agents in social learning, graphical models in BP depict the conditional dependency between nodes (random variables)– they do not imply the actual communications, for more detail see[36].

1.3.3 Tracking Degree Distribution of Social Networks

With a large number of rational agents, social networks can be viewed as social sensors for extracting information about the world or people. For example, the paper [132] presents a social sensor (based on Twitter) for real-time event detection of earthquakes in Japan, namely, the target event. They perform semantic analysis on tweets (which are related to the target event) in order to detect the occurrence and the location of the target event (earthquake). Another example is the use of the social network as a sensor for early detection of contagious outbreaks [40]. Using the fact that central individuals in a social network are likely to be infected sooner than others, a social sensor is designed for the early detection of contagious outbreaks in [40]. The performance of this social sensor was verified during a flu outbreak at Harvard College in 2009—the experimental studies showed that this social sensor provides a significant additional time to react to epidemics in the society. Social sensing in the context where physical sensors present in mobile devices such as GPS or Bluetooth are used to infer social interactions is studied in [6, 32, 34, 54]. Here, we consider a scenario that a social network considered as a sensor of social interactions, human activities, or behavior and aims to track the degree distribution of a random graph via stochastic approximation algorithms.

Stochastic approximation algorithms have several applications in diverse areas such as system identification, control theory, adaptive filtering, state estimation, wireless communications, target tracking, change detection, and economics [20, 50–52, 93, 99, 110, 111, 150, 153]. The ubiquitous use of stochastic approximation algorithms is mainly due to their ability to track a time-varying unknown parameter of a system; this is called "tracking capability", see [20]. Tracking a time-varying parameter that evolves according to a finite-state Markov chain has several applications in target tracking [63], change detection [20], multi-user detection in wireless systems [153], and economics [93]. Tracking capability of regime switching stochastic approximation algorithms is further investigated in [154] in terms of the mean squared error. The interested reader is referred to [99] for a comprehensive development of stochastic approximation algorithms.

For the background and fundamentals on social and economic networks, we refer to [80]. Here, the related literature on dynamic social networks is reviewed briefly. The book [53] provides a detailed exposition of random graphs. The dynamics of random graphs are investigated in the mathematics literature, for example, see [41, 60, 104] and the reference therein. In [121], a duplication model is proposed where at each time step a new node joins the network. However, the dynamics of this model do not evolve over time. In [41], it is shown that the degree distribution of such networks satisfies a *power law*. In random graphs which satisfy the power law, the number of nodes with a specific degree depends on a parameter called "power law exponent" [82, 147]. A generalized Markov graph model for dynamic social networks along with its application in social network synthesis and classification is also presented in [149]. The degree distribution analysis of real-world

networks has attracted much attention recently, [8, 45, 55, 67, 85, 114, 118]. A large network dataset collection can be found in [103], which includes datasets from social networks, web graphs, road, internet, citation, collaboration, and communication networks. The paper [114] investigates the structure of scientific collaboration networks in terms of degree distribution, existence of giant component, and the average degree of separation. In the scientific collaboration networks, two scientists are connected if they have co-authored a paper. Another example is the degree distribution of the affiliation network of actors⁸, which is studied in [118] based on real data from IMDb. In [8], the structure and characteristics of three different online social networks (Cywork, Myspace, and Orkut) are investigated. The authors use snowball sampling method⁹ [22, 102] to estimate the degree distribution of the network when it is not possible to access all nodes in the network (specially when the size of the network is large). It is further shown in [55] that the degree distribution of email networks satisfies a power law.

Finally, different applications of social sensors in detection and estimation are investigated in [9, 40, 132]. The differences between social sensors, social sensing, and pervasive sensors along with challenges and open areas in social sensors are further presented in [88, 131].

1.4 Thesis Outline

In this section, we present the organization of this dissertation which is illustrated in Figure 1.3. The rest of this thesis is divided into two parts and four chapters as outlined below:

Motivated by different information diffusion patterns in interactive sensing over social networks, Part I considers multi-agent state estimation and learning problem over directed acyclic graphs in social networks and comprises of two chapters:

• Chapter 2 considers Bayesian estimation over directed acyclic graphs where agents transmit their beliefs about state of nature (the posterior distribution of state of nature given private observations and beliefs of other agents which are available due to the structure of the underlying social networks) instead of raw observations. It then formulates data incest (also known as mis-information propagation) that arises in such estimation problems using a graph-theoretic setup. It is, then, shown that under some necessary and sufficient conditions on the topology of the communication graph among agents, mis-information can completely be removed from the estimates of agents. Assuming that the communication graph is known, an optimal mis-information removal algorithm is proposed. We also provide a sub-optimal algorithm for reducing the effect of mis-information when the communication graph is not completely

⁸Collaboration network of movie actors.

⁹There are different methods of sampling a network, for example, link sampling, node sampling, and snowball sampling. For a complete survey, we refer to [102]. In link or node sampling, a given fraction of links or nodes are sampled. In snowball sampling method, one node is chosen randomly and the next samples are chosen from its neighbors.



Figure 1.3: Main results and organization of thesis.

known. This chapter is concluded with a numerical study that illustrates the excellent performance of the proposed algorithms.

• Motivated by online rating and review systems, Chapter 3 employs social learning to model the interactions among agents in a multi-agent estimation problem where the actions of agents are transmitted over the network (instead of raw observations or private beliefs). In such a setup, each agent—in order to estimate an underlying state of nature—chooses an action from a finite set of actions to minimize a local cost function and then transmits this action over the network. We give necessary and sufficient conditions on the graph topology of social interactions to eliminate data incest. A data incest removal algorithm is, then, proposed in this chapter such that the public belief of social learning (and, hence, the actions of agents) is not affected by data incest propagation. This results in an online rating and review system with a higher trust rating. This chapter then presents an actual psychology experiment that was conducted by our colleagues at the Department of Psychology of University of British Columbia in September and October, 2013, to illustrate social learning, data incest and social influence. Finally, numerical examples are provided to illustrate the performance of the proposed optimal data incest removal algorithm.

Motivated by applications of degree distribution in social network analysis and tracking a Markovian dynamics of graphs, Part II considers Markov modulated duplication-deletion random graphs and is comprised of one chapter:

• Chapter 4 considers a Markov-modulated duplication-deletion random graph where at each time instant, one node can either join or leave the network; the probabilities of joining or leaving evolve according to the realization of a finite state Markov chain. This chapter comprises of two inter-related research problems. First, motivated by social network applications, the asymptotic behavior of the degree distribution is analyzed. Second, a stochastic approximation algorithm is presented to track empirical degree distribution as it evolves over time. The tracking performance of the algorithm is analyzed in terms of mean square error and a functional central limit theorem is presented for the asymptotic tracking error. Chapter 4, then, presents a Hilbert-space-valued stochastic approximation algorithm that tracks a Markov-modulated probability mass function with support on the set of nonnegative integers. Finally, this chapter is concluded with some numerical example that illustrates the performance of tracking algorithms and corroborate the findings of this chapter.

Chapter 5 briefly outlines a summary of findings in these two parts and provides a direction for future research and development in the fields related to this dissertation.

A brief review of some graph theoretic definitions and tools that have been used throughout this dissertation is presented in Appendix A.

An important associated problem to numerical studies of social networks is how to actually construct random graphs via simulation algorithms. In particular, for large social networks, only the degree sequence is available, and not the adjacency matrix. (The degree sequence is a non-increasing sequence of vertex degrees.) Does a simple graph exist that realizes a particular degree sequence? How can all graphs that realize a degree sequence be constructed? Appendix B presents a discussion of these issues.

Part I

Estimation and Learning Over Directed Acyclic Graphs

Constrained Estimation Over Random Graphs

2.1 Introduction

This chapter deals with Bayesian estimation problem over networks and considers a social network where each group of individuals use received information from the other social groups and employs Bayesian models of information aggregation to evaluate their belief about state of nature. In this context, each group of individuals form a social sensor of a an economic or a social parameter. The process of exchanging information between social groups is crucial for individuals to evaluate their beliefs. An important parameter that characterizes how the belief evolves is the delay in this information exchange. This delay can be extrinsic - individuals take different amounts of time to form beliefs and communicate them, or intrinsic to the network; highly connected nodes exchange information faster compared to nodes that have fewer connections. The most important consequence of this delay is mis-information (or incest) propagation as we will explain shortly. Let us first formulate the observation and information exchange using a graph-theoretic notation.

State of Nature: Let *x* represent a state of nature that individuals in the social network aim to estimate such as quality of a restaurant. Assume that *x* belongs to a finite set $\mathbf{X} = \{\overline{x}_1, \overline{x}_2, ..., \overline{x}_N\}$. Here, \overline{x}_i (for $1 \le i \le N$) is in \mathbb{R}^d or \mathbb{R}^{+d} or \mathbb{N}^d where *d* is a positive integer number. Assume that *x* has prior distribution π_0 .

Observation Protocol: To estimate x, each individual in the social network obtains an M-dimensional observation vector where M is a positive integer number. To simplify the analysis, we assume that the set of individuals in the social network is partitioned into S social groups such that within each social group individuals record the same observations. At time k, the noisy observation of social group s, $z_{|s,k|}$ has conditional probability distribution

$$p(z_{\lfloor s,k \rfloor} \le \overline{z} | x = \overline{x}_i) = \sum_{z \le \overline{z}} B_{iz}, \quad 1 \le i \le N.$$

$$(2.1)$$

Here \sum_{z} denotes integration with respect to Lebesgue measure (in which case B_{iz} is the conditional probability distribution function) or the counting measure (in which case B_{iz} is the conditional probability mass function). Assume that z_n given x for different values of "s" and "k" are independent

random variables with respect to *s* and *k*. Each social group *s* combines its private observation $z_{\lfloor s,k \rfloor}$, with information received from other groups in social network to update its belief about state of nature *x*. Then, it communicates this updated belief to other groups in the social network.

Information Exchange Protocol: Let $G_{\lfloor s,k \rfloor} = (V_{\lfloor s,k \rfloor}, E_{\lfloor s,k \rfloor}), k = 1, 2, ..., s = 1, 2, ..., S$ denote a sequence of time-dependent directed graphs of information flow in the social network until and including time k. Here $V_{\lfloor s,k \rfloor}$ denotes the set of vertices,

$$V_{|s,k|} = \{(s,k') | k' \le k, s \in \{1, 2, \dots, S\}\}$$
(2.2)

and $E_{\lfloor s,k \rfloor} \subseteq V_{\lfloor s,k \rfloor} \times V_{\lfloor s,k \rfloor}$ is the set of edges which depicts the connections between vertices in $G_{\lfloor s,k \rfloor}$. For example if $((s,k'), (s',k'')) \in E_{\lfloor s,k \rfloor}$, it means that the information from social group *s* at time *k'* is available at social group *s'* at time *k''* $(k' \leq k'' \leq k)$. Each social group uses Bayesian model to estimate the underlying state of nature *x*.

As a result of the recursive nature of Bayesian estimators, mis-information propagation can arise in a social network with the above information exchange protocol. For example, assume that the estimates of social group 1 at time 1, $\theta_{\lfloor 1,1 \rfloor}$, reach social group 2 at time 2. Also suppose the estimates from social group 2, $\theta_{\lfloor 2,2 \rfloor}$, reach social group 1 at time 3. Since social group 2 used $\theta_{\lfloor 1,1 \rfloor}$, the estimate generated by social group 2 is a function of the $\theta_{\lfloor 1,1 \rfloor}$. Therefore, if social group 1 naively combines the estimate of social group 2 received at time 3, $\theta_{\lfloor 2,2 \rfloor}$, with its own private estimates, it would have double counted its estimate at time 1, $\theta_{\lfloor 1,1 \rfloor}$. In the above graph theoretic notation, we can depict graph $G_{\lfloor 2,3 \rfloor}$ as

where the two-tuples denote vertices defined in (2.2) and the arrows denote edges of directed acyclic graph. The fact that there exists two distinct paths between (1,1) and (1,3) in the graph of (2.3) shows that information in (1,1) is double counted leading to mis-information propagation. As the above example shows, the mis-information (rumor) propagation can be viewed as the destructive re-use of observation information. It leads to an overconfidence phenomenon i.e the variance is underestimated. The recursive nature of Bayesian estimation requires careful design to cope with the possible ruinous re-use of information. In more realistic problems considered in this chapter, there are multiple groups in social network (and thus an arbitrarily complex network topology) together with random delays in the network. For such cases, mis-information management is a non-trivial problem.

2.1.1 Chapter Goals

A network of social groups is considered in this chapter that aim to estimate the underlying state of nature *x*. Before proceeding, let us introduce the following scalar index *n* instead of $\lfloor s, k \rfloor$ for the sake of notational simplicity:

$$n \triangleq s + S(k-1), \quad s \in \{1, \dots, S\}, \ k \in \{1, 2, 3, \dots\}.$$
 (2.4)

Notice that n is a composite of time k and social group s. Subsequently, we will refer to n as a "node" of a time dependent graph namely information flow graph. This estimation problem can be expressed in the following abstract form:

Estimate x with prior π_0 subject to:

$$\begin{cases}
G_n = (V_n, E_n) \text{ is given.} \\
z_n \sim B_{iz}, \quad x = \overline{x}_i, \quad \text{(observation process)} \\
\theta_n = \mathscr{A}(\Theta_n, z_n), \quad \text{(filter constraint)}
\end{cases}$$
(2.5)

Here, Θ_n denotes the set of beliefs from nodes (social groups at previous times) available at node n (social group s at time k) which depends on the information flow network G_n . Let θ_n denote the posterior distribution¹⁰ of x given Θ_n and z_n . In 2.5, \mathscr{A} denotes the algorithm used by each node to update the belief θ_n . The aim of this chapter is to construct the information aggregation algorithm \mathscr{A} such that estimates θ_n are not affected by mis-information propagation. If \mathscr{A} is not constructed properly, then mis-information can propagate in social network as explained earlier in this chapter. Thus, from an abstract point of view, mis-information removal can be interpreted as optimal Bayesian estimation on a directed acyclic graph with information exchange constraints. This chapter aims to address the following questions:

- 1. *Existence Problem*: Under what constraints on the information flow, is complete mis-information removal possible?
- 2. *Design Problem*: Synthesize an algorithm \mathscr{A} such that mis-information propagation is prevented.
- 3. *Reconstruction Problem*: If the information flow graph, G_n , is not completely known at each time, design an algorithm to mitigate the mis-information propagation.

¹⁰For some distributions, instead of transmitting the posterior distribution, it is sufficient to broadcast the sufficient statistic. In the finite case, the sufficient statistic and posterior distribution are similar.
2.1.2 Main Results and Organization of Chapter

This chapter considers mis-information propagation through a social network with arbitrary network topologies. Each social group records their observation of state of nature with any arbitrary aposteriori probability distribution and any arbitrary observation noise. The recursive nature of Bayesian estimation in decentralized fusion requires careful design to cope with the possible re-use of information such that the estimates θ_n are equal to mis-information free estimates of the optimal scenario which is described in more detail in Section 2.2.2. In more realistic problems considered in this chapter, there are multiple groups together with random delays in the network. For such cases, mis-information management is a non-trivial problem. A combination of graph theory and Bayesian estimation is employed to remove the mis-information removal generated by different delays in links. The rest of the chapter is organized as follows:

- We represent information flow in a social network by a family of directed acyclic graphs in Section 2.2. The communication among social groups is modeled by information exchange Protocol 2.1 where mis-information propagation may arise. Information exchange Protocol 2.2 is introduced to benchmark against Protocol 2.1. From the benchmark Protocol 2.2, the mis-information removal algorithm can be specified.
- Section 2.3 presents a necessary and sufficient condition (called Constraint 2.3.1) on topology of the network that guarantees optimal, mis-information free estimates. It is shown that with the full knowledge of information flow graph, Constraint 2.3.1 leads to an algorithm for exact mis-information removal using optimal Bayesian estimation defined in Section 2.3.
- A sub-optimal algorithm is proposed in Section 2.4 to remove the mis-information associated with estimates of social groups when the information flow graph is not completely known at each time.
- Numerical results that show the effect of mis-information propagation and also the excellent performance of the proposed mis-information removal algorithms are presented in Section 2.5.

2.2 Modeling Information Flow in Social Networks

Section 2.1 outlined the goals and main results of this chapter and informally described *constrained estimation* over social networks. In this section, first, using a graph-theoretic notation, the following types of communication protocols are presented:

• *Constrained information flow protocol:* This protocol mimics information exchange, and inference in a social network where beliefs are communicated among groups (nodes). As stated in Section 2.1, mis-information propagation arises in this protocol due to the abusive repetition of information.

• *Full information flow protocol:* This protocol is an ideal (and, thus, impractical) communication protocol that prevents mis-information propagation. To devise an algorithm to mitigate mis-information propagation, this protocol is used as a benchmark against the constrained information flow protocol, as we explain shortly.

Then, we assert via Theorem 2.2.1 that the flow of information in a social network can be represented by a family of time dependent *Directed Acyclic Graphs* (DAGs). Some essential graph theoretic tools that will be used to formulate the mis-information propagation problem are outlined in Appendix A.

2.2.1 Constrained Information Flow Protocol

This protocol refers to a social network where nodes aim to estimate an underlying state of nature As described in Section 2.1, instead of raw observations, the posterior distribution of state of nature (beliefs) are broadcasted over the network. Due to the information exchange constraint in this protocol, the complete history of beliefs are not available at each node. It is in such a constrained information flow network that mis-information propagation arises.

The information exchange protocol in constrained estimation over social networks described in Section 2.1 can be summarized as following:

Protocol 2.1 Constrained Information Flow Network Protocol at each node *n*

Step 1. Observation: Node *n* (social group *s* at time *k*) records its private observation vector z_n according to (2.1), that is,

$$z_n \sim B_{iz}, \quad x = \overline{x}_i,$$

where n = s + S(k - 1), see (2.4).

Step 2. Interaction with other social groups: Node *n*, then, accesses the network for beliefs from other social groups at previous time instants Θ_n .

Step 3. Mis-information removal and Bayesian data fusion: Node *n* uses mis-information removal algorithm together with Bayesian data fusion to combine Θ_n with its private observation z_n and updates its belief θ_n .

Step 4. Transmit the updated belief: Node *n*, then, broadcasts the updated belief over the network.

Remarks: We assume a reasonable degree of flexibility that each node deploys for broadcasting its information over the network. In Step 2 above we assumed for simplicity (to avoid collision of information) that only one social group is allowed to transmit information at each time instant.

2.2.2 Benchmark Full Information Flow Protocol

The goal of this chapter is to solve estimation problem (2.5) subject to the information exchange Protocol 2.1. We now describe an idealized (and therefore impractical) Protocol 2.2 that will be used as a benchmark against Protocol 2.1. In the benchmark protocol, we assume that instead of transmitting posterior distribution θ_n , each node transmits its own private observations z_n and all raw observations received over the network. In this protocol, since each node has the entire available observation history from previous nodes, there is no room for mis-information propagation, i.e., there is no chance for inadvertent re-use of private observations by any node. Let Z_n be the set of observations from previous nodes (recorded or received until time k at social group s where $n = s + S(k-1))^{11}$. The benchmark protocol proceeds as follows:

Protocol 2.2 Benchmark Full Information Flow Network Protocol at each node *n*

Step 1. Observation: Node n records its private observation vector z_n according to (2.1).

Step 2. Interaction with other social groups: Node *n* then accesses the network and receives the private observations from other nodes, Z_n .

Step 3. Information aggregation and Bayesian data fusion: Node *n* uses z_n , Z_n to compute it's belief $y_n = p(x|z_n, Z_n)$.

Step 4. Transmit augmented data: Node *n*, then, broadcasts the set of observations $Z_n \cup \{z_n\}$ over the network.

Since Protocol 2.2 serves as an idealized benchmark for designing mis-information removal algorithms, its efficiency is irrelevant. However, it can be made more efficient by requiring each node to only broadcast the observations which have not been already integrated in the belief computed by the other nodes in the network. By comparing the posterior distribution of state of nature in Protocol 2.1 with the same in benchmark Protocol 2.2, mis-information removal algorithm is specified in Section 2.3.

With Protocol 2.2 defined above, the benchmark estimation problem can be summarized as:

Estimate x with prior π_0 subject to:

$$\begin{cases}
G_n = (V_n, E_n) \text{ is given.} \\
z_n \sim B_{iz}, \quad x = \overline{x}_i, \quad \text{(observation process)} \\
y_n = \mathscr{F}(Z_n, z_n), \text{(standard estimation problem)}
\end{cases}$$
(2.6)

The estimates y_n are free of mis-information because node *n* uses all available raw observations (and not estimates) form other previous nodes. Note that estimation problem (2.5) is a dynamic constrained estimation on the directed acyclic graphs. One set of constraints are on the topology of the information flow graph (which are also valid for the estimation problem (2.6) in benchmark protocol). However, there exists another constraint on the algorithm \mathscr{A} (which does not hold for

¹¹In the example (3) in Section 2.1, $Z_{[1,3]} = \{z_{[1,2]}, z_{[2,2]}, z_{[1,1]}, z_{[2,1]}\}$.

the benchmark scenario). As will show in Section 2.3, the algorithm \mathscr{A} has a specific linear form which is depicted by (2.11) in Section 2.3.1.

2.2.3 Modeling Time Evolution of the Information Flow

Before proceeding, we refer the interested reader to Appendix A for a summary of graph theoretic definitions that will be used to model information flow graph in Protocols 2.1 and 2.2. Recall from Section 2.1, G_n denotes the time-dependent information flow graph of the social network. Each node n' in G_n represent a social group s' at time k' such that n' = s' + S(k' - 1), see (2.4). Each directed edge of G_n between node i and node j shows that the information (belief in Protocol 2.1 or observation in Protocol 2.2) of node i is available at node j in the social network represented by G_n . Note that G_n is always a sub-graph of G_{n+1} . Therefore we can use a family of time dependent Directed Acyclic Graphs (DAGs)¹² to model the time evolution of the information flow in the social network. Indeed, the following proposition shows that information flow in a (group-based) social network can always be represented by a family of DAGs.

Theorem 2.2.1. The information flow in a social network defined in Protocol 2.1 and Protocol 2.2 comprising of *S* groups up and until time *k* can be represented by a family of DAGs $\mathscr{G} = {G_n}_{n \in \{1,...,N\}}$ where N = Sk. Each DAG $G_n = (V_n, E_n)$ represents the information flow between the *n* first nodes, where the generic node *n* is defined by (2.4).

Proof. The proof is presented in in Section 2.7.1.

The Adjacency and the Transitive Closure matrices of G_n are denoted by A_n and T_n , respectively (see Appendix A for detail). Because of the fact that the information of each node cannot travel backwards in time, A_n and T_n are upper triangular matrices.

Memory Requirement: In this chapter, we assume that beliefs are valid for duration of *K* timeinstants, where *K* is a positive integer, i.e., social groups at time *k* do not *remember* beliefs generated before time k - K. This means that the size of the adjacency and transitive closure matrices of each graph in \mathscr{G} is limited to N = SK.

2.3 Optimal Mis-information Propagation Removal Algorithm

This section considers the estimation problem (2.5) with constrained information flow Protocol 2.1 in Section 2.2. The aim is to devise the information aggregation algorithm \mathscr{A} in (2.5) such that the estimates of (2.5) with Protocol 2.1 are equal to those of (2.6) with the benchmark Protocol 2.2, i.e., $y_n = \theta_n$. Also, we provide necessary and sufficient conditions on the information flow graph under which the mis-information removal is possible.

¹²see Appendix A

2.3.1 Optimal Combination Scheme in Constrained Information Flow Protocol

Consider the estimation problem (2.5) on directed acyclic graphs where social groups deploy Protocol 2.1. In this section, we address the following question: *How should each social group combine its private observation with the received information (beliefs) from the network so that its updated belief is misinformation free?*

To answer this question, consider estimation problem (2.6) with the idealized benchmark Protocol 2.2, where the set of raw observations are transmitted over the network and, thus, the estimates are mis-information free. In this scenario, since the history of all observations are available at each node and these observations are independent, the standard Bayesian update is used to evaluate

$$y_n = p(x|Z_n, z_n).$$

Estimates y_n are free of mis-information, therefore, to prevent mis-information propagation in constrained information flow Protocol 2.1, the information aggregation algorithm \mathscr{A} should devised such that

$$p(x|\Theta_{u(G_n)}, z_n) = p(x|Z_{v(G_n)}, z_n), \text{ for } n = 1, 2, \dots$$
 (2.7)

So, the first step in building the optimal information aggregation algorithm is to compute the estimates y_n of the benchmark protocol in terms of raw observations and the information flow graph. Before proceeding, let us define

$$\widehat{y}_n^{full} = \log(y_n) = \log(p(x|Z_n, z_n)) \quad \text{for } n = 1, 2, \dots$$
 (2.8)

Since the logarithm is monotonically increasing, surjective function, we can work with logarithm of estimates \hat{y}_n^{full} instead of y_n^{13} . The following proposition gives an expression for the estimates \hat{y}_n^{full} in full information flow network.

Theorem 2.3.1. *Consider estimation problem (2.6) with information exchange Protocol 2.2 of Section 2.2. The mis-information free estimate at node n is:*

$$\widehat{\mathbf{y}}_n^{full} = (t_n \otimes \mathbf{I}_d) \mathbf{l}_{1:n-1} + \mathbf{l}_n, \tag{2.9}$$

where ι_n denotes $\log(p(z_n|x))$ and $\iota_{1:n-1} \triangleq [\iota'_1, \ldots, \iota'_{n-1}]' \in \mathbb{R}^{(n-1)d \times 1}$. Here \otimes denotes Kronecker (tensor) product and \mathbf{I}_d denotes the $d \times d$ identity matrix. Recall that ι_n defined in (A.6) as the first n-1 elements of the n^{th} column of T_n .

Proof. The proof is in Section 2.7.2

¹³Because the logarithm of product is the sum of individual logarithms, it is more convenient to use logarithm of estimates in Bayesian estimation context.

According to Theorem 2.3.1, the optimal mis-information free estimates can be expressed as a linear combinations of $\iota_i = \log(p(z_i|x))$ in terms of Transitive Closure Matrix T_n of graph G_n . Eq. (2.9) is quite intuitive. In information exchange Protocol 2.2, a node broadcasts its own raw observations and also passes the observations received from others nodes so that each node has the entire history of all possible observations, i.e., if there exists a path from node *i* to node *n*, the observation of node *i*, z_i , is available at node *n*. Therefore, the estimate \hat{y}_n^{full} of node *n* is sum of the information from all nodes that are connected to node *n* (by single-hop or multi-hop paths) in information flow graph G_n .

Evaluating the estimates of benchmark Protocol 2.2, we are now ready to tackle the algorithm design problem. Consider the estimation problem (2.5) with information exchange Protocol 2.1 of Section 2.2. The aim is to devise algorithm \mathscr{A} such that (2.7) holds. Define

$$\widehat{y}_n = \log \theta_n \quad \text{and} \quad \widehat{y}_{1:n-1} \triangleq [\widehat{y}'_1, \dots, \widehat{y}'_{n-1}]'.$$
 (2.10)

With the n-1 dimensional vector w_n below denoting a weight vector (a more precise construction is given in Eq. (2.11) below), and t_n defined in (2.9), we propose the following optimal combination scheme:

$$\widehat{y}_n = (w_n \otimes \mathbf{I}_d) \widehat{y}_{1:n-1} + \iota_n, \tag{2.11}$$

Before describing why estimations \hat{y}_n in (2.11) are free of mis-information, we introduce the following constraints on the n-1 dimensional weight vector w_n .

Constraint 2.3.1. Consider the estimation problem (2.5) with information exchange Protocol 2.1. Then the set of weights $\{w_n\}_{n \in \{1,...,N\}}$ in (2.11) satisfies the topological constraint for constrained flow network if $\forall j \in \{1,...,n-1\}$ and $\forall n \in \{1,...,N\}$

$$a_n(j) = 0 \implies w_n(j) = 0,$$
 (2.12)

where a_n is defined in (A.6).

Constraint 2.3.1 imposes a topological condition on the weight vector w_n Assuming that Constraint 2.3.1 holds, Theorem 2.3.2 below asserts that estimates computed from (2.3.1) are identical to the optimal, mis-information free estimates of information exchange Protocol 2.2.

Theorem 2.3.2. Consider estimation problem (2.5) with information exchange Protocol 2.1 of Section 2.2. Then the set of weights $\{w_n\}_{n \in \{1,...,N\}}$ in (2.11) satisfies the topological constraint for constrained flow network if $\forall j \in \{1,...,n-1\}$ and $\forall n \in \{1,...,N\}$. Then the following optimality

property holds for the estimates \hat{y}_n in the constrained information flow network:

$$\widehat{y}_n = \widehat{y}_n^{full} \iff \begin{cases} w_n = t_n \left((T_{n-1})' \right)^{-1} \\ and \ w_n \ satisfy \ Constraint \ 2.3.1, \end{cases}$$
(2.13)

where \hat{y}_n and \hat{y}_n^{full} are defined in (2.10) and (2.8) respectively. Recall that t_n defined in (A.6) as the first n-1 elements of the n^{th} column of T_n .

Proof. The proof is presented in Appendix 2.7.3

In words: A necessary and sufficient condition for $\hat{y}_n = \hat{y}_n^{full}$ to be held is that the weight vector w_n satisfies $w_n = t_n ((T_{n-1})')^{-1}$. As a result of constrained information flow, w_n should simultaneously satisfy topological constraint (2.12) to ensure that required information for mis-information removal is available at node n.

Discussion: According to Theorem 2.3.1, the mis-information free estimates of full information flow protocol at node n is linear in the estimates computed by the previous nodes i.e. $t_{1:n-1}$ and the information collected l_n at node n. This linearity enables us to remove the mis-information in the constrained information flow protocol by employing the optimal combination scheme (2.11) with n-1 dimensional weight vector w_n defined in Theorem 2.3.2. However, according to the communication topology described by Adjacency Matrix A_n , some nodes do not transmit their estimates to node n. Consequently, the estimates of those nodes are not available at node n and must not be used to compute the estimate \hat{y}_n . An obvious way to introduce this constraint in the estimation of \hat{y}_n is to set the weight related to an unavailable estimate to zero. This is Constraint 2.3.1. Therefore, it is also clear that (2.12) is a necessary condition for exact mis-information removal. Now, assuming that all the estimates of the n-1 latest nodes are optimal estimates free of mis-information i.e. $\widehat{y}_{1:n-1} = \widehat{y}_{1:n-1}^{full}$ and are available at node *n*, is it possible to find a vector w_n so that \widehat{y}_n is equal to the optimal estimates free of mis-information i.e. \hat{y}_n^{full} ? Theorem 2.3.2 provides the answer, with $w_n = t_n ((T_{n-1})')^{-1}$. The non-zero elements of w_n show the nodes whose estimates should be available at node n to remove the mis-information. But we know that due to the topology of the graph some of the estimates form previous n-1 nodes are not available at node n. Constraint 2.3.1 basically ensures that the essential estimates (to remove the mis-information) from previous nodes are available at node n and gives a necessary and sufficient condition on the topology of the graph for $\hat{y}_n = \hat{y}_n^{full}$. In Section 2.5.1, we give more intuition on Constraint 2.3.1 by a simple example.

2.4 Sub-optimal Mis-information Removal Algorithm Without Complete Knowledge of Information Flow Graph

So far in this chapter, an optimal information aggregation scheme is proposed for estimation problem (2.5) with Protocol 2.1. Theorem 2.3.2 asserts that the estimates obtained via optimal aggregation scheme (2.11) are equal to those of the benchmark Protocol 2.2. Recall that mis-information propagation arises due to the abusive repetition of information received from other nodes. This happens at one node, for example node n, when there exists a node in information flow graph G_n with two or more links to node n. When, the information flow graph is known, each node can identify the origin of mis-information propagation and, thus, remove that (under the assumptions of Constraint (2.3.1) via optimal information aggregation scheme (2.11). However, this scheme requires the full knowledge of the information flow graph G_n . In this section, we relax that assumption and propose a mis-information removal algorithm for the arbitrary case that each node does not know the path of the received message. Instead, the 'expected adjacency matrix' of the information flow graph is known at all nodes. Our goal here, is to devise a sub-optimal information aggregation scheme based on (2.11) to reduce the effect of mis-information propagation where the information flow graph is unknown. Before presenting the estimation problem when information flow is unknown, let's take a closer look into the expected adjacency matrix of a graph. Recall that the adjacency (or connectivity) matrix of a graph shows the connections among nodes in a graph, i.e., the element on row i and column j of the adjacency matrix is equal to one if there exists a (single-hop) link from node *i* to node *j*, otherwise, it is zero. Similarly, the expected adjacency matrix of a graph is defined as follows:

$$\widetilde{A}_n = [\widetilde{a}_{i,i}] \text{ for } 1 \leq i, j \leq n,$$

where $\tilde{a}_{i,j}$ denotes the probability of having a link from node *i* to node *j*. We assume that instead of adjacency matrix, \tilde{A}_n is known at node *n*. The estimation problem when G_n is unknown can be summarized in the following abstract form:

Estimate x with prior π_0 subject to:

$$\begin{cases} \widetilde{A}_{n} \text{ is given,} \\ z_{n} \sim B_{iz}, \quad x = \overline{x}_{i}, \\ \theta_{n} = p(x | \Theta_{n}, z_{n}, \widetilde{A}_{n}) = \mathscr{B}(\Theta_{n}, z_{n}, \widetilde{A}_{n}), \end{cases}$$
(2.14)

The aim is to devise an algorithm to reduce the effect of mis-information propagation in estimation problem (2.14). Knowing \tilde{A}_n , nodes which are more likely to have multiple paths to node *n* can be identified. Note that in the optimal information aggregation scheme (2.11), the weight vector w_n is the only term which depends on the information flow graph. More specifically, Theorem 2.3.2 computes the optimal weight vector in terms of the transitive closure matrix of the information flow

graph T_n . In estimation problem (2.14), G_n (and consequently T_n) is unknown as opposed to the estimation problem (2.5). Therefore, the main challenge in the scenario where G_n is unknown, is to approximate T_n and, then, compute w_n in terms of the expected transitive closure matrix.

2.4.1 Sub-optimal Combination Scheme

Having known the expected adjacency matrix, our sub-optimal approach to reduce the effect of mis-information at each node, for example node n, consists of three steps:

• First, we approximate the transitive closure matrix (probability of having a path between each pair of nodes in the graph) from the expected adjacency matrix \widetilde{A}_n , that is,

$$T_n(i,j) = p(T_n(i,j) = 1), \quad 1 \le i, j \le n,$$

where $p(\cdot)$ is used to denote probability of an event.

• Second, From \widetilde{T}_n , nodes that are more likely to have a path to node *n* can be identified. From this, the hard estimation of the transitive closure matrix can be constructed, that is,

$$\overline{T}_n(i,j) = \begin{cases} 1 & \overline{T}_n(i,j) > \lambda_{th} \\ 0 & \text{otherwise} \end{cases},$$
(2.15)

where λ_{th} is threshold and \overline{T}_n is the hard estimate of transitive closure matrix of G_n .

• Having computed \overline{T}_n , Algorithm \mathscr{A} , (in the estimation problem (2.5)) is used to reduce effect of the mis-information propagation.

$$\overline{\mathbf{y}}_n = (\overline{w}_n \otimes \mathbf{I}_d) \overline{\mathbf{y}}_{1:n-1} + \iota_n, \qquad (2.16)$$

where weight vector \overline{w}_n satisfies $\overline{w}_n = \overline{t}_n \left(\left(\overline{T}_{n-1} \right)' \right)^{-1}$ and simultaneously satisfies topological Constraint 2.3.1. Here, \overline{T}_n is defined in (2.15) and \overline{t}_n is the first n-1 elements of "*n*"th column of \overline{T}_n and $t_n = \log \left(p(z_n | x) \right)$.

Algorithm 2.3 summarizes the sub-optimal mis-information removal problem without knowledge of information flow network.

2.5 Numerical Examples

In this section, we first provide an example to give more intuition on the topological Constraint 2.3.1 which is required for exact mis-information removal. Then, the performance of the optimal misinformation removal algorithm presented in Theorem 2.3.2 is compared with that of the full infor-

Algorithm 2.3 Algorithm for mis-information removal in step 3 of Protocol. 2.1

For n = 1, 2, ...

- 1. Reconstruct the weighted adjacency matrix of the information flow, \widetilde{A}_n .
- 2. Compute \widetilde{T}_n (2.4.1) and \overline{T}_n using threshold, λ_{th} (2.15).
- 3. Using \overline{t}_n , compute $\overline{w}_n = \overline{t}_n \left(\left(\overline{T}_{n-1} \right)' \right)^{-1}$.
- 4. Update the estimates $\overline{y}_n = (\overline{w}_n \otimes \mathbf{I}_d)\overline{y}_{1:n-1} + \iota_n$.

mation flow communication protocol where instead of the beliefs about the state of nature, all raw observations are transmitted over the network. Finally, the performance of the sub-optimal misinformation removal algorithm (Algorithm 2.3) is investigated in two scenarios: (i) accurate and (ii) inaccurate estimation of the information flow graph.

2.5.1 Example and Intuition on Theorems 2.2.1 and 2.3.2

In this subsection, we provide an example that shows the propagation of mis-information in a simple social network and the mis-information removal algorithm proposed in Section 2.3 to prevent that. Consider a social network consisting of two groups with the following information flow graph until time K = 3



Figure 2.1: Example of constrained information flow network, S = 2 and K = 3. Circles represent a social group at a specific time indexed by (2.4) in the social network and each edge depicts a communication link between two nodes.

There are S = 2 groups and the total time duration K = 3. From (2.4), the element indexed by n = s + 2(k - 1) in Fig.2.1, represents node *s* at time *k*. According to Theorem 2.2.1, we can build the family of N = SK = 6 DAGs, namely, $\{G_1, G_2, G_3, G_4, G_5, G_6\}$. Based on the information flow in Fig.2.1, since nodes 1 and 2 do not communicate (see Fig.2.1), clearly A_1 and A_2 are zero matrices. Nodes 1 and 3 and Nodes 2 and 3 communicate, hence A_3 has two ones; and so on. The adjacency matrix associated with graphs G_1, G_2, G_3, G_4 and G_5 are:

$$A_{1} = \begin{bmatrix} 0 \end{bmatrix}, A_{2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, A_{3} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, A_{4} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A_{5} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The transitive closure matrices T_n are obtained using (A.4). Using (A.4), we derive the transitive closure matrices from the adjacency matrices associated with graph G_1 , G_2 , G_3 , G_4 and G_5 :

$$T_1^{-1} = \begin{bmatrix} 1 \end{bmatrix}, T_2^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, T_3^{-1} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}, T_4^{-1} = \begin{bmatrix} 1 & 0 & -1 & -1 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, T_5^{-1} = \begin{bmatrix} 1 & 0 & -1 & -1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that $T_n(i, j)$ is non-zero only for $i \ge j$ due to the causality—since information sent by a social group can only arrive at another social group at a later time instant. Also note that since G_n is the subgraph of G_{n+1} with node n+1 removed, the adjacency matrix and transitive closure matrix A_{n+1} and T_{n+1} contain A_n and T_n , respectively. Also, they are upper left $n \times n$ matrices, see Remark A.2 in Section A. The weight vectors are derived from the Transitive Closure Matrices via (2.13):

 $w_{2} = \begin{bmatrix} 0 \end{bmatrix},$ $w_{3} = \begin{bmatrix} 1 & 1 \end{bmatrix},$ $w_{4} = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix},$ $w_{5} = \begin{bmatrix} -1 & -1 & 1 & 1 \end{bmatrix}.$

Let us examine these weight vectors. w_2 means that node 2 does not use estimate from node 1. This formula is consistent with the constraint information flow because estimate from node 1 is not available to node 2; see Fig.2.1. w_3 means that node 3 uses estimates from node 1 and 2; w_4 means that node 4 only uses estimates from node 1 and node 2. The estimate from node 3 is not available at node 4. As shown in Fig.2.1, the mis-information propagation occurs at node 5. The vector w_5 says that node 5 adds estimates from nodes 3 and 4 and removes estimates from nodes 1 and 2 to avoid double counting of these estimates already integrated in estimates from node 3 and 4. Indeed, using the algorithm and the weight vector proposed in Theorem 2.3.2, the mis-information propagation is completely prevented in this example. Now consider the case that the edge between node 3 and node 5 does not exist. In this scenario $a_5(2) = 0$ while $w_5(2) \neq 0$, therefore Constraint 2.3.1 does not hold and exact mis-information removal is not possible.

2.5.2 Numerical Examples Illustrating Algorithm *A* in Estimation Problem (2.5)

In this section, numerical results are given to illustrate the effect of mis-information propagation on the performance of multi-agent Bayesian estimation. The excellent efficacy of the optimal misinformation removal algorithm proposed in Section 2.3 is also corroborated in this section. We consider a social network consisting of S = 2 different groups that aim to estimate a scalar state of nature x in the network with a given prior distribution. We simulate a social network with communication delays between different groups chosen randomly from $\{1, 2, ..., 10\}$. The prior distribution of x is uniform distribution U[0, 4] and the observation noise is zero-mean normal distribution N(0, 1). The sample of state of nature is $x^* = 2.78$. The simulation is repeated M = 100 times. At each iteration i, estimated value of state of nature for node n is computed via

$$x_n^i = \sum_{j=1}^N \overline{x}_j p(x = \overline{x}_j | \Theta_n, z_n).$$
(2.17)

Then, the results of all 100 iterations x_n^l are averaged to find the conditional mean of the state of nature x for node n as $\frac{1}{M} \sum_{l=1}^{M} x_n^l$.



Figure 2.2: The conditional mean of the state of nature given the observation in estimation with optimal mis-information removal algorithm compared to the full information network.

Fig.2.2 illustrates the effect of mis-information propagation in the Bayesian estimators. We consider three scenarios:

(i) Full information flow protocol which is free of mis-information as discussed in Section 2.2,

(ii) Constrained information flow protocol with standard Bayesian filter (naive mixing of observations and received information) which may contain mis-information,

(iii) Constrained information flow protocol with optimal mis-information removal algorithm proposed in Section 2.3.

As can be seen in this figure, the performance of the Bayesian estimator is ruined in the existence of the mis-information propagation. The dash-dash line, which represents the standard Bayesian estimation without mis-information removal algorithm, converges to a slightly different value than 2.78. Fig.2.2 also shows the excellent performance of the mis-information removal algorithm presented in Theorem 2.3.2. This figure shows that, the expected value of *x* (unknown state of nature) given estimates obtained by the optimal information aggregation scheme (2.11)–depicted with dashdot line marked with " \Diamond "–is similar to that of the optimal mis-information free estimate of the full information flow protocol depicted by the solid line. This verifies the results of Theorem 2.3.2. We also investigate the performance of the mis-information removal algorithm in terms of the mean squared error in estimation of state of nature, namely $\sigma_n = \frac{1}{M} \sum_{l=1}^{M} (x_n^l - x^*)^2$ where x^* is the true state of nature and x_n^l is the estimates of node *n* at iteration *l* as in (2.17).

We can see in Fig.2.3 that mean squared error associated with the estimates obtained by misinformation removal algorithm proposed in Section 2.3 (dash-dot line marked with " \diamond ") is lower than that of the constrained information flow protocol without mis-information removal proposal (dash-dash line).

2.5.3 Numerical Examples Illustrating Algorithm *B* in Estimation Problem (2.14)

Performance of sub-optimal Algorithm 2.3 is studied in a social network consisting of two groups S = 2 over a duration of K = 25 instants. From (2.4), social group *s* at time *k* is represented by node n = s + 2(k - 1) in information flow graph. Information from one social group reaches the others after a random delay. In our numerical study, the prior of *x* is uniform distribution U[0,4]. Furthermore, the observation noise is zero-mean normal distribution N(0,1). The sample of state of the nature is x = 2.78. In this study, we assume that information flow graph is not known at each node, but a weighted adjacency matrix (which can be considered as a noisy version of true adjacency matrix) is available at each node. This is motivated by the fact that nodes are able to construct the adjacency matrix A_n from the additional information of communication delays. Therefore, each node can construct the weighted adjacency matrix of the information flow graph with elements (i, j) which depicts the probability that the information of a node *i* reaches node *j*. To investigate the performance of the sub-optimal information removal algorithm, we choose the elements of the



Figure 2.3: Comparison of the the mean squared errors of the estimates obtained by optimal misinformation removal algorithm, Bayesian estimator in full information flow network (free of misinformation), and standard Bayesian estimator in constrained information flow network (with misinformation propagation).

weighted adjacency matrix, \widetilde{A}_n as follows:

$$\widetilde{a}_{ij} = \begin{cases} a_{ij} - \beta u_{ij}, & a_{ij} \neq 0 \\ 0, & a_{ij} = 0 \end{cases}$$
(2.18)

where u_{ij} has a uniform distribution U[0,1] and β is a positive real number in (0,1). We choose $\beta = 0.2$ for the "accurate estimation" and $\beta = 0.8$ for the "inaccurate estimation" of the information flow graph. The estimation problem (2.14) is investigated in the following four cases:

(i) Full information flow network with full knowledge of information flow graph at each time (optimal mis-information free scenario) which is shown by the solid line.

(ii) Standard (naive in this context) Bayesian estimation in constrained information flow network with full knowledge of information flow graph at each time which is depicted by the dash-dash line. (iii) Optimal mis-information removal algorithm in constrained information flow network with full knowledge of the information flow graph at each time which is shown via the dotted line marked with " \diamond ".

(iv) Sub-optimal mis-information removal algorithm in constrained information flow network without knowledge of the information flow graph (using the weighted adjacency matrix with $\lambda_{th} = 0.6$) which is shown by the dash-dot line marked with "×".

Fig.2.4 illustrates the expected value of state of nature in the above four scenarios. Similar to



Figure 2.4: Comparison of the conditional mean of the state of nature x given the observations obtained by sub-optimal mis-information removal algorithm, optimal mis-information removal algorithm (knowing the exact information flow graph), Bayesian estimator in full information flow network (free of mis-information), and standard Bayesian estimator in constrained information flow network (with mis-information propagation) in "accurate estimation" scenario ($\beta = 0.2$).

Section 2.5.2, the estimates are found by means of the Monte-Carlo simulations with M = 100 iterations. As can be seen in this figure, the estimations of the state of nature with employing suboptimal mis-information removal algorithm in "good estimation" scenario ($\beta = 0.2$ in (2.18)) is very close to the mis-information free estimates in full information flow network and those obtained by the optimal mis-information removal, knowing the exact adjacency matrix of information flow graph.

Mean squared errors associated with four scenarios studied in this section, are compared in Fig.2.5. As can be seen in the figure, the mean squared error of estimates obtained by the sub-optimal mis-information removal algorithm is lower than the mean squared error of the estimates



Figure 2.5: Comparison of the mean squared errors of the estimates obtained by sub-optimal mis-information removal algorithm, optimal mis-information removal algorithm (knowing the exact information flow graph), Bayesian estimator in full information flow network (free of mis-information), and standard Bayesian estimator in constrained information flow network (with mis-information propagation) in "accurate estimation" scenario ($\beta = 0.2$).

obtained by the standard Bayesian estimation without mis-information removal algorithm. Figures 2.4 and 2.5 show the excellent performance of the proposed sub-optimal algorithm when the information flow graph is not completely known but a good approximation of it is available at each time.

To study the effect of the estimated adjacency matrix of the information flow graph on the performance of the sub-optimal algorithm for mitigating the mis-information propagation, we repeat the simulation for $\beta = 0.8$ in (2.18) (inaccurate estimation scenario). The expected value of the state of nature given the available information in four different scenarios described above are depicted in Fig.2.6.

Fig.2.7 shows the means squared errors of estimates for state of nature in four scenarios under investigation with $\beta = 0.8$ in (2.18). As can be inferred from Figures 2.6 and 2.7, the performance of the sub-optimal depends on the estimated adjacency matrix of information flow graph. When an accurate approximation is available the performance of the sub-optimal algorithm is very close



Figure 2.6: Comparison of the conditional mean of the state of nature x given the observations obtained by sub-optimal mis-information removal algorithm, optimal mis-information removal algorithm (knowing the exact information flow graph), Bayesian estimator in full information flow network (free of mis-information), and standard Bayesian estimator in constrained information flow network (with mis-information propagation) in "inaccurate estimation" scenario ($\beta = 0.8$).

to the performance of the optimal mis-information removal algorithm. However, in the presence of high variance noise in the estimation of the adjacency matrix of the information flow graph, the performance of the sub-optimal algorithm for mitigation mis-information drops.

2.6 Closing Remarks

In this chapter, the problem of mis-information propagation among different groups in social networks is addressed. We considered the most general scenario with arbitrary observation noise and any priori distribution of state of nature. A sufficient and necessary condition for mis-information removal problem is derived based on the topology of the information flow network. Also the performance of the proposed mis-information removal algorithm is illustrated in numerical examples. We also proposed a sub-optimal algorithm to mitigate the mis-information propagation when the information flow network is not known. Numerical results are presented to illustrate the performance of



Figure 2.7: Comparison of the the mean squared errors of the estimates obtained by sub-optimal mis-information removal algorithm, optimal mis-information removal algorithm (knowing the exact information flow graph), Bayesian estimator in full information flow network (free of mis-information), and standard Bayesian estimator in constrained information flow network (with mis-information propagation) "inaccurate estimation" scenario ($\beta = 0.8$).

the proposed algorithms.

Note that data incest considered in this chapter may also arise in any set of sensors that interact over graphs (possibly with random communication delays) and employ Bayesian models for information aggregation. Examples of such sensor network setup includes decentralized target tracking, localization, and fault detection. Although this chapter is motivated by social networks, but the data incest removal algorithm presented in Section 2.3 can applied to remove the mis-information associated with estimates of sensors in such setups.

2.7 **Proof of Theorems**

Here, we present proof for propositions and results of this chapter in the order they appeared.

2.7.1 Proof of Theorem 2.2.1

To prove that the graph $G_n = (V_n, E_n)$ from family \mathscr{G}_n is a directed acyclic graph, we only need to show that the adjacency matrix of G_n is an upper triangular matrix. Then from Lemma A.1, the graph G_n is a directed acyclic graph. Suppose that v_i and v_j are two vertices of G_n , that is $v_i, v_j \in V_n$. From re-indexing scheme (3.7), v_i and v_j represents agents s_i and s_j at time instants k_i and k_j , respectively. We have $v_i = s_i + S(k_i - 1)$ and $v_j = s_j + S(k_j - 1)$. Because of the information flow, information of each agent may become available at other agents at later time instants, a message cannot travel back in the time! This means that if $k_i < k_j$, there should not be an edge from v_j to v_i , $(v_j, v_i) \notin E_n$. Using re-indexing scheme if $k_i < k_j$, then $v_i < v_j$ (because k_i and k_j are integers and $s_i, s_j \leq S$). Therefore, we deduce that

$$i < j \Rightarrow (v_i, v_i) \notin E_n. \tag{2.19}$$

Consequently, the adjacency Matrix is a strictly upper triangular matrix so that G_n is a DAG. Then it follows from the construction of the DAGs that \mathscr{G}_N is a family of DAGs.

2.7.2 Proof of Theorem 2.3.1

In the full information flow protocol, each node uses Bayesian estimation to update the probability distribution of *x* given the set of available observations. The following recursive equation is used at each node to update the estimation of the probability distribution at node n + 1:

$$p(x|Z_{\nu(G_{n+1})}, z_{n+1}) = \pi_0 p(z_{n+1}|x) \prod_{i \in Z_{\nu(G_{n+1})}} p(z_i|x)$$
(2.20)

There is a path from each node $j \in v(G_n)$ to node *n*, therefore from (A.6), $t_n(j) = 1$. Thus, $(t_n \otimes \mathbf{I}_d) \iota_{1:n-1}$ can be written as

$$(t_n \otimes \mathbf{I}_d)\iota_{1:n-1} = \sum_{j \in \nu(G_n)} \iota_j = \sum_{j \in \nu(G_n)} \log \left(p(z_j | x) \right) = \sum_{j \in \nu(G_n)} \log \left(p(z_j | x) \right).$$
(2.21)

Taking logarithms of (2.20), yields:

$$\log(p(x|Z_{\nu(G_n)}, z_n)) = \sum_{j \in \nu(G_n)} \log(p(z_j)|x)).$$
(2.22)

Using (2.22) and (2.21), we can rewrite \hat{y}_n^{full} and complete the proof,

$$\hat{y}_{n}^{full} = \sum_{j \in v(G_{n})} \log \left(p(z_{j}) | x \right) = \sum_{j \in v(G_{n})} \log \left(p(z_{j}) | x \right) + \log \left(p(z_{n}) | x \right) = (t_{n} \otimes \mathbf{I}_{d}) \iota_{1:n-1} + \iota_{n}.$$
(2.23)

40

(Note that we omit $\log \pi_0$ which is the same for both full and constrained information flow protocols for simplicity.)

2.7.3 Proof of Theorem 2.3.2

We, first, show that the left hand side of (2.13) (i.e., $\hat{y}_n = \hat{y}_n^{full}$) implies the right hand side of (2.13). Start with (2.11) for \hat{y}_n and replacing \hat{y}_n with \hat{y}_n^{full} yields

$$\widehat{y}_n^{full} = (w_n \otimes \mathbf{I}_d) \widehat{y}_{1:n-1}^{full} + \iota_n .$$
(2.24)

Using Theorem 2.3.1 it follows that $y_{1:n-1}^{full} = (T'_{n-1} \otimes \mathbf{I}_d)\iota_{1:n-1}$. Then, incorporating this into (2.24) yields

$$\widehat{\mathbf{y}}_{n}^{full} = (w_{n} \otimes \mathbf{I}_{d})(T_{n-1}' \otimes \mathbf{I}_{d})\boldsymbol{\iota}_{1:n-1} + \boldsymbol{\iota}_{n} .$$
(2.25)

From Theorem 2.3.1, we have $\hat{y}_n^{full} = (t_n \otimes \mathbf{I}_d) \iota_{1:n-1} + \iota_n$. Equating the right hand sides of this equation and (2.25) yields

$$(\iota_n \otimes \mathbf{I}_d)\iota_{1:n-1} = (w_n \otimes \mathbf{I}_d)(T'_{n-1} \otimes \mathbf{I}_d)\iota_{1:n-1} = ((w_n T'_{n-1}) \otimes \mathbf{I}_d))\iota_{1:n-1} .$$
(2.26)

The last equality above follows from the distributive property of tensor products. (2.26) is true for any information vector $t_{1:n-1}$. This implies $t_n = w_n T'_{n-1} \implies w_n = t_n (T'_{n-1})^{-1}$, since T_n is an upper triangular matrix with ones on the diagonal¹⁴ and so invertible. To complete the proof that the left hand side of (2.13) implies the right hand side, recall that the information structure for constrained flow is such that if $a_n(j) = 0$ then certain components of the vector $\hat{y}_{1:n-1}$ are not available to node *n*. If the corresponding weight $w_n(j)$ is non-zero it is impossible to reconstruct \hat{y}_n according to (2.11) to be equal to \hat{y}_n^{full} . This is simply the topological constraint (2.12). Showing that the right hand side of (2.13) implies the left hand side is very similar to the above proof and is omitted.

¹⁴See Appendix A.

Mis-information Management Problem in Social Learning Over Directed Acyclic Graphs

3.1 Introduction

Motivated by online rating and review systems, we investigate social learning in a network where agents interact on a time dependent graph to estimate an underlying state of nature. Agents record their own private observations, then update their private beliefs about the state of nature using Bayes' rule. Based on their belief, each agent, then, chooses an action (rating) from a finite set and transmits this action over the social network. An important consequence of such social learning over a network is the ruinous multiple re-use of information known as data incest (or mis-information propagation). In this chapter, the data incest management problem in social learning context is formulated on a directed acyclic graph. We give necessary and sufficient conditions on the graph topology of social interactions to eliminate data incest. A data incest removal algorithm is proposed such that the public belief of social learning (and hence the actions of agents) is not affected by data incest propagation. This results in an online rating and review system with a higher trust rating. Numerical examples are provided to illustrate the performance of the proposed optimal data incest removal algorithm.

In social learning, agents aim to estimate the state of nature using their private observations and actions from other agents [2]. The process of updating belief by agents can be done using Bayesian models [1, 64] or non-Bayesian models [56, 57]. Classical social learning is used to model the behavior of expected cost minimizer agents. Also, social learning can be generalized to of risk averse minimizers. The resulting risk-averse social learning filter is studied in [95].

In this chapter, we consider Bayesian social learning that models expected cost minimizers along with data incest (mis-information propagation). This results in a non-standard information pattern for Bayesian estimation. Before proceeding to the formal definition of data incest in learning over social networks, let us describe the social learning model.

3.1.1 Social Learning Protocol on Network

Consider a social network comprising of *S* agents that aim to estimate (localize) an underlying state of nature (a random variable). Let $x \in {\overline{x}_1, \overline{x}_2, \dots, \overline{x}_X}$ represent a state of nature (such as quality of a hotel) with known prior distribution π_0 where *X* denotes the dimension of the state space. Let $k = 1, 2, 3, \ldots$ depict epochs at which events occur. These events comprise of taking observations, evaluating beliefs and choosing actions as described below. The index *k* depicts the historical order of events and not necessarily absolute time. However, for simplicity, we refer to *k* as "time" in the rest of this chapter. Assume that there exists a network administrator who provides the *network belief* $\pi_{-\lfloor s,k \rfloor}$ defined in Step 5 to node *s* at time *k*. Network belief can be considered as a summary of information received from nodes whose actions are available at node $\lfloor s,k \rfloor$ due to the constraints imposed by the structure of social network. The agents use the following Bayesian social learning protocol to estimate the state of nature:

Step 1. Private observations: To estimate the state of nature *x*, each agent records its *M*-dimensional private observation vector. At each time k = 1, 2, 3, ..., each agent $s \ (1 \le s \le S)$ obtains a noisy private observation $z_{[s,k]}$ from the finite set¹⁵ $\mathbf{Z} = \{\overline{z}_1, \overline{z}_2, ..., \overline{z}_Z\}$ with conditional probability

$$B_{ij} = p(z_{[s,k]} = \overline{z}_j | x = \overline{x}_i).$$
(3.1)

It is assumed that the observations $z_{[s,k]}$ are independent random variables with respect to agent *s* and time k^{16} .

Step 2. Private belief: After obtaining its private observation, each agent combines its private observation with the network belief to evaluate its private belief about state of nature. Each agent *s* combines its private observation $z_{[s,k]}$ with the network belief (which is provided by the network administrator) and evaluates its private belief of state of nature¹⁷. Private belief, $\mu_{[s,k]}$, is evaluated via Bayesian models from the network belief and private observations, that is

$$\mu_{[s,k]} = (\mu_{[s,k]}(i), 1 \le i \le X), \text{ where } \quad \mu_{[s,k]}(i) = p\left(x = \overline{x}_i | \pi_{-[s,k]}, z_{[s,k]}\right). \tag{3.2}$$

Note that private belief of each agent is only available to herself and not to the other agents or the network administrator, (that is why the term "private" is used).

¹⁵The results of this chapter also apply to continuous-valued observations. We consider discrete-valued observations since humans typically record discrete observations.

¹⁶It is not necessary for agents to record observations at each time k and this does not interfere with the common knowledge assumption in social learning where agents all know about the structure of social learning model. Agents at different time instants are treated as different nodes in our graphical model. The assumption that agents record observation at each time k simplifies notation.

¹⁷The scenario where agents choose their actions according to the network belief is similar to the classical social learning formulation [35] where actions are transmitted over the network.

Step 3. Myopic action: Based on its private belief $\mu_{[s,k]}$, agent *s* at time *k* chooses an action $a_{[s,k]}$ from a finite set $\mathbf{A} = \{1, 2, ..., A\}$ to minimize its expected cost function (based on the current information available on the network). That is

$$a_{[s,k]} = \operatorname*{argmin}_{a \in \mathbb{A}} \mathbf{E}\{C(x,a) | \boldsymbol{\mu}_{[s,k]}\}.$$
(3.3)

Here **E** denotes expectation and C(x,a) denotes the cost incurred by the agent if action *a* is chosen when the state of nature is *x*. In the context of rating and review systems, cost function can be considered as the cost of loosing reputation in that review system. For example, if one under-rate a good restaurant, her reputation will be affected negatively and this is costly for her. After agent *s* at time *k* records its action $a_{[s,k]}$, the network administrator automatically computes the updated "*public belief*" at this node by combining action $a_{[s,k]}$ with network belief $\pi_{-[s,k]}$; that is, the public belief $\pi_{[s,k]}$ is

$$\pi_{[s,k]} = (\pi_{[s,k]}(i), 1 \le i \le X), \text{ where } \pi_{[s,k]}(i) = p\left(x = \overline{x}_i | \pi_{-[s,k]}, a_{[s,k]}\right).$$
(3.4)

Step 4. Social network: Unlike private beliefs, public beliefs are visible to the other agents and are broadcasted over the social network¹⁸. These public beliefs are observed by other agents after a random delay (communication delay). We model this information exchange using a family of directed acyclic graphs. Let

$$G_{[s,k]} = (V_{[s,k]}, E_{[s,k]}), \quad k = 1, 2, 3, \dots, s = 1, 2, \dots, S,$$
(3.5)

denote a sequence of time-dependent graphs of information flow in the social network until and including time k. Each vertex in $V_{[s,k]}$ represents an agent s in the social network at time k and each edge ([s',k'],[s'',k'']) in $E_{[s,k]} \subseteq V_{[s,k]} \times V_{[s,k]}$ shows that the public belief (or action) of agent s' at time k' reaches agent s'' at time k''.

Step 5. Network belief: For the past actions (those from other agents at previous time instants), the network administrator has already computed the public beliefs, see Step 3. Define

$$\Theta_{[s,k]} = \{\pi_{[i,j]}; \text{ for all } [i,j] \in V_{[s,k]} \text{ where } ([i,j],[s,k]) \in E_{[s,k]} \}.$$

¹⁸We assume that multiple agents can transmit simultaneously over the network without interfering with each other. This is realistic in a social network, since the time required to exchange (broadcast) information is substantially smaller than the time to record observations, update beliefs or take actions.

At each node, the automated network administrator fuses all the available public beliefs (for example $\Theta_{[s,k]}$) into a single network belief; that is, the network belief is

$$\pi_{-[s,k]} = p(x|\Theta_{[s,k]}) = \mathscr{A}(\Theta_{[s,k]}), \qquad (3.6)$$

where \mathscr{A} denotes the information fusion algorithm used to aggregate the public beliefs received from the network. If algorithm \mathscr{A} is not constructed properly, mis-information propagation (data incest) occurs.

As we will see shortly, a major issue with the above protocol with naive information aggregation in Step 5, is the inadvertent reuse of information (actions of previous agents) which makes the estimates of state of nature biased; that is data incest.

3.1.2 Chapter Goals

The above protocol models the interaction of agents in a social network that aim to estimate the underlying state of nature *x*. An example is where users aim to localize a target event by tweeting the location of the detected "target" on Twitter [132]. Another example is where the state of nature is the true quality of a social unit (such as restaurant). Online rating and review systems such as Yelp or Tripadvisor maintain logs of votes by agents (customers). Each agent visits a restaurant based on reviews on a review website such as Yelp. The agent then obtains private noisy measurement of the state (quality of food in a restaurant). The agent then reviews the restaurant on that review website. Such a review typically is a quantized version (for example, rating) of the total information (private belief) gathered by the agent¹⁹. With such a protocol, how can agents obtain a fair (unbiased) estimate of the underlying state?²⁰. The aim of this chapter is for the network administrator to maintain an unbiased rating and review system, or alternatively modify the actions of agents, to avoid incest.

From a statistical signal processing point of view, estimating the state of nature x using the above five-step protocol is non-standard in two ways: First, agents are influenced by the rating of other agents, this is prior influences their posterior and hence their rating. This effect of agents learning from the actions (ratings) of other agents along with their own private observation is termed "social learning" in the economics literature. Social learning can result in an interesting phenomenon where rational agents can all end up making the same decision (herding and information cascades; [35]). Second, (and this effect is more complex), an agent might be influenced by his own rating leading

¹⁹The dimension of private beliefs is typically larger than that of actions. Also, individuals tend not to provide their private beliefs at the time of their further social interactions. Therefore, agents map their beliefs to a finite set of actions which are easier to broadcast.

²⁰Having fair estimates of quality of a social unit is a problem of much interest in business. Most of hotel managers (81%) regularly check the reviews on Tripadvisor [79]. In [109], it is found that a one-star increase in the average rating of users in Yelp is mapped to about 5-9 % revenue increase.

| Target Localization | Online Rating and Review System |
|---|---|
| Aim: To estimate location of a target x : Geographical coordinates of the target a_n : (action) region of detected target | Aim: To estimate quality of a social unit \mathcal{X} : (State of nature) quality of the social unit \mathcal{A}_n : (action) rating of the social unit |

Examples of Social Learning in Social Networks

Figure 3.1: Two examples of multi-agent social learning in social networks: (i) target localization, and (ii) online rating and review systems.

to data incest.

To explain what can go wrong with the above protocol, suppose an agent wrote a poor rating of the restaurant on a social media site at time 1. Another agent is influenced by this rating and also gives the restaurant a poor rating at time 2. Assume that the information exchange is modeled by the graph depicted in Fig.3.2. The first agent visits the social media site at time 3 and sees that another agent has also given the restaurant a poor rating - this double confirms his rating and he enters another poor rating. In a fair system, the first agent should have been aware that the rating of the second agent was influenced by his rating - so that first agent has effectively double counted his first rating by casting the second poor rating. Data incest is a consequence of the recursive nature of Bayesian social learning and the communication graph. The data incest in a social network is defined as the naive re-use of actions of other agents in the formation of the belief of an agent when these actions could have been initiated by the agent. In Figure 3.2, the fact that there exist two distinct paths between Agent 1 at time 1 and Agent 1 at time 3 (depicted in red) implies that the information of Agent 1 at time 1 is double counted, thereby leading to a data incest event.

The twin effects of social learning and data incest lead to non-standard information patterns in state estimation. Herd occurs when the public belief overrides the private observations and thus actions of agents are independent of their private observations. An extreme case of this is an information cascade when the public belief of social learning hits a fixed point and does not evolve any longer. Each agent in a cascade acts according to the fixed public belief and social learning stops [35]²¹. Data incest results in bias in the public belief as a consequence of the unintentional re-use of identical actions in the formation of public belief of social learning; the information gathered by

²¹There are subtle differences between an individual agent herding, a herd of agents and an information cascade; see for example [35, 91].





Figure 3.2: Example of communication graph, with two agents (S = 2) and over three event epochs (K = 3). The arrows represent exchange of information regarding actions taken by agents.

each agent is mistakenly considered to be independent. This results in over confidence and bias in estimates of state of nature. Due to the lack of information about the topology of the communication graph, data incest arises in Bayesian social learning in social networks. Therefore, the Bayesian social learning protocol requires a careful design to ensure that data incest is mitigated. The aim of this chapter is to modify the five-step protocol presented in Section 3.1.1 such that data incest does not arise. As we will see in Section 3.3.4, the proposed data incest removal algorithm can be applied to the state estimation problems shown in Fig.3.1.

3.1.3 Main Results and Organization of Chapter

With the above five-step social learning protocol in social networks, we are now ready to outline the main results of this chapter:

- 1. In Section 3.2, the data incest problem is formulated on a family of time dependent directed acyclic graphs
- In Section 3.3, a necessary and sufficient condition on the graph is provided for exact data incest removal. This constraint is on the topology of communication delays (communication graph). Also examples where exact incest removal is not possible, are illustrated.
- 3. A data incest removal algorithm is proposed for the five-step social learning protocol in Section 3.3. The data incest removal algorithm is employed by the network administrator to update the network belief in Step 5 of the social learning protocol of Section 3.1.1²².

²²In this chapter we consider Bayesian estimation over a finite time horizon. We do not consider the asymptotic agreement of social learning or consensus formation in social networks. Consensus formation is asymptotic and typically non-Bayesian. From a practical point of view, information exchange in a social network is typically over a finite horizon.

Finally in Section 3.4, numerical examples are provided which illustrate the data incest removal algorithm.

3.2 Social Learning Over Social Networks

The five-step social learning protocol is introduced in Section 3.1.1. We also discussed that as a result of the loopy information exchange graph, data incest (or mis-information propagation) arises because of the abusive re-use of information of the other agents. Here, with the graph theoretic definitions provided in Appendix A, we discuss the diffusion of information in the social network in more details. Before proceeding, for notational simplicity, instead of [s,k], the following scalar index *n* is used:

$$n \triangleq s + S(k-1), \quad s \in \{1, \dots, S\}, \ k \in \{1, 2, 3, \dots\}.$$
 (3.7)

Note that, in the social learning model considered in this chapter, the historical order of events is important and k is used to denote the order of occurrence of events in real time. Subsequently, we will refer to n as a "node" of the time dependent communication graph G_n . Recall from Section 3.1, $G_n = (V_n, E_n)$ denotes the time-dependent communication graph of the social network. Each node n' in G_n represents an agent s' at time k' such that n' = s' + S(k' - 1), see (3.7). Each directed edge of G_n shows a communication link in the social network represented by G_n . This means that if $(n,n') \in E_n$, agent s' at time k' uses the information of agent s at time k to update his private belief about the underlying state of nature x. Note that with the way we defined the communication graph, G_n is always a sub-graph of G_{n+1} . Therefore, as the following theorem proves, diffusion of actions can be modeled via a family of time-dependent Directed Acyclic Graphs (DAGs)²³.

Theorem 3.2.1. The information flow in a social learning over social networks comprising of *S* agents for k = 1, 2, 3, ..., K can be represented by a family of DAGs $\mathscr{G} = \{G_n\}_{n \in \{1,...,N\}}$ where N = SK. Each DAG $G_n = (V_n, E_n)$ represents the information flow between the *n* first nodes, where the generic node *n* is defined by (3.7).

Proof. The proof is similar to that presented in Section 2.7.1.

The adjacency and the transitive closure matrices of G_n are denoted by \overline{A}_n and T_n , respectively (see Appendix A). Using the adjacency and transitive closure matrices of G_n , the following two sets which have involved in formulation of data incest problem in social learning over networks, are introduced:

$$\mathcal{F}_n = \{k, \quad \overline{A}_n(k,n) = 1\},$$

$$\mathcal{H}_n = \{k, \quad T_n(k,n) = 1\}.$$
(3.8)

²³See (A.3) in Appendix A.



Protocol 1: Constrained Social Learning in Social Networks

Figure 3.3: Protocol 1: Constrained social learning in social networks described in Section 3.1.1. As a result of random (unknown) communication delays, data incest arises.

In words, \mathscr{F}_n consists of all nodes who have a single-hop link (edge) to node *n* and \mathscr{H}_n includes the ones with either a single-hop or multi-hop (path) link to node *n*.

3.2.1 Constrained Social Learning in Social Networks

The five-step constrained social learning protocol introduced in Section 3.1.1, is illustrated in Fig.3.3. Note that in the constrained social learning problem, agents do not have information about the communication graph. This is why the term "constrained" is used. The constrained social learning in social networks can be summarized in an abstract form as

Choose action
$$a_n = \underset{a \in \mathbb{A}}{\operatorname{argmin}} \mathbb{E}\{C'_a \mu_n\}$$
 subject to: (3.9)

$$\begin{cases}
z_n \sim B_{iz}, \quad x = i, \quad \text{(observation process)} \\
\Theta_n = \{\pi_i; \quad i \in \mathscr{F}_n\}, \quad \text{(network constraint)} \\
\pi_{-n} = \mathscr{A}(\Theta_n), \quad \text{(network belief formation)} \\
\mu_n = p(x|\pi_{-n}, z_n), \quad \text{(private belief evaluation)}
\end{cases}$$

In (3.9) Algorithm \mathscr{A} is the automated information fusion algorithm employed by the network administrator to compute the network belief. Due to the loopy communication graph and the recursive nature of Bayesian models, data incest (mis-information propagation) arises in constrained social learning if algorithm \mathscr{A} is not designed properly. The aim of this chapter is to devise the algorithm \mathscr{A} such that the public belief of social learning (and consequently, actions a_n for all n = 1, 2, ...) are not affected by data incest.

The following lemma summarizes the social learning filters in (3.9).

Lemma 3.2.1. Consider the five-step social learning protocol presented in Section 3.1.1 with *S* agents and the communication graph G_n . Let π_{-n} denote the network belief of social network at this node. Then, the social learning elements (private belief, action, and public belief) of node *n* with observation vector $z_n = \overline{z_l}$ can be computed from $(1 \le m \le X)$

$$\mu_{n}(m) = p\left(x = \overline{x}_{m} | \Theta_{n}, z_{n}\right) \propto c \pi_{-n}(m) B_{ml},$$

$$a_{n} = \operatorname*{argmin}_{a \in \mathbb{A}} \mathbf{E}\{C(x, a) | \Theta_{n}, z_{n}\} = \operatorname*{argmin}_{a \in \mathbb{A}} \mathbf{E}\{C'_{a} \mu_{n}\},$$

$$\pi_{n}(m) \propto c \pi_{-n}(m) \sum_{j=1}^{Z} \left[\prod_{\widehat{a} \in \mathbf{A} - \{a_{n}\}} \mathbb{I}(C'_{a_{n}} B_{j} \pi_{-n} < C'_{\widehat{a}} B_{j} \pi_{-n})\right] B_{mj},$$
(3.10)

where c is a generic normalizing constant, $B_j = \text{diag}(B_1 j, \dots, B_{X_j})$, and $\mathbb{I}(\cdot)$ is indicator function. Here, C_a is the cost vector defined as $C_a = [C(1,a) \ C(2,a) \ \dots \ C(X,a)]$. Using a matrix notation,

$$\mu_n = \frac{B_{\overline{z}_l} \pi_{-n}}{\mathbf{1}' . B_{\overline{z}_l} \pi_{-n}},$$

where **1** denotes a all-one vector. Also, the public belief π_n can be written as

$$\pi_n = rac{R_{a_n}^{\pi_{-n}}\pi_{-n}}{\mathbf{1}'.R_{a_n}^{\pi_{-n}}\pi_{-n}}.$$

Here, $R_{a_n}^{\pi_{-n}} = \text{diag}(r_1, \dots, r_X)$, where $r_m = \sum_{j=1}^{Z} \left[\prod_{\widehat{a} \in \mathbf{A} - \{a_n\}} \mathbb{I}(C'_{a_n} B_j \pi_{-n} < C'_{\widehat{a}} B_j \pi_{-n}) \right] B_{mj}$. *Proof.* The proof is presented in Section 3.7.1.

Lemma 3.2.1 summarizes the social learning problem considered in this chapter. Each node combines the network belief with its private observation to evaluate its private belief. Based on this private belief, action a_n is chosen such that a local cost function is minimized. Action a_n is used by the network administrator to automatically update the public belief of social learning. Then, the public belief is transmitted over the network. As described in Section 3.1, a major issue with the above protocol is data incest. The aim of this chapter is to devise a data incest removal algorithm for the network administrator to deploy such that the estimates of agents are unbiased.

Remark 3.2.1. In order to choose an action from the finite set of all possible actions, agents minimize a cost function. This cost function can be interpreted in terms of the reputation of agents in online rating and review systems. For example if the quality of a restaurant is good and an agent wrote a bad review for it in Yelp and he continues to do so for other restaurants, his reputation becomes lower among the users of Yelp. Consequently, other people ignore reviews of that (low-reputation) agent in evaluation of their opinion about the social unit under study (restaurant).

Therefore, agents minimize the penalty of writing inaccurate reviews (or equivalently increase their reputations) by choosing proper actions. This behavior is modeled by minimizing a cost function in our social learning model.

Remark 3.2.2. In comparison to the public belief which can be computed by the network administrator (who monitors the agents' actions and communication graph), the agents' private beliefs cannot be computed by the network administrator. The private belief depends on the local observation which is not available to the network. Note that in Step 2 of the constrained social learning Protocol 1, the results of Lemma 3.2.1 are used to compute μ_n using z_n and π_{-n} .

Remark 3.2.3. The constrained social learning protocol is practiced in many online rating and review systems such as Yelp or Tripadvisor²⁴

3.3 Data Incest Removal Algorithm

So far in this chapter, Bayesian social learning model and communication amongst agents in social networks have been described. This section presents the main result of this chapter, namely the solution to the constrained social learning problem (3.9). We propose a data incest removal algorithm such that the public belief of social learning (and consequently the chosen action) is not affected by data incest. To devise the data incest removal algorithm, an idealized framework is presented that prevents data incest as we will describe shortly. Comparing the public belief of the idealized framework with the same of the constrained social learning, the data incest removal algorithm is specified. This data incest removal algorithm is used by the network administrator and replaces Step 5 of the social learning protocol presented in Section 3.1.1. A necessary and sufficient condition for the data incest removal problem is also presented in this section.

3.3.1 The Idealized Benchmark for Data Incest Free Social Learning in Social Networks

In this subsection, an idealized (and therefore impractical) framework that will be used as a benchmark to derive the constrained social learning protocol, is described. In the idealized protocol, it is assumed that the entire history of actions along with the communication graph are known at each node. Due to the knowledge about the entire history of actions and the communication graph (dependencies among actions) in the idealized framework, data incest does not arise²⁵. Define

$$\Theta_n^{\text{full}} = \{a_i; \quad i \in \mathscr{H}_n\},\tag{3.11}$$

 $^{^{24}}$ In Section 3.3, a discussion is presented on the data incest removal algorithms in consumer rating web sites such as Yelp[®] – www.yelp.com.

²⁵In the constrained social learning algorithm, each node receives the most recent public beliefs of its neighbors or equivalently the updated public belief.



Protocol 2: Idealized Social Learning in Social Networks

Figure 3.4: Protocol 2: Idealized benchmark social learning in social networks. In this protocol, the complete history of actions chosen by agents and the communication graph are known. Hence, data incest does not arise. This benchmark protocol will be used to design the data incest removal protocol.

where, \mathcal{H}_n is defined in (3.8). In the idealized framework, the network belief can be written as

$$\pi_{-n}^{\text{full}} = p(x|\Theta_n^{\text{full}}) \propto \pi_0 \prod_{a_i \in \Theta_n^{\text{full}}} p(a_i|x, S_i),$$
(3.12)

where $S_i \subset \Theta_n^{\text{full}}$ denotes the set of actions that a_i depends on them. The public belief in the idealized social learning is free of data incest, as it can be inferred from (4.37). The idealized social learning in social networks (Protocol 2) is illustrated in Fig.3.4. The private belief of node *n* in the idealized social learning is denoted by $\hat{\mu}_n$.

Note that if there exists a path between node *i* and node *n*, then action $a_i \in \Theta_n^{\text{full}}$. Since the history of actions and the dependencies among them (communication topology) are available in the idealized social learning, π_{-n}^{full} is free of data incest.

3.3.2 The Data Incest Free Belief in the Idealized Social Learning Protocol 2

The goal of this chapter is to replace Step 5 of the five-step constrained social learning protocol with an algorithm that mitigates data incest. As described earlier, to solve the data incest management problem, we introduced the idealized social learning that prevents data incest. By comparing the network belief (or equivalently the public beliefs of agents) in the idealized social learning with that in the constrained social learning Protocol 1, the data incest removal algorithm can be invented. Our aim is to devise algorithm \mathscr{A} in (3.9)–also in Step 5 of the five-step social learning protocol in

Section 3.1- such that

$$p(x|\Theta_n^{\text{full}}) = \pi_{-n}.$$
(3.13)

Here, first, an expression is derived for the public beliefs of agents in the idealized social learning Protocol 2. Then, using that, algorithm \mathscr{A} is constructed such that (3.13) holds; that is, data incest is mitigated. Let θ_n^{full} denote the logarithm of public belief of node *n* in the idealized social learning Protocol 2, that is

$$\boldsymbol{\theta}_n^{\text{full}} = \log\left(p(\boldsymbol{x}|\boldsymbol{\Theta}_n^{\text{full}}, \boldsymbol{a}_n)\right). \tag{3.14}$$

Theorem 3.3.1 below gives an expression for θ_n^{full} in the idealized social learning Protocol 2.

Theorem 3.3.1. Consider problem (2.6) with the idealized social learning Protocol 2. The data incest free public belief of node n (which represents agent s at time k according to re-indexing equation (3.7)) is:

$$\theta_n^{\text{full}} = \sum_{i=1}^{n-1} t_n(i) v_i + v_n, \qquad (3.15)$$

where v_k denotes $\log(p(a_k|x, S_k))$. Recall that t_n defined in (A.6) in Appendix A as the first n-1 elements of the n^{th} column of T_n .

Proof. The proof is presented in Section 3.7.2.

As can be seen in (3.15), the (logarithm of the) public belief of node *n* can be written as a linear function in terms of v_i using t_n . Due to this linearity, the data incest removal algorithm can be constructed as we will explain later in this section. Also (3.15) implies that the optimal data incest free public beliefs of agents in the idealized social learning Protocol 2 depend on the communication graph explicitly in terms of the transitive closure matrix²⁶. Basically the non-zero elements of t_n show all nodes who have a path to node *n* and thus their actions contribute in the formation of the private belief of node *n*. Eq. (3.15) is quite intuitive from the fact that each agent employs a recursive Bayesian filter to combine its private observation with the information received from the network.

3.3.3 Data Incest Removal Algorithm for Problem (3.9) With Constrained Social Learning Protocol 1

Given the expression for the public belief of the idealized social learning Protocol 2, the aim here is to propose an optimal information aggregation scheme (that replaces Step 5) such that the public belief of the constrained social learning Protocol 1 is equal to the same of the idealized social

²⁶See (A.3) in Appendix A.

learning Protocol 2 (which is free of data incest). That is, (3.13) holds or equivalently

$$p(x|a_i; \quad i \in \mathscr{H}_n) = p(x|\pi_i; \quad i \in \mathscr{F}_n).$$
(3.16)

Similar to θ_n^{full} , let $\hat{\theta}_n$ denote the logarithm of the after action public belief of node *n*,

$$\widehat{\theta}_n = \log\left(p(x|\Theta_n, a_n)\right). \tag{3.17}$$

We propose the following optimal information aggregation scheme to evaluate the public belief using a n - 1 dimensional weight vector w_n as follows,

$$\widehat{\theta}_n = \sum_{i=1}^{n-1} w_n(i) \widehat{\theta}_i + v_n, \qquad (3.18)$$

where w_n with elements $w_n(i)$ $(1 \le i \le n-1)$ is defined more precisely in (3.20). Using optimal information aggregation scheme (3.18) and (3.10) in Lemma 3.2.1, algorithm \mathscr{A} in (3.9) can be specified.

Remark 3.3.1. The optimal information aggregation scheme (3.18) is deployed by the automated network administrator in Step 5 of the social learning protocol presented in Section 3.1.1 to combine the received information (beliefs or equivalently actions) form other nodes and compute $\sum_{i=1}^{n-1} w_n(i)\hat{\theta}_i$, this is the network belief at node n. Then, node n updates its private belief based on the most updated network belief (provided by the network administrator) and chooses its action a_n accordingly and then transmits it over the network. Then, the network administrator automatically evaluates v_n and updates public belief by computing $\hat{\theta}_n = \sum_{i=1}^{n-1} w_n(i) \hat{\theta}_i + v_n$.

The weight vector w_n depends on the communication graph and can be computed simply by (3.20). Theorem 3.3.2 below asserts that by using the optimal information aggregation scheme (3.18) with w_n defined in (3.20), data incest can be completely mitigated. However, for some network topologies, it is not possible to remove data incest completely. The following constraint presents the necessary and sufficient condition on the network for the exact data incest removal.

Constraint 3.3.1. Consider the constrained social learning problem (3.9) with Protocol 1. Then, the weight vector w_n used in optimal information aggregation scheme (3.18) satisfies the topological constraints if $\forall j \in \{1, ..., n-1\}$ and $\forall n \in \{1, ..., N\}$

$$b_n(j) = 0 \implies w_n(j) = 0, \tag{3.19}$$

where b_n is defined in (A.6) and denotes the n-th column of the adjacency matrix of G_n . Basically

Constraint 3.3.1 puts the "availability constraint" on the communication graph.

This means that if information of node j is required at node n ($w_n(j) \neq 0$), there should be a communication link between node j and node n ($b_n(j) \neq 0$). Assuming that Constraint 1 holds, Theorem 3.3.2 below ensures that the public belief of nodes in problem (3.9) with the constrained social learning Protocol 1 is identical to the same of the problem (2.6) with the idealized social learning Protocol 2.

Theorem 3.3.2. Consider problem (3.9) with the constrained social learning Protocol 1 of Section 3.2. Then using the optimal information aggregation scheme (3.18), data incest can be mitigated by using the optimal set of weights $\{w_n\}_{n \in \{1,...,N\}}$ given that the topological Constraint 3.3.1 is satisfied. The optimal weight vector is

$$w_n = t_n \left((T_{n-1})' \right)^{-1}. \tag{3.20}$$

By using the optimal combination scheme (3.18) and optimal weight vector defined in (3.20), the data incest in social learning problem (3.9) is completely mitigated, that is $\hat{\theta}_n = \theta_n^{\text{full}}$ if w_n satisfies topological Constraint 3.3.1 where $\hat{\theta}_n$ and θ_n^{full} are defined in (3.17) and (3.14) respectively. Recall that t_n is defined in (A.6) as the first n-1 elements of the n^{th} column of T_n .

Proof. The proof is presented in Section 3.7.3.

Theorem 3.3.2 proves that if $w_n = t_n ((T_{n-1})')^{-1}$ then $\hat{\theta}_n = \theta_n^{\text{full}}$, that is, data incest (misinformation propagation mitigated).

Using the optimal information aggregation scheme (3.18), the five-step Bayesian social learning protocol in Section 3.1.1 with data incest removal algorithm can be summarized as

 \Box

Algorithm 3.4 Constrained Bayesian social learning with data incest removal algorithm at each node n

Step 1. Observation process: Private observation vector z_n is obtained according to (3.1).

Step 2. Private belief: Node n accesses the network and evaluates its private belief according to (3.2) using the most recent network belief at node n,

$$\mu_n = \frac{B_{\overline{z}_l} \pi_{-n}}{\mathbf{1}' . B_{\overline{z}_l} \pi_{-n}}.$$

Step 3. Myopic action: Action a_n is chosen via (3.3), that is

$$a_n = \operatorname*{argmin}_{a \in \mathbb{A}} \mathbf{E}\{C'_a \mu_n\}$$

Then, the automated network administrator compute the public belief at node n which is defined in (3.4), that is

$$\pi_n = rac{R_{a_n}^{\pi_{-n}}\pi_{-n}}{\mathbf{1}'.R_{a_n}^{\pi_{-n}}\pi_{-n}}.$$

Step 4. Social network: Social network model is similar to the same in Step 4 of the protocol presented in Section 3.1.1.

Step 5. Network belief update: The automated network administrator evaluates the network belief using $\Theta_n = \{\pi_i, i \in \mathcal{H}_n\}$ and the optimal weight vector

$$w_n = t_n \left(\left(T_{n-1} \right)' \right)^{-1}.$$

Discussion of topological constraint (3.19): The non-zero elements of w_n are corresponding to the nodes whose information are required at node *n* to remove data incest. This imposes a topological constraint on the communication graph. If $w_n(j)$ is non-zero, this means that information of node *j* is needed at node *n* and there should be an edge in G_n that connects node *j* to node *n*, this is the topological Constraint 3.3.1. Constraint 3.3.1 ensures that the essential elements for data incest removal are available at node *n* and Theorem 3.3.2 specifies the exact data incest removal algorithm. From Theorem 3.3.2, it is simple to show that Constraint 3.3.1 is a necessary and sufficient condition for data incest removal in learning problem (3.9). Consider two examples of communication graph shown in Figure 3.5.

The optimal weight vector at node 5 for both networks of Fig.3.5 computed from (3.20) is $w_5 = [-1, -1, 1, 1]$. This means that there should be a link between node 2 and node 5 for exact data incest removal according to the topological constraint (3.19). Hence, Constraint 3.3.1 does not hold for the network of Figure 3.5b, while the topological constraint is satisfied in network depicted in Fig.3.5a. Also as it is clear from the network shown in Fig.3.5a, there is no need for the communication graph to be a tree.



Figure 3.5: Two examples of networks: (a) satisfies the topological constraint, and (b) does not satisfy the topological constraint.

3.3.4 Discussion of Data Incest Removal in Social Learning

Here, we discuss the application of data incest removal Algorithm 3.4 (presented in Theorem 3.3.2) in two examples of multi-agent state estimation problem which are presented in the introductory part of this chapter (i) online rating and review systems, and (ii) target localization using social networks, see Fig.3.1. Both problems can be formulated using the five-step constrained social learning protocol presented in Section 3.1.1. As illustrated in Fig.3.6, agents observe the underlying state of nature in noise and practice social learning to choose an action such that a local cost function is minimized. But as a result of unknown communication graph and the recursive nature of Bayesian estimators, data incest or abusive re-use of information occurs. To mitigate data incest, the network administrator plays an intermediating role. Instead of transmitting the communication graph and complete history of actions, the network administrator monitors all the information exchanges and provides the data incest free network belief of social learning at each node. To compute the data incest free public belief, the network administrator uses the optimal information aggregation scheme (3.18) with the optimal weight vector w_n , see (3.20). Using the most updated public belief and its own private observation z_n , node *n* evaluates its private belief. Based on this private belief (which is free of data incest), action a_n is chosen and transmitted it over the network. Given that the communication graph satisfies the topological Constraint 1, Theorem 3.3.2 ensures that by means


Figure 3.6: Data incest removal algorithm employed by network administrator in the state estimation problem over social network. The underlying state of nature could be geographical coordinates of an event (target localization problem) or reputation of a social unit (online rating and review systems).

of the optimal weight vector w_n , action a_n is not affected by data incest and, therefore, performance of state estimation process is improved.

3.4 Numerical Examples

In this section, numerical examples are given to illustrate the performance of data incest removal Algorithm 3.4 presented in Section 3.3. As described in the five-step protocol of Section 3.1, agents interact on a graph to estimate an underlying state of nature (which represents the location of a target event in target localization problem, or the reputation of a social unit in online rating and review systems). The underlying state of nature *x* is a random variable uniformly chosen from $\mathbf{X} = \{1, 2, \dots, 20\}$, and actions are chosen from $\mathbf{A} = \{1, 2, \dots, 10\}$. We consider the following three scenarios for each of four different types of social networks:

- 1. Constrained social learning without data incest removal algorithm (data incest occurs) depicted with dash-dot line
- 2. Constrained social learning with Protocol 1 with data incest removal algorithm depicted with dashed line
- Idealized framework where each node has the entire history of raw observations and thus data incest cannot propagate. This scenario is only simulated for comparison purposes and is depicted by solid line.

The effect of data incest on estimation problem and the performance of the data incest removal algorithm, proposed in Section 3.3, is investigated for the networks shown in Fig.3.7.

We first consider a communication graph with 41 nodes. The communication graph under study, which is shown in Fig.3.7a, satisfies the topological constraint (3.19). The action of node 1 reaches all other nodes and node 41 receives all actions of previous 40 nodes (some edges are omitted from the figure to make it more clear).



Figure 3.7: Three different communication topologies: (a) the communication graph with 41 nodes, (b) agents interact on a fully interconnected graph and the information from one agent reach other agents after a delay chosen randomly from $\{1,2\}$ with the same probabilities, (c) star-shaped communication topology with random delay chosen from $\{1,2\}$.

As can be seen in Fig.3.8, data incest makes agents' actions in the constrained social learning without data incest removal different from the same in the idealized framework. Also Fig.3.8 corroborates the excellent performance of data incest removal Algorithm 3.4. As illustrated in Fig.3.8, the actions of agents in social learning with data incest removal algorithm are exactly similar to those of the idealized framework without data incest. The social learning problem over the graph shown in Fig.3.7a is simulated 100 times to investigate the difference between the estimated state of nature with the true one (x = 10). The estimates of state of nature (obtained in three different scenarios discussed in the beginning of the section) are depicted in Fig.3.9. As can be seen from the figure, the estimates obtained with data incest removal algorithm are very close to data incest

free estimates of Scenario (iii). The bias in estimates in presence of data incest is also clear in this figure.

In the next simulation, a different communication topology is considered. We repeat the simulation for a star-shaped communication graph comprising of six agents (S = 6) at four time instants, K = 4, so the total number of nodes in the communication graph is 24, see Fig.3.7c. The communication delay is randomly chosen from $\{1,2\}$ with the same probabilities. We simulated the social learning in three different scenarios discussed above, to investigate the effect of data incest on the actions and the estimates of agents in the star-shaped social network. The actions chosen by nodes are depicted in Fig.3.10. As can be seen from Fig.3.10, using the data incest removal algorithm, the agents' actions in the constrained social learning with Protocol 1 are very close to those of the idealized social learning with Protocol 2 which are free of data incest. Also the estimates of state of nature are very close to the true value of state of nature compared to the constrained social learning without data incest removal algorithm. Also note that the effect of data incest, as expected, in this communication topology is different for each agent; the agent who communicates with all other nodes is affected more by data incest. This fact is verified in Figures 3.10 and 3.11.

In the third example, a complete fully interconnected graph (where agents communicate with all other agents) is considered. In this example, action of each agent becomes available at all other agents after a random delay chosen from $\{1,2\}$ with the same probabilities. The agents' actions are shown in Fig.3.12. Similar to the star-shaped graph, using data incest removal Algorithm 3.4 makes the agents' actions in the constrained social learning very similar to those of the idealized (data incest free) framework. Also, the excellent performance of data incest removal Algorithm 3.4 in the estimation problem is depicted in Fig.3.13.

We also extend our numerical studies to an arbitrary random network with five agents, S = 5, K = 4. We consider a fully connected network and assume that the interaction between two arbitrary agents (say agent *i* and agent *j*) at time *k* has four (equiprobable) possible statuses: (i) connected with delay 1, (ii) connected with delay 2, (iii) connected with delay 3, and (iv) not connected. If the link is connected with delay τ , this means that the information from agent (i) at time *k* becomes available at agent *j* at time $k + \tau$. If the link is not connected, the information of agent *i* at time *k* never reaches agent *j*. We verify that the underlying communication graph, G_n , satisfies the topological Constraint 3.3.1. Fig.3.14 depicts the agents' actions in three different scenarios (with data incest, without data incest, and with data incest removal algorithm). The simulation results show that, even in this case with arbitrary network (that satisfies topological constraint), the actions obtained by the constrained social learning with data incest removal algorithm is very close to those in the idealized social learning. As expected, using the data incest removal algorithm, the data incest associated with the estimates of agents can be mitigated completely, as shown in Fig.3.15.

Here, we also present numerical studies to investigate the accuracy of the state estimation considered in this chapter in terms of mean squared error. The mean squared error of estimates obtained



Figure 3.8: Actions of agents obtained with social learning over social networks in three different scenarios described in Section 3.4 with communication graph depicted in Fig.3.7a.



Figure 3.9: Mean of the estimated state of nature in the state estimation problem with social learning over social networks in three different scenarios described in Section 3.4 with communication graph depicted in Fig.3.7a.



Figure 3.10: Actions of agents obtained with social learning over social networks in three different scenarios described in Section 3.4 with communication graph depicted in Fig.3.7c.



Figure 3.11: Mean of the estimated state of nature in the state estimation with social learning over social networks in three different scenarios described in Section 3.4 with communication graph depicted in Fig.3.7c.



Figure 3.12: Actions of agents obtained with social learning over social networks in three different scenarios described in Section 3.4 with communication graph depicted in Fig.3.7b.



Figure 3.13: Mean of the estimated state of nature in the state estimation problem with social learning over social networks in three different scenarios described in Section 3.4 with communication graph depicted in Fig.3.7b.



Figure 3.14: Actions of agents obtained with social learning over social networks in three different scenarios described in Section 3.4 with arbitrary communication graph.



Figure 3.15: Mean of the estimated state of nature in the state estimation problem with social learning over social networks in three different scenarios described in Section 3.4 with arbitrary communication graph.

in social learning with three different scenarios discussed in the beginning of this section (with data incest, with data incest removal algorithm, and the idealized framework) is computed for each of four scenarios considered in our numerical studies. Fig. 3.16 depicts the mean squared error of estimates obtained in the first example with social learning over the communication graph of Fig. 3.7a. As can be seen from this figure, the mean squared error associated with the estimates of the constrained social learning with data incest removal Algorithm 3.4 are lower than the those of the constrained social learning in presence of data incest. This means that the performance of estimation problem with social learning is improved using data incest removal algorithm proposed in this chapter.

Figures 3.17 and 3.18 show the mean squared error of estimation with communication graphs presented in Figures 3.7b and 3.7c, respectively, and Figure 3.19 depicts the same for the arbitrary random network with five agents and random communication delays described earlier in this section. As can be seen in these figure, as a result of herding, in star shaped and random communication topologies, the mean squared error of estimates is slightly (compared to the scenario without data incest removal algorithm) more than the idealized framework at each time.

3.5 Psychology Experiment

This section presents an experimental study to investigate the learning and decision making behavior of individuals in a human society. Social learning is used as the mathematical basis for modeling interaction of individuals that aim to perform a perceptual task interactively. A psychology experiment was conducted on a group of undergraduate students at the University of British Columbia to examine whether the decision (action) of one individual affects the decision of the subsequent individuals. The major experimental observation that stands out here is that the participants of the experiment (agents) were affected by decisions of their partners in a relatively large fraction (60%) of trials. We fit a social learning model that mimics the interactions between participants of the psychology experiment. Mis-information propagation (also known as data incest) within the society under study is further investigated in this experiment.

3.5.1 Experiment Setup

Here, a detailed description of the psychology experiment we carried out to study the learning behavior of individuals in a human society is presented:

- *Experiment Date:* The psychology experiment was conducted in September and October 2013.
- *Society under study:* The participants were 36 undergraduate students of the department of Psychology of the University of British Columbia who completed the experiment for course



Figure 3.16: Mean squared error of estimates (of state of nature) obtained with social learning with communication graph depicted in Fig.3.7a.



Figure 3.17: Mean squared error of estimates (of state of nature) obtained with social learning with communication graph depicted in Fig.3.7b (complete fully interconnected graph).



Figure 3.18: Mean squared error of estimates (of state of nature) obtained with social learning with communication graph depicted in Fig.3.7c (star-shaped communication graph).



Figure 3.19: Mean squared error of estimates (of state of nature) obtained with social learning with arbitrary communication graph.

credit.

• Experiment Setup: Participants were asked to perform a perceptual task interactively. Two arrays of circles were given to each pair of participants, then, they were asked to judge which array had the larger average diameter; that is, picking their actions. On each trial, two 4×4 grids of circles were generated by randomly drawing from the radii: $\{20, 24, 29, 35, 42\}$ (in pixels). The average diameter of each grid was computed, and if the means differed by more than 8% or less than 4%, new grids were made, i.e., each trial had arrays of circles differing in the average diameter length by 4-8%²⁷. One participant was chosen randomly and started the experiment by choosing an action according to his observation. Thereafter, each member saw their partner's previous response (action) and his own previous action prior to making their own judgment; this is social learning. The participants continued choosing actions until their responses stabilized for a run of at least three (two participants did not necessarily agree, but each was fixed in her responses). In this experimental study, each participant chose an action in $\mathbb{A} = \{0, 1\}$; a = 0 when she judged that the left array of circles had the larger diameter and a = 1 when her judgments was that the right array of circles had the larger diameter. In each experiment, judgments (actions) of participants are recorded along with the amount of time taken to make that judgment. Fig. 3.24 shows the judgments of two participants within a group at different trials in one experiment. In this experiment, the average diameter of the left array of circles was 32.1875 and the right array was 30.5625 (in pixels).

3.5.2 Experimental Results

The results of our experimental study, which are summarized in Fig.3.23, are as follows:

• Social learning Model: As mentioned above, the experiment for each pair of participants was continued until both participants' responses stabilized. A question that may arise here is: *In what percentage of these experiments, an agreement is made between two participants*? The answer to this question unveils that whether in our experiments "herding" occurred or not. In other words, it reveals that if participants exercised a social learning (influenced by their partners) or not. Interestingly, our experimental study shows that in 66% of total experiments (1102 among 1658), participants reached an agreement; that is herding occurred. Further, our experimental studies show that in 32% of experiments, the social learning was successful and both participants made the right judgment after few interactions. To find a proper social learning model, we focus on the experiments where both participants reached an agreement. Define the social learning (SL) success rate as

SL Success Rate: $\frac{\text{No. of experiments where both participants chose the correct side}}{\text{No. of experiments where both participants reached an agreement}}$

²⁷These numbers are based on the work in Treisman et. al. paper [142].



Figure 3.20: Two arrays of circles were given to each pair of participants on a screen. Their task is to interactively determine which side (either left or right) had the larger average diameter. The partner's previous decision was displayed on screen prior to the stimulus.

In this experimental study, the state of nature belongs to $x \in \{0, 1\}$ where x = 0, when the left array of circles has the larger diameter and x = 1, when the right array has the larger diameter. The initial belief for both participants is considered to be $\pi_0 = [0.5, 0.5]$. The observation state is assumed to be $z \in \{0, 1\}$. We fit a social learning model to our experimental data which gives the same success rate as the experimental study. The social learning parameters (probability observations, $B_{iz} = p(z_k = z | x = i), i \in \{0, 1\}, z \in \{0, 1\}$ and the cost function $C(i, a), i \in \{0, 1\}, a \in \mathbf{A}$), obtained by exhaustive search, are as follows:

$$B_{iy} = \begin{bmatrix} 0.61 & 0.39 \\ 0.41 & 0.59 \end{bmatrix}$$
$$C(i,a) = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}.$$

• *Data incest:* Here, we study the effect of data incest on the judgments of participants in our experimental study. Since we do not have access to the private observations of individuals (almost no one has such information!), we cannot exactly verify that whether data incest changed the judgment of an individual in each trial of the experiment. However, two scenarios which are depicted in Fig.3.22 are used to find data incest events in the experiments. In these



Figure 3.21: Actions of two participants in a group at different trials in one experiment.

two events, as can be seen in Fig.3.22, the action of the first participant at time k influences the action of the second participant at time k + 1, and thus, is double counted by the first participant at time k + 2. As discussed above, since we do not have access to the private observations of participant, we cannot exactly say that data incest affects the action of the first participant at time k + 2 or not. However, it is clear that in these events, data incest occurs. As we expect from the communication topology between partners, data incest occurred in relatively large percentage of trials in the experiment. More precisely, in 79% of experiments one of the data incest events shown in Fig.3.22 occurred (1303 experiments with data incest among 1658 experiments). Our experimental study further shows that in 21% of experiments, data incest resulted in changing the decision of one of the participants in the group, i.e., the judgment of participant at time k + 1 differed from her judgments at time k + 2 and k in the events shown in Fig.3.22. This experimental study reveals that data incest is quite common in social learning in human societies (happened frequently even in our simple social learning setup) and, therefore, social learning protocols require a careful design to handle and mitigate data incest.

• *Discussion:* Among 3316 (non-unique) participants of this experiment, 1336 participants (around 40%) did not change their judgments after observing the action of their partners, while the other 60% changed their initial judgment and were influenced by the action of their partners. An experimental observation that stands out here is that the individuals can be divided into two types: (i) *boundary agents* who stand firm on their decisions during the trial,



Figure 3.22: Two scenarios where data incest arose in our experimental studies.

i.e., their decisions are independent of decisions of the other agents and (ii) *internal agents* who are affected by decisions of the other agents. Fig. 3.24 shows the sample path of two participants in a group, Participant 1 is an internal node while Participant 2 is a boundary node.

To study the decision making behavior of individuals of each type, we investigate the time taken by each participant to make his judgment. Let $\mu_{judg.}$ and $\sigma_{judg.}$ denote the mean and the standard deviation of the time taken by participants to make their judgments in milliseconds. The results of our experimental study, which are presented in Table I, show that the internal nodes, in average, required more time to make their judgments compared to the boundary nodes; this is quite intuitive from the fact that the boundary nodes stood firm on their decisions and ignored the judgment of their partners and thus required less time to make their judgments.

| Type of nodes | relative frequency | $\mu_{ m judg.}$ | $\sigma_{ m judg.}$ |
|---------------|--------------------|------------------|---------------------|
| Internal | 40 % | 1058 ms | 315 ms |
| Boundary | 60 % | 861 ms | 403 ms |

Table 3.1: The frequency of the internal and the boundary nodes in a community of 3316 undergraduate students of the University of British Columbia along with the statistics of the time required by participants (of both types) to make their judgments in milliseconds.

3.6 Closing Remarks

In this chapter, the state estimation problem in social networks with social learning is investigated. State of nature could be geographical coordinates of an event (target localization problem) or quality



Figure 3.23: Social learning with data incest that is exercised by groups of students who were asked to perform a conceptual task in our experimental study.

of a social unit (online rating and review system). In online rating and review systems, privacy concerns impose a constraint on the resolution of information that users reveal to other people. People are more likely to share a lower resolution action to others rather than their detailed private observations²⁸.

As discussed in the chapter, data incest arises in this setup as a result of the recursive nature of Bayesian estimation and random communication delays in social networks. We proposed a data incest removal algorithm for the multi-agent social learning in social networks in this chapter along with a topological necessary and sufficient condition for data incest free estimation. The main difference of this work with the data incest removal algorithms in Chapter 2 is that in this chapter we considered a social learning context where only public belief of agents (which can be computed directly from actions) is transmitted over the network while in the previous chapter, the private belief of agents which depends on their private observations are transmitted through the network.

The results of this chapter can be applied to a scenario where the network administrator provides the public beliefs to agents (instead of the updated network belief). In this scenario, agents combine the received public beliefs using the optimal weight vector w_n to compute the updated public belief

²⁸One of the issues that comes with proliferation of online social networks (especially content-aware recommender systems), due to the astronomical amounts of information that these sites have about their users, is privacy. People are not usually willing to disclose their private information to a large group of audience. That's the reason why finding a trade-off between accuracy and privacy of the users have been studied widely in the literature of recommender systems [47, 126, 129, 151].



Figure 3.24: Actions of two participants in a group at different epochs. Participant 1 can be considered as an internal node and Participant 2 can be viewed as a boundary node.

and then evaluate their private belief accordingly. Optimal weights (which depends on the topology of the communication graph) and set of available public beliefs are the essential ingredients that one needs in order to compute the network belief. As can be seen in Figure 3.25, these ingredients can be computed by separate units (that do not communicate with each other) and the user can aggregate these two and compute the most updated network belief. This mean that only he has access to the updated network belief and, thus, his privacy is preserved²⁹.

3.7 **Proof of Results**

Here, we present proof for propositions and results of this chapter in the order of their appearance.

3.7.1 Proof of Lemma 3.2.1

Proof. We assume that each node has the most up-to-date public belief of social learning, $\pi_{-n} = p(x|\Theta_n)$. This node records its own private observation $z_n = \overline{z}_l$. The private belief is

$$\mu_n(m) = p(x = \overline{x}_m | \Theta_n, z_n). \tag{3.21}$$

²⁹Another type of privacy concerns that may arise here is the fact that the network administrator can compute the network belief for users, and thus, can predict the actions that users are about to make.



(Node n: User of Online Rating and Review System)

Figure 3.25: Optimal weights (which depends on the topology of the communication graph) and set of available public belief are computed in separate units. The user, then, can compute the most updated data-incest free network belief.

Using Bayes' theorem, (3.22) can be written as

$$\mu_n(m) = p(x = \overline{x}_m | \Theta_n, z_n) = c p(z_n | x) p(x | \Theta_n)$$

= $c \pi_{-n}(m) B_{ml}.$ (3.22)

The normalizing factor *c* is used to make μ_n a true probability mass function, that is $\sum_{m=1}^{X} \mu_n(m) = 1$. Expected cost given μ_n is equal to $C'_a \mu_n$ thus the action a_n is $a_n = \operatorname{argmin}_{a \in \mathbb{A}} \{C'_a \mu_n\}$. To complete the proof, we need to compute the public belief, $\pi_{+n} = p(x|\Theta_n, a_n)$. Applying Bayes' theorem, the after action public belief can be written as

$$\pi_{+n}(m) = p(x = \overline{x}_m | \Theta_n, a_n) = c p(a_n | \Theta_n, x) p(x = \overline{x}_m | \Theta_n)$$
$$= c p(a_n | x, \pi_{-n}) \pi_{-n}(m) = c \sum_{j=1}^Z p(a_n | x, z = \overline{z}_j, \pi_{-n}) p(z = \overline{z}_j).$$
(3.23)

Knowing observations and public belief, the private belief can be computed. From the private belief, the action a_n is speified. Thus

$$p(a_n|x, z = \overline{z}_j, \pi_{-n}) = \begin{cases} 1 & \text{if } a_n = \operatorname{argmin}_{a \in \mathbb{A}} \{ C'_a B_j \pi_{-n} \} \\ 0 & \text{if } a_n \neq \operatorname{argmin}_{a \in \mathbb{A}} \{ C'_a B_j \pi_{-n} \} \end{cases}$$
(3.24)

where $B_j = \text{diag}(B_1 j, \dots, B_{X_j})$. Using indicator function $\mathbb{I}(\cdot)$, Eq. (3.24) can be reorganized as

$$p(a_n|x,z=\overline{z}_j,\pi_{-n}) = \prod_{\widehat{a}\in\mathbf{A}-\{a_n\}} \mathbb{I}(C'_{a_n}B_j\pi_{-n} < C'_{\widehat{a}}B_j\pi_{-n})$$
(3.25)

Substituting (3.25) in (3.23) completes the proof as follows

$$\pi_{+n}(m) = c\pi_{-n}(m) \sum_{j=1}^{Z} \left[\prod_{\widehat{a} \in \mathbf{A} - \{a_n\}} \mathbb{I}(C'_{a_n} B_j \pi_{-n} < C'_{\widehat{a}} B_j \pi_{-n}) \right] B_{mj}$$
(3.26)

3.7.2 Proof of Theorem 3.3.1

Proof. The logarithm of the public belief of learning problem (2.6) with benchmark information exchange Protocol 2, θ_n^{full} , is $\log(p(x|\Theta_n^{\text{full}}, G_n))$. Recall that Θ_n^{full} denotes the entire history of actions from previous nodes who have a path to node *n* and *S_i* denotes the set of all actions that *a_i* depends on them. Also from definition of the transitive closure matrix (A.3) and *t_n* in (A.6), the nodes who have a path to node *n* are corresponding to non-zero elements of *t_n*. Because if *t_n(i) = 1*, then there exists a path from node *i* to node *n*. Therefore, the public belief can be written as

$$p(x|\Theta_n^{\text{full}}, a_n, G_n) = cp(a_n|S_n, x)p(x|\{a_i; a_i \in \Theta_n^{\text{full}}\})$$
$$= c\pi_0 p(a_n|S_n, x) \prod_{a_i \in \Theta_n^{\text{full}}} p(a_i|S_i, x).$$
(3.27)

Note that Bayes' theorem is used recursively to expand $p(x|\{a_i;a_i \in \Theta_n^{\text{full}}\})$ and S_i includes actions(from Θ_n^{full}) into account that a_i depends on them. Taking the logarithm of both sides of (3.27) yields

$$\theta_n^{\text{full}} = \log \left(p(x|\Theta_n^{\text{full}}, a_n, G_n) \right)$$

= $\log \left(c \pi_0 p(a_n | S_n, x) \prod_{a_i \in \Theta_n^{\text{full}}} p(a_i | S_i, x) \right)$
= $\log \left(p(a_n | S_n, x) \right) + \sum_{t_n(i) \neq 0} \log \left(p(a_i | S_i, x) \right),$
= $\sum_{i=1}^{n-1} t_n(i) \mathbf{v}_i + \mathbf{v}_n,$ (3.28)

where v_i denotes $\log(p(a_i|x, S_i))$. Note that the normalizing constant c and π_0 are omitted for the sake of simplicity as they are the same for both learning problems (3.9) with the constrained social learning Protocol 1 and (2.6) with the benchmark Protocol 2.

3.7.3 Proof of Theorem 3.3.2

Proof. The aim here is to show that if $w_n = t_n (T'_{n-1})^{-1}$ then $\hat{\theta}_n$ defined in (3.18) is exactly equal to θ_n^{full} in (3.14). Before proceeding, let us first rewrite (3.18) and (3.14) using the following notations

$$\theta_n^{\text{full}} = \mathbf{v}_n + (t_n \otimes \mathbf{I}_d) \mathbf{v}_{1:n-1},$$

$$\widehat{\theta}_n = \mathbf{v}_n + (w_n \otimes \mathbf{I}_d) \widehat{\theta}_{1:n-1},$$
(3.29)

where $\widehat{\theta}_{1:n-1} \triangleq [\widehat{\theta}'_1, \dots, \widehat{\theta}'_{n-1}]'$, $v_{1:n-1} \triangleq [v'_1, \dots, v'_{n-1}]' \in \mathbb{R}^{(n-1)d \times 1}$. Here \otimes denotes Kronecker (tensor) product and \mathbf{I}_d denotes the $d \times d$ identity matrix.

To prove Theorem 3.3.2, we first start from

$$\widehat{\theta}_n = \theta_n^{\text{full}}.\tag{3.30}$$

Assume that (3.30) holds for all *i* where $1 \le i \le n$. From (3.18), θ_n^{full} can be written as (given that Eq. (3.30) holds)

$$\boldsymbol{\theta}_{n}^{\text{full}} = \widehat{\boldsymbol{\theta}}_{n} = (w_{n} \otimes \mathbf{I}_{d})\widehat{\boldsymbol{\theta}}_{1:n-1} + \boldsymbol{v}_{n}$$
$$= (w_{n} \otimes \mathbf{I}_{d})\boldsymbol{\theta}_{1:n-1}^{\text{full}} + \boldsymbol{v}_{n}. \tag{3.31}$$

Eq. (3.30) holds for all *i* where $1 \le i \le n$. Therefore, $\hat{\theta}_{1:n-1} = \theta_{1:n-1}^{\text{full}}$. From (3.14) in Theorem 3.3.1, $\theta_{1:n-1}^{\text{full}}$ can be expressed as

$$\boldsymbol{\theta}_{1:n-1}^{\text{full}} = \left(T_{n-1}'\right) \boldsymbol{v}_{1:n-1}. \tag{3.32}$$

Note that in the derivation of (3.32), we use the definition of t_{n-1} in (A.6) as the first n-2 elements of T_{n-1} and so on. Using (3.32), (3.31) can be written as

$$\boldsymbol{\theta}_{n}^{\text{full}} = \left(\boldsymbol{w}_{n} \otimes \mathbf{I}_{d}\right) \left(\boldsymbol{T}_{n-1}^{\prime}\right) \boldsymbol{v}_{1:n-1} + \boldsymbol{v}_{n}. \tag{3.33}$$

From (3.14) in Theorem 3.3.1, we have another expression for θ_n^{full} . Comparing (3.33) and (3.14) yields

$$(t_n \otimes \mathbf{I}_d) \mathbf{v}_{1:n-1} = (w_n \otimes \mathbf{I}_d) (T'_{n-1}) \mathbf{v}_{1:n-1}$$
$$= ((w_n T'_{n-1}) \otimes \mathbf{I}_d) \mathbf{v}_{1:n-1}.$$
(3.34)

Note that in going from the first line to the second line in (3.34), the distributive property of tensor products is used. From (3.34) it can be inferred that $t_n = w_n T'_{n-1}$. As presented in Appendix A, T_n is upper triangular matrix with ones in the diagonal. Therefore T_n is invertible and $w_n = (t_n T'_{n-1})^{-1}$. To complete the proof we need to start from $w_n = (t_n T'_{n-1})^{-1}$ and obtain $\hat{\theta}_n = \theta_n^{\text{full}}$. This part of proof

is straightforward and thus omitted from the chapter. Note that the topological Constraint 3.3.1 says that if $b_n(j) = 0$ then the *j*-th entry of $v_{1:n-1}$ is not available to the node *n* and thus the corresponding element of the weight vector $w_n(j)$ should be equal to zero as well. Also note that v_n is computed by the network administrator and the data incest free public belief, π_{-n} , is available to the network administrator.

Part II

Tracking Degree Distribution in Dynamic Social Networks

Tracking a Markov Modulated Degree Distribution

4.1 Introduction

Dynamic random graphs have been widely used to model social networks, biological networks [42] and Internet graphs [41]. Such dynamic models can be viewed as a sequence of graphs where the random graph at each time may depend on all the earlier graphs (snapshots of the evolving graph at earlier times) [41]. Motivated by analyzing social networks, we introduce *Markov-modulated duplication-deletion random graphs*³⁰ where at each time instant, nodes can either be added to or eliminated from the graph with probabilities that change according to a finite-state Markov chain. Such graphs mimic social networks where the interactions between nodes evolve over time according to a Markov process that undergoes infrequent jumps. An example of such a social network is the friendship network among residents of a city, where the dynamics of the network change in the event of a large festival.

Social networks can be viewed as complex sensors that provide information about interacting individuals and an underlying state of nature³¹. In this chapter, we consider a dynamic social network where at each time instant one node can join or leave the network. The probabilities of joining or leaving evolve according to the realization of a finite state Markov chain that represents the state of nature. This chapter presents two results. First, motivated by social network applications, the asymptotic behavior of the degree distribution of the Markov-modulated random graph is analyzed. Second, using noisy observations of nodes' connectivity, a "*social sensor*" is designed for tracking the underlying state of nature as it evolves over time.

4.1.1 Chapter Goals

As explained above, in this chapter, Markov-modulated dynamic random graphs are introduced to mimic social networks where the evolution of the network is varying over time. The most important parameter of a network that characterizes its structure is the degree distribution. It yields useful

³⁰The duplication-deletion procedure for Markov-modulated random graphs is described in Section 4.2.

³¹For example, real-time event detection from Twitter posts is investigated in [132] or the early detection of contagious outbreaks via social networks is studied in [40].

information about the connectivity of the random graph [10, 86, 116]. For example, if a majority of nodes in the random graph have relatively high degrees, the graph is highly connected and a message can be transferred between two arbitrary nodes with shorter paths. However, if a majority of nodes have smaller degrees then for transmitting a message throughout the network, longer paths are needed, see [80]. The degree distribution can further be used to investigate the diffusion of information or disease through social networks [108, 146]. The existence of a "giant component"³² in complex networks can be studied using the degree distribution. The size and existence of a giant component has important implications in social networks in terms of modeling information propagation and spread of human disease [62, 115, 118]. The degree distribution is also used to analyze the "searchability" of a network. The "search" problem arises when a specific node in a network faces a problem (request) whose solution is at other node, namely, destination (e.g., delivering a letter to a specific person, or finding a web page with specific information) [4, 146]. The searchability of a social network [146] is the average number of nodes that need to be accessed to reach the destination. Degree distribution is also used to investigate the robustness and vulnerability of a network in terms of the network response to attacks on its nodes or links [33, 76]. The papers [148, 149] further use degree-dependent tools for classification of social networks.

The fist goal of this chapter is to provide a degree distribution analysis that allows us to determine the relation between the structure of the network (in terms of connectivity) and the underlying state of nature. Indeed, it will be shown in Section 4.3 that there exists a unique stationary degree distribution for the Markov-modulated graph for each state of the underlying Markov chain. It thus suffices to estimate the degree distribution in order to track the underlying state of nature. The second goal of the chapter is to propose a stochastic approximation algorithm to track the empirical degree distribution of the Markov-modulated random graph. In particular, our goals are to address the following two questions in Section 4.4:

- How can a social sensor estimate (track) the empirical degree distribution using a stochastic approximation algorithm with no knowledge of the Markovian dynamics?
- How accurate are the estimates generated by the stochastic approximation algorithm when the random graph evolves according to the duplication-deletion model with Markovian switching?

By tracking the degree distribution of a Markov-modulated random graph, we can design a social sensor to track the underlying state of nature using the noisy measurements of nodes' connectivity.

 $^{^{32}}$ A giant component is a connected component with size O(n), where *n* is the total number of vertices in the graph. If the average degree of a random graph is strictly greater than one, then there exists a unique giant component with probability one [41], and the size of this component can be computed from the expected degree sequence.

4.1.2 Main Results and Organization of Chapter

Section 4.2 describes the construction of Markov-modulated duplication-deletion random graphs. Section 4.3 provides an asymptotic degree distribution analysis for the non-Markov modulated case of two different scenarios: (i) fixed size duplication-deletion random graph, and (ii) infinite duplication-deletion random graph. Theorem 4.3.1 in Section 4.3.1 asserts that the expected degree distribution of the fixed size Markov-modulated random graph at each time can be computed in terms of the expected degree distribution of the graph at the previous time and the dynamics of the graph via recursive equation (4.9). Section 4.3.2 extends the results of Section 4.3.1 to infinite random graphs. Theorem 4.3.2 parameterizes the degree distribution of such a graph by the *power law exponent* which depends on the dynamics of the graph.

Section 4.4 considers the problem of adaptively estimating the degree distribution of a fixed size Markov-modulated duplication-deletion random graph given observations of the degree distribution. A stochastic approximation algorithm is presented for tracking the degree distribution as it evolves over time. In particular, Section 4.4 presents three results regarding the tracking performance of the stochastic approximation algorithm:

- *Mean square error analysis:* Theorem 4.4.1 analyzes the asymptotic mean square error between the expected degree distribution and the estimate obtained via the stochastic approximation algorithm. Deriving this result uses error bounds on two-time scale Markov chains and perturbed Lyapunov function methods.
- *Weak convergence analysis:* Theorem 4.4.2 shows that the asymptotic behavior of the stochastic approximation algorithm converges weakly to the solution of a switched Markovian ordinary differential equation.
- *Functional central limit theorem for scaled tracking error:* Finally, Theorem 4.4.3 investigates the asymptotic behavior of the scaled tracking error. Similar to [94], it is shown that the interpolated scaled tracking error converges weakly to the solution of a switching diffusion process.

Section 4.5 extends the results of Section 4.4 to infinite (denumerable) duplication-deletion random graphs where the number of nodes in the graph (and so the support of degree distribution) is no longer fixed and increases over time. A Hilbert-space-valued stochastic approximation algorithm is proposed to track the degree distribution of the infinite graph with support on the set of non-negative integers. To study the tracking performance of such a Hilbert-space-valued stochastic approximation algorithm, limit system characterization and asymptotic analysis of scaled tracking error are provided. Numerical examples are presented in Section 4.6.

4.2 Markov-modulated Dynamic Random Graph of Duplication-deletion Type

This section outlines the construction of Markov-modulated dynamic random graphs of duplicationdeletion type. Let n = 0, 1, 2, ... denote discrete time. Denote by θ_n a discrete-time Markov chain with state space

$$\mathscr{M} = \{1, 2, ..., M\},\tag{4.1}$$

and initial probability distribution π_0 .

Assumption 4.2.1. The Markov chain θ_n evolves according to the transition matrix

$$A^{\rho} = I + \rho Q. \tag{4.2}$$

Here, I is an $M \times M$ identity matrix, ρ is a small positive real number, and $Q = [q_{ij}]$ is an irreducible³³ generator of a continues-time Markov chain satisfying

$$q_{ii} > 0 \text{ for } i \neq j, \text{ and } Q\mathbf{1} = \mathbf{0}, \tag{4.3}$$

where **1** and **0** represent column vectors of ones and zeros, respectively. The transition probability matrix A^{ρ} is therefore close to identity matrix. Here and henceforth, we refer to such a Markov chain θ_n as a "slow" Markov chain. The initial distribution π_0 is assumed independent of ρ .

A Markov-modulated duplication-deletion dynamic random graph is parameterized by the 7tuple $(M, A^{\rho}, \pi_0, r, p, q, G_0)$. Here, p and q are M-dimensional vectors with elements p(i) and $q(i) \in$ [0,1], i = 1, ..., M, where p(i) denotes the connection probability, and q(i) denotes the deletion probability. Also, $r \in [0,1]$ denotes the probability of the duplication step, and G_0 denotes the initial graph at time 0. In general, G_0 can be any finite simple connected graph. For simplicity, assume that G_0 is a simple connected graph with size N_0 . The duplication-deletion random graph is constructed via the duplication-deletion Procedure 4.5^{34} .

The Markov-modulated random graph generated by the duplication-deletion Procedure 4.5 mimics social networks where the interactions between nodes evolve over time due to the underlying dynamics (state of nature) such as seasonal variations (e.g., the high school friendship social network evolving over time with different winter/summer dynamics). In such cases, the connection/deletion

³³The irreducibility assumption implies that there exists a unique stationary distribution $\pi \in \mathbb{R}^{M \times 1}$ for this Markov chain such that $\pi' = \pi' A^{\rho}$.

 $^{^{34}}$ In Procedure 4.5, Step 1 is executed with probability *r*. Then, regardless of execution of Step 1, Step 2 is implemented. For convenience in the analysis, assume that a node generated in the duplication step cannot be eliminated in the deletion step immediately after its generation. Also, nodes whose degrees change in the edge-deletion part of Step 2, remain unchanged in the duplication part of Step 2 at that time instant. Finally, to prevent formation of isolated nodes, assume that the neighbor of a node with degree one cannot be eliminated in the deletion step. Note also that the duplication step in Step 2 ensures that the graph size does not decrease.

4.2. Markov-modulated Dynamic Random Graph of Duplication-deletion Type

Procedure 4.5 Markov-modulated Graph parameterized by $(M, A^{\rho}, \pi_0, r, p, q, G_0)$

At time *n*, given the graph G_n and Markov chain state θ_n , simulate the following events: **Step 1: Duplication step**: With probability *r* implement the following steps:

- Choose node u from graph G_n randomly with uniform distribution.
- Vertex-duplication: Generate a new node v.
- *Edge-duplication*:
 - Connect node *u* to node *v*. (A new edge between *u* and *v* is added to the graph.)
 - Connect each neighbor of node u with probability $p(\theta_n)$ to node v. These connection events are statistically independent.

Step 2: Deletion Step: With probability $q(\theta_n)$ implement the following steps:

- *Edge-deletion*: Choose node w randomly from G_n with uniform distribution. Delete node w along with the connected edges in graph G_n .
- Duplication Step: Choose a node from graph x from G_n randomly and implement Vertexduplication and Edge-duplication processes as described in Step 1.

Step 3: Denote the resulting graph by G_{n+1} . Generate θ_{n+1} (Markov chain) using transition matrix A^{ρ} . Set $n \to n+1$ and go to Step 1.

probabilities p,q evolve with time. Procedure 4.5 models these time variations as a finite state Markov chain θ_n with transition matrix A^{ρ} .

The Markov-modulated random graph generated by Procedure 4.5 mimics social networks where the interactions between nodes evolve over time due to the underlying dynamics (state of nature) such as seasonal variations (e.g., the high school friendship social network evolving over time with different winter/summer dynamics). In such cases, the connection/deletion probabilities p, q depend on the state of nature and evolve with time. Procedure 4.5 models these time variations as a finite state Markov chain θ_n with transition matrix A^{ρ} .

Discussion:

The connection/deletion probabilities p, q can be determined by the solution of a utility maximization problem. Let $U^{\text{join}} : [0,1] \times \mathcal{M} \to \mathbb{R}$ denote a utility function that gives payoff to an individual who considers to expand his neighbors in "Edge-duplication step" of Procedure 4.5 as a function of (p, θ) . Similarly, let $U^{\text{leave}} : [0,1] \times \mathcal{M} \to \mathbb{R}$ denote a utility function that pays off to an individual who considers to leave the network in "Deletion step" of Procedure 4.5 as a function of (q, θ) . With the above utility functions, the probabilities of connection/deletion when the state of nature is θ can be viewed as the solutions of the following maximization problems:

$$p(\theta) = \underset{p}{\operatorname{argmax}} \left\{ U^{\text{join}}(p, \theta) \right\},$$

$$q(\theta) = \underset{q}{\operatorname{argmax}} \left\{ U^{\text{leave}}(q, \theta) \right\}.$$
(4.4)

These utility functions can be interpreted in terms of mutual benefits and privacy concerns. One example could be $U^{\text{join}}(p,\theta) = b^{\text{join}}(p,\theta) - v$, where $b^{\text{join}}(p,\theta)$ is the benefit one obtains by expanding his network with probability p when the underlying state of nature is θ , and v is the cost incurred by sacrificing his "privacy". In this example, when an individual decides to leave the network, the utility he obtains will be $U^{\text{leave}}(q,\theta) = b^{\text{leave}}(q,v) - c(\theta)$, where $b^{\text{leave}}(q,v)$ is the benefit he earns by preserving privacy and $c(\theta)$ is the benefit he loses by leaving the network when the underlying state of nature is θ .

4.3 Asymptotic Degree Distribution Analysis for Non-Markov Modulated case

This section presents degree distribution analysis for duplication-deletion random graphs generated according Procedure 4.5 for the non-Markov modulated case, i.e., M = 1. The stationary degree distribution obtained in Section 4.3.1 below will be used in the Markov modulated case. The results in this section constitute a minor extension of [41] to the duplication-deletion random graphs.

Notation At each time *n*, let N_n denote the number of nodes of graph G_n . Also, let f_n be a N_n dimensional vector such that its *i*-th element, f_n^i , denotes the number of vertices of graph G_n with degree *i*. Clearly $f'_n \mathbf{1} = N_n$ where **1** denotes the vector of ones. Here, ' is used to denote the transpose of a vector or matrix. Define the "empirical vertex degree distribution" as

$$g_n = (g_n^i, i = 1, 2, ...), \text{ where } g_n^i = \frac{f_n^i}{N_n}.$$
 (4.5)

Note that g_n can be viewed as a probability mass function since all of its elements are non-negative and $g'_n \mathbf{1} = 1$.

4.3.1 Fixed Size Random Graph

This subsection analyzes the evolution of the expected degree distribution for a fixed size duplicationdeletion random graph generated according to Procedure 4.5 with r = 0, M = 1. (Recall *r* denotes the probability of Step 1 in Procedure 4.5.) Therefore, the number of vertices in the graph remains fixed, i.e., $N_n = N_0$ for n = 0, 1, 2, ... Theorem 4.3.1 below gives a recursion for the expected degree distribution of the fixed size Markov-modulated duplication-deletion random graph.

Theorem 4.3.1. Consider the fixed size duplication-deletion random graph generated according to Procedure 4.5, where r = 0, M = 1. Let \overline{g}_n denote the expected degree distribution of nodes at time *n*. Then, \overline{g}_n satisfies the recursion

$$\overline{g}_{n+1} = (I + \frac{1}{N_0}L')\overline{g}_n, \tag{4.6}$$

where *L* is a generator matrix³⁵ with elements (for $1 \le i, j \le N_0$):

$$l_{ji} = \begin{cases} 0, & j < i - 1, \\ qp^{i-1} + q(1 + p(i-1)), & j = i - 1, \\ iqp^{i-1}(1-p) - q(i+2+pi), & j = i, \\ q\binom{i+1}{i-1}p^{i-1}(1-p)^2 + q(i+1), & j = i + 1, \\ q\binom{j}{i-1}p^{i-1}(1-p)^{j-i+1}, & j > i + 1. \end{cases}$$
(4.7)

Proof. The proof is presented in Appendix 4.8.1.

Theorem 4.3.1 shows that evolution of the expected degree distribution in a fixed size Markovmodulated duplication-deletion random graph satisfies (4.6). One can rewrite (4.6) as

$$\overline{g}_{n+1} = B'_{N_0}\overline{g}_n$$
, where $B_{N_0} = I + \frac{1}{N_0}L.$ (4.8)

Since *L* is a generator matrix, B_{N_0} can be considered as the transition matrix of a slow Markov chain. It is also straightforward to show that B_{N_0} is irreducible and aperiodic³⁶. Hence, there exists a unique stationary distribution $\overline{g} = (\overline{g}^i, i = 1, 2, ...)$ such that

$$\overline{g} = B'_{N_0} \overline{g}. \tag{4.9}$$

The stationary distribution \overline{g} is the stationary expected degree distribution of a fixed size duplicationdeletion random graph generated according to Procedure 4.5 where r = 0. Note that the underlying Markov chain $\{\theta_n\}$ depends on the small parameter ρ . The main idea is that, although θ_n is timevarying but it is piecewise constant (since ρ is small parameter)—it changes slowly over time. Further, in light of (4.6), the evolution of \overline{g}_n depends on $\frac{1}{N_0}$. Our assumption throughout this chapter is that $\rho \ll \frac{1}{N_0}$. Therefore, the evolution of \overline{g}_n is faster than the evolution of θ_n . That is, \overline{g}_n reaches its stationary distribution \overline{g} before the state of θ_n changes. From (4.9), the expected degree distribution

³⁵That is, each row adds to zero and each non-diagonal element of L is positive.

³⁶It is straightforward to show that all elements of $(B_{N_0})^{N_0}$ are strictly greater than zero. Therefore, B_{N_0} is irreducible and aperiodic.

of the fixed size Markov-modulated duplication-deletion random graph can be uniquely computed for each state of the underlying Markov chain $\theta_n = \theta$.

Example: Searchability of a Network

So far in this section, an asymptotic analysis of the degree distribution was presented for a random graph generated according to Procedure 4.5. We now comment briefly on how the degree distribution can be used to investigate the searchability of the network. This also motivates the stochastic approximation algorithm presented in Section 4.4 as will be described below. The search problem arises in a network when a specific node faces a problem (request) whose solution is at other node (e.g., delivering a letter to a specific person or finding a web page with specific information). Assume [146] that on receiving a search request, each node follows the following protocol: (a) It address the request if it or its neighbors have the solution; otherwise (b) it relays the request to one of its neighbors chosen uniformly. The objective is to find the expected search delay, that is, the expected number of steps until the request is addressed.

Lemma 4.3.1. Consider the sequence of fixed size Markov-modulated duplication-deletion random graph obtained by Procedure 4.5, $\{G_n\}$, with $(M, A^{\rho}, \pi_0, \prime, q, p, G_0)$ where $A^{\rho} = I + \rho Q$ and p = 0 and expected degree distribution \overline{g}_n . The expected search delay is

$$\lambda(N_0) = O\left(\frac{N_0\delta}{\overline{d}_2 - \delta}\right),\tag{4.10}$$

as $n \to \infty$ where $\delta = \sum_{i=1}^{N_0} i \overline{g}_n(i)$ and $\overline{d}_2 = \sum_{i=1}^{N_0} i^2 \overline{g}_n(i)$.

Proof. See Chapter 5 of [146] and recall that size of the considered random graph is N_0 .

Lemma 4.3.1 implies that, if the empirical degree distribution of the possibly time-varying network can tracked accurately, then such an estimate can be used to track the searchability of the network. Also, using the estimated degree distribution and Lemma 4.3.1, we can address the following design problem as: *How can p and q in Algorithm 4.5 be chosen so that the average delay does not exceed a threshold?*

Using the stochastic approximation algorithm in (4.14) (see Section 4.4 below for the convergence proof), we can estimate the expected degree distribution, \hat{g}_n , and from that, we can compute δ and \overline{d}_2 . Then, from Lemma 4.3.1 we can find the measure of searchability and compare it with the maximum acceptable average delay and modify the parameters of Procedure 4.5 accordingly. We illustrate searchability in numerical examples given in Section 4.6.

4.3.2 Power Law Exponent for Infinite Duplication-deletion Random Graph

The degree distribution analysis provided in the previous subsection was for a fixed size random graph generated according to the duplication-deletion Procedure 4.5 with r = 0. This section extends this analysis to infinite duplication-deletion random graphs (obtained by choosing r = 1). Assume that G_0 is an empty set. Since r = 1, at time n, the graph G_n has n nodes. By employing the same approach as in the proof of Theorem 4.3.1, it will be shown that the infinite duplication-deletion random graph without Markovian dynamics satisfies a power law. An expression is further derived for the power law exponent.

Definition 4.3.1 (Power Law Distribution). *The degree distribution* $g = (g^i, i = 1, 2, ...)$, of a graph \mathscr{G} has a power law distribution³⁷ if there exists an integer i^* such that for all $i \ge i^*$,

$$\log g^i = \alpha - \beta \log i$$

where α is a constant³⁸ and $\beta > 1$. Parameter β is called the power law exponent.

The power law is satisfied in many networks such as WWW-graphs, peer-to-peer networks, phone call graphs, co-authorship graph and various massive online social networks (e.g. Yahoo, MSN, Facebook) [17, 27, 44, 135, 139]. The following theorem states that the graph generated according to Procedure 4.5 with r = 1 and M = 1 satisfies a power law.

Theorem 4.3.2. Consider the infinite random graph with Markovian dynamics G_n obtained by *Procedure 4.5 with 7-tuple* $(1,1,1,1,p,q,G_0)$ with the expected degree distribution \overline{g}_n . Then, if $\log p + p < \frac{q}{1+q} < p$, the expected degree of nodes in G_n has a power law distribution with exponent $\beta > 1$. The power law exponent is computed from

$$(1+q)(p^{\beta-1}+p\beta-p) = 1+\beta q.$$
(4.11)

Here, p and q are the probabilities defined in duplication and deletion steps, respectively.

Proof. The proof is similar to that of Theorem 4.3.1 with some modifications, see [72]. Here, we only present an outline of the proof which is comprised of two steps: (i) finding the power law exponent, and (ii) showing that the degree distribution converges to a power law with the computed exponent as $n \to \infty$. To find the power law exponent, we derive a recursive equation for the number of nodes with degree i + 1 at time n + 1, denoted by f_{n+1}^{i+1} , in terms of the degrees of nodes in

³⁷There is a difference between "power law" and "power law distribution". Power law is a functional relationship between two parameters where one parameter is proportional to the power of another, i.e., $x \propto y^{-\beta}$, where β can be any real number. In comparison, the exponent of a power law distribution is strictly greater that one [117]. Otherwise, the probability distribution does not add up to one.

³⁸The normalization constant α is computed from $\alpha = -\log[\zeta(\beta, i^*)]$, where $\zeta(\beta, i^*) = \sum_{k=i^*}^{\infty} k^{-\beta}$ denotes the incomplete Riemann ζ -function.

graph G_n . Then, rearranging this recursive equation yields an equation for the power law exponent. To prove that the degree distribution satisfies a power law, we show that $\lim_{n\to\infty} \sum_{k=1}^{i} \mathbf{E}\{f_n^k\} = \sum_{k=1}^{i} ck^{-\beta}$, where $\beta > 1$ is the power law exponent computed in the first step and f_n^k is the *k*-th element of f_n .

Theorem 4.3.2 asserts that the infinite duplication-deletion random graph without Markovian dynamics generated by Procedure 4.5 satisfies a power law and provides an expression for the power law exponent. The significance of this theorem is that it ensures, with use of one single parameter (the power law exponent), we can describe the degree distribution of graphs with relatively large number of nodes. The above result slightly extends [42, 121], where only a duplication model was considered. Theorem 4.3.2 allows us to explore characteristics (such as searchability, diffusion, and existence/size of the giant component) of large networks which can be modeled with the infinite duplication-deletion random graphs. Remark 1. Outline of Proof: The proof of Theorem 4.3.2, which is presented in Appendix 4.8.2, consists of two steps: (i) finding the power law component and (ii) showing that the degree distribution converges to a power law as $n \to \infty$. To find the power law component, we derive a recursive equation for the number of nodes with degree i+1 at time n+11, $f_{n+1}(i+1)$, in terms of degree of nodes in graph G_n . Then, this recursive equation is rearranged to equation for the power law component. To prove that the degree distribution satisfies a power law, we define a new parameter $h_n(i) = \frac{1}{n} \sum_{k=1}^{i} \mathbf{E} \{f_n(k)\}$ and we show that $\lim_{n\to\infty} h_n(i) = \sum_{k=1}^{i} Ck^{-\beta}$ where β is the power law component computed by the solving the recursive equation. Theorem 4.3.2 asserts that the infinite duplication-deletion random graph without Markovian dynamics generated by Procedure 4.5 satisfies a power law and provides an expression for the power law component. The significance of this theorem is that it ensures that with use of one single parameter (the power law component), we can describe the degree distribution of large numbers of nodes in graphs that model social networks.

Remark 2. Power Law Component: Let β^* denote the solution of (4.11). Then the power law component is defined as $\beta = \max\{1, \beta^*\}$. Fig.4.1 shows the the power law component and β^* versus *p* for different values of probability of deletion, *q*. As can be seen in Fig.4.1, the power law component is increasing in *q* and decreasing in *p*.

4.4 Estimating (Tracking) the Degree Distribution of the Fixed Size Markov-modulated Duplication-deletion Random Graph

In Section 4.2, an expression was given for the unique stationary degree distribution \overline{g} for the non-Markov modulated case, see (4.9). In this section, we consider fixed size Markov modulated duplication deletion random graphs. Consider Procedure 4.5 and assume that there are *M* possible stationary degree distributions, namely $\mathbf{g} = \{\overline{g}(1), \overline{g}(2), \dots, \overline{g}(M)\}$ corresponding to the *M* states



Figure 4.1: The power law component for the non-Markovian random graph generated according to Procedure 4.5 obtained by (4.11) for different values of p and q in Procedure 4.5.

of a Markov chain. Here each $\overline{g}(i)$ is computed using (4.9) where the corresponding parameters p(i), q(i) are used. At each time *n*, a stationary distribution $\overline{g}(\theta_n) \in \mathbf{g}$ is chosen where θ_n evolves according to an *M*-state Markov chain as described in Section 4.2. We assume that the stationary degree distribution of the graph is sampled by a network administrator. How can the network administrator track the expected degree distribution of the fixed size Markov-modulated duplication deletion random graph without knowing the dynamics of the graph? The motivation for tracking the stationary degree distribution stems from social networks where the dynamics of the degree distribution evolve on a faster time scale than the Markov chain θ_n . Therefore, it suffices to track $\overline{g}(\theta_n)$ given observations.

At each time *n*, the network administrator samples a node from the graph based on degree distribution $\overline{g}(\theta_n)$ and records its degree $y_n(\theta_n)$. Let $z_n(\theta_n) = \mathbf{e}_{y_n(\theta_n)}$ denote the observation vector where $\mathbf{e}_i \in \mathbb{R}^{N_0 \times 1}$ is the *i*-th standard unit vector. Such a sampling procedure can be time correlated. Therefore, we allow $z_n(\theta_n)$ to be a mixing process with the following assumption:

Assumption 4.4.1. For each $\theta \in \mathcal{M}$, the sequence $\{y_n(\theta)\}$ is stationary ϕ -mixing with sufficiently fast mixing rate such that the sequence $\{y_n(1), \ldots, y_n(M)\}$ is independent of $\{\theta_n\}$ and that for each $\theta \in \mathcal{M}$, $\{z_n(\theta)\}$ is a stationary ϕ -mixing sequence with mixing rate ψ_n satisfying $\sum_{j=0}^{\infty} \psi_j^{1/2} < \infty$.

Remark 4.4.1. Because $\{y_n(\theta)\}$ is a stationary ϕ -mixing sequence for each $\theta \in \mathcal{M}$, $\{z_n(\theta)\}$ is

a bounded sequence of ϕ -mixing process for each $\theta \in \mathcal{M}$ [97, p. 82] (see also [25, p.170]). The stationarity implies that

$$\mathbf{E}z_n(\theta) = \mathbf{E}z_1(\theta) = \sum_{i=0}^{\infty} \mathbf{e}_i P(y_1(\theta) = i) = \sum_{i=0}^{\infty} \overline{g}^i(\theta) \mathbf{e}_i = \overline{g}(\theta)$$
(4.12)

The mixing rate given requires that for any positive integers i and j,

$$\begin{aligned} \|\mathbf{E}_{k}I\{y_{n}(\boldsymbol{\theta})=i\}-\overline{g}^{i}(\boldsymbol{\theta})\| &\leq \psi_{n-k} \text{ for } n \geq k, \\ \|\mathbf{E}[I\{y_{l}(\boldsymbol{\theta})=j\}-\overline{g}^{j}(\boldsymbol{\theta})][I_{\{y_{n}=i\}}-\overline{g}^{i}(\boldsymbol{\theta})]\| &\leq \psi_{n-l}^{1/2}\psi_{l-k}^{1/2} \text{ for any } k < l < n, \end{aligned}$$

$$(4.13)$$

where \mathbf{E}_k denotes the conditional expectation on the past data up to time k (i.e., condition on the σ -algebra \mathscr{F}_k generated by $\{z_j(\theta) : j \le k\}$) and $I\{\cdot\}$ denotes the indicator function. Here, $\|\cdot\|$ is used to denote the Euclidean norm.

The analysis in this chapter can be generalized to include certain non-stationary cases for the observation process $\{y_n(\theta)\}$. For example, for each $\theta \in \mathcal{M}$, suppose $\{\zeta_n(\theta)\}$ is an ergodic finite state Markov chain³⁹. Let $y_n(\theta) = \tilde{f}(\zeta_n(\theta))$. The n-step transition probability matrix of the Markov chain converges to a matrix (with identical rows consisting of its stationary distribution) with exponential rate. Then it can be verified similar to [25, pp.178] that $y_n(\theta)$ is mixing. Although (4.12) does not hold, the analysis using mixing inequalities can still be obtained.

Given the observation sequence $z_n(\theta_n)$, n = 0, 1, 2, ..., the aim is to adaptively estimate $\overline{g}(\theta_n)$ via the following stochastic approximation algorithm with (small positive) constant step-size ε :

$$\widehat{g}_{n+1} = \widehat{g}_n + \varepsilon \left(z_n(\theta_n) - \widehat{g}_n \right), \quad \widehat{g}_0 = \mathbf{e}_1$$
(4.14)

To summarize, the evolution of the slow Markov chain θ_n and stochastic approximation algorithm (4.14) form a two-time-scale Markovian system as follows when $\rho, \varepsilon = o(\frac{1}{N_0})$

True system:
$$\overline{g}(\theta_n) \in \{\overline{g}(1), \dots, \overline{g}(M)\}$$
, where θ_n evolves according to $A^{\rho} = I + \rho Q$,
Algorithm: $\widehat{g}_{n+1} = \widehat{g}_n + \varepsilon (z_n(\theta_n) - \widehat{g}_n), \quad z_n(\theta_n) = \mathbf{e}_{y_n(\theta_n)}$, where $y_n(\theta_n) \sim \overline{g}(\theta_n)$. (4.15)

Note that the stochastic approximation algorithm (4.14) does not assume any knowledge of the Markov-modulated dynamics of the graph. The Markov chain assumption for the random graph dynamics is only used in our convergence and tracking analysis. By means of the stochastic approximation (4.14), the network administrator can track the stationary expected degree distribution $\overline{g}(\theta_n)$.

³⁹Respondent driven sampling (RDS) was introduced in [74] as an approach for sampling from hidden populations in social networks. RDS has been selected by the U.S. Centers for Disease Control and Prevention as part of the HIV behavioral surveillance system. RDS can be viewed as a form of Markov Chain Monte Carlo sampling [68].

4.4.1 Tracking Error of the Stochastic Approximation Algorithm

The goal here is to analyze how well algorithm (4.14) tracks the empirical degree distribution of the fixed size Markov-modulated duplication-deletion graph. Define the tracking error as $\tilde{g}_n = \hat{g}_n - \overline{g}(\theta_n)$. Theorem 4.4.1 below shows that the difference between the sample path and the stationary degree distribution is small—implying that the stochastic approximation algorithm can successfully track the Markov-modulated node distribution given the noisy measurements. We again emphasize that no knowledge of the Markov chain parameters are required in the algorithm. It also finds the order of this difference in terms of ε and ρ .

Theorem 4.4.1. Consider the random graph $(M, A^{\rho}, \pi_0, p, q, r, G_0)$. Suppose $\rho^2 \ll \varepsilon$ and Assumptions 4.2.1 and 4.4.1 hold⁴⁰. Then, for sufficiently large *n*, the tracking error of the stochastic approximation algorithm (4.14) is

$$\mathbf{E} \|\tilde{g}_n\|^2 = O\left(\varepsilon + \rho + \frac{\rho^2}{\varepsilon}\right).$$
(4.16)

Proof. The proof uses the perturbed Lyapunov function method and is provided in Appendix 4.8.5. \Box

Remark 4.4.2. Most existing literature analyzes stochastic approximation algorithms for tracking a parameter that evolves according to a "slowly time-varying" sample path of a continuous-valued process so that the parameter changes by small amounts over small intervals of time. When the rate of change of the underlying parameter is slower than the adaptation rate of the stochastic approximation algorithm (e.g., a slow random walk), the mean square tracking error can be analyzed as in [19, 69, 99, 111, 127, 137]. In comparison, our analysis covers the case where the underlying parameter evolves with discrete jumps that can be arbitrarily large in magnitude on short intervals of time. Also, the jumps occur on the same time scale as the speed of adaptation of the stochastic approximation algorithm. We explicitly consider this Markovian time-varying parameter in our mean square error and weak convergence analysis.

As a corollary of Theorem 4.4.1, we obtain the following mean square error convergence result.

Corollary 4.4.1. Under the conditions of Theorem 4.4.1, if $\rho = O(\varepsilon)$,

$$\lim_{n\to\infty}\mathbf{E}\|\widetilde{g}_n\|^2=O(\varepsilon).$$

Therefore,

$$\lim_{\varepsilon \to 0} \lim_{n \to \infty} \mathbf{E} \|\widetilde{g}_n\|^2 = 0$$

⁴⁰In this work, we assume that $\rho = O(\varepsilon)$. Therefore, $\rho^2 \ll \varepsilon$.

4.4.2 Limit System of Regime-Switching Ordinary Differential Equations

The following theorem asserts that the sequence of estimates generated by the stochastic approximation algorithm (4.14) follows the dynamics of a Markov-modulated ordinary differential equation (ODE).

Before proceeding with the main theorem below, let us recall a Definition.

Definition 4.4.1 (Weak Convergence). Let Z_k and Z be \mathbb{R}^r -valued random vectors. We say Z_k converges weakly to $Z(Z_k \Rightarrow Z)$ if for any bounded and continuous function $f(\cdot)$, $Ef(Z_k) \rightarrow Ef(Z)$ as $k \rightarrow \infty$.

Weak convergence is a generalization of convergence in distribution to a function space⁴¹.

Theorem 4.4.2. Consider the Markov-modulated random graph generated according to Procedure 4.5, and the sequence of estimates $\{\hat{g}_n\}$, generated by the stochastic approximation algorithm (4.14). Suppose Assumptions 4.2.1 and 4.4.1 hold and $\rho = O(\varepsilon)$. Define the continuous-time interpolated process

$$\widehat{g}^{\varepsilon}(t) = \widehat{g}_n, \ \theta^{\varepsilon}(t) = \theta_n \ \text{for } t \in [n\varepsilon, (n+1)\varepsilon).$$
(4.17)

Then, as $\varepsilon \to 0$, $(\widehat{g}^{\varepsilon}(\cdot), \theta^{\varepsilon}(\cdot))$ converges weakly to $(\widehat{g}(\cdot), \theta(\cdot))$, where $\theta(\cdot)$ is a continuous-time Markov chain with generator Q, $\widehat{g}(\cdot)$ satisfies the Markov-modulated ODE

$$\frac{d\widehat{g}(t)}{dt} = -\widehat{g}(t) + \overline{g}(\theta(t)), \quad \widehat{g}(0) = \widehat{g}_0$$
(4.18)

and $\overline{g}(\theta) \in \mathbf{g}$.

The above theorem asserts that the limit system associated with the stochastic approximation algorithm (4.14) is a Markovian switched ODE (4.18). As mentioned in Section 4.1, this is unusual since typically in averaging of stochastic approximation algorithms, convergence occurs to a deterministic ODE. The intuition behind this is that the Markov chain evolves on the same time-scale as the stochastic approximation algorithm. If the Markov chain evolved on a faster time-scale, then the limiting dynamics would be a deterministic ODE weighed by the stationary distribution of the Markov chain. If the Markov chain evolved slower than the dynamics of the stochastic approximation algorithm, then the asymptotic behavior would also be a deterministic ODE with the Markov chain being a constant.

4.4.3 Scaled Tracking Error

Next, we study the behavior of the scaled tracking error between the estimates generated by the stochastic approximation algorithm (4.14) and the expected degree distribution. The following theorem states that the tracking error should also satisfy a switching diffusion equation and provides a

⁴¹We refer the interested reader to [99, Chapter 7] for further details on weak convergence and related matters.

functional central limit theorem for this scaled tracking error. Let $v_k = \frac{\widehat{g}_k - \overline{g}(\theta_k)}{\sqrt{\varepsilon}}$ denote the scaled tracking error.

Theorem 4.4.3. Suppose Assumptions 4.2.1 and 4.4.1 hold. Define $v^{\varepsilon}(t) = v_k$ for $t \in [k\varepsilon, (k+1)\varepsilon)$. Then, $(v^{\varepsilon}(\cdot), \theta^{\varepsilon}(\cdot))$ converges weakly to $(v(\cdot), \theta(\cdot))$ such that $v(\cdot)$ is the solution of the following Markovian switched diffusion process

$$\mathbf{v}(t) = -\int_0^t \mathbf{v}(s)ds + \int_0^t \Sigma^{\frac{1}{2}}(\boldsymbol{\theta}(\tau))d\boldsymbol{\omega}(\tau).$$
(4.19)

Here, $\omega(\cdot)$ *is an* \mathbb{R}^{N_0} *-dimensional standard Brownian motion. The covariance matrix* $\Sigma(\theta)$ *in* (4.19) *can be explicitly computed as*

$$\Sigma(\theta) = Z(\theta)' D(\theta) + D(\theta) Z(\theta) - D(\theta) - \overline{g}(\theta) \overline{g}'(\theta).$$
(4.20)

Here, $D(\theta) = \operatorname{diag}(\overline{g}(\theta))$ and $Z(\theta) = (I - B_{N_0}(\theta) + \mathbf{1}\overline{g}'(\theta))^{-1}$, where $\overline{g}(\theta) \in \mathbf{G}$. For each $\theta \in \mathcal{M}$, $B_{N_0}(\theta)$ is computed using (4.8) where the corresponding parameters p(i), q(i) are used.

For general switching processes, we refer to [157]. In fact, more complex continuous-state dependent switching rather than Markovian switching are considered there. Equation (4.20) reveals that the covariance matrix of the tracking error depends on $B_{N_0}(\theta)$ and $\overline{g}(\theta)$ and, consequently, on the parameters p and q of the random graph. Recall from Section 4.2 that $B_{N_0}(\theta)$ is the transition matrix of the Markov chain which models the evolution of the expected degree distribution in duplication-deletion random graphs and can be computed from Theorem 4.3.1.

4.5 Estimating the Degree Distribution of Infinite Duplication-deletion Random Graphs

This section has two results: First, the results of Section 4.4 are extended to infinite random graphs without Markovian dynamics generated according to Procedure 4.5. Second, we show how this analysis can be extended to Markov-modulated probability mass functions with denumerable support. The analysis is non-standard, since it is formulated on a Hilbert space.

4.5.1 Infinite Random Graphs without Markovian Dynamics

Consider the infinite duplication-deletion random graph without Markovian dynamics generated according to Procedure 4.5 with 7-tuple $(1,1,1,1,p,q,G_0)$. In this section, let g_n represent the degree distribution of the infinite graph with support on the set of non-negative integers; its elements are denoted by g_n^i , i = 0, 1, 2, ... Recall from Section 4.3.2 that, the size of such a graph increments at each time by one and thus the size of the graph at time *n* is equal to *n*; that is $N_n = n$. Therefore,
the maximum degree of the graph at time *n* cannot exceed n-1 and $g_n^j = 0$ for $j \ge n$. Similar to the proof of Theorem 4.3.1, the following theorem asserts that the expected degree distribution of the infinite duplication-deletion random graph satisfies a recursive equation.

Theorem 4.5.1. Consider the infinite duplication-deletion random graph without Markovian dynamics generated according to Procedure 4.5 with 7-tuple $(1,1,1,1,p,q,G_0)$, Let $\overline{g}_n = \mathbf{E}\{g_n\}$ denote the expected degree distribution of nodes with support on the set of non-negative integers. Then, \overline{g}_n satisfies the following recursion

$$\overline{g}_{n+1} = \overline{g}_n + \frac{1}{n} L^{(n)'} \overline{g}_n, \qquad (4.21)$$

where $L^{(n)}$ is a generator matrix of infinite size with elements:

$$l_{ji}^{(n)} = \begin{cases} (1+q) \left(p^{i-1} + 1 + p(i-1) \right), & j = i-1 \text{ and } 1 \le i, j \le n \\ (1+q) \left(ip^{i-1}(1-p) + 1 + pi \right) - q(i+1), & j = i \text{ and } 1 \le i, j \le n \\ (1+q) \binom{i+1}{i-1} p^{i-1}(1-p)^2 + q(i+1), & j = i+1 \text{ and } 1 \le i, j \le n \\ (1+q) \binom{j}{i-1} p^{i-1}(1-p)^{j-i+1}, & j > i+1 \text{ and } 1 \le i, j \le n, \\ 0, & \text{otherwise} \end{cases}$$
(4.22)

Proof. The proof is similar to the proof of Theorem 4.3.1 and is omitted due to the lack of space. \Box

Remark 4.5.1. Theorem 4.3.2 in Section 4.3.2 asserts that the expected degree distribution converges to a power law probability distribution \overline{g} with exponent $\beta > 1$, if $\log p + p < \frac{q}{1+q} < p$; that is $\lim_{n\to\infty} \overline{g}_n^i = \frac{i^{-\beta}}{\zeta(\beta)}$. We assume that the dynamics of the degree distribution evolve on a faster time scale than the stochastic approximation algorithm. Therefore, it suffices to track the stationary degree distribution \overline{g} given observations.

At each time *n*, the network administrator samples from the graph and records the degree of a randomly chosen vertex of the graph which is denoted by y_n . Let $z_n = \mathbf{e}_{y_n}$ denote the observation vector. Here, \mathbf{e}_i is the *i*-th standard unit vector with support on the set of non-negative integers (i.e., $\mathbf{e}_i = (0, ..., 1, ...)' \in \mathbb{R}^{\infty}$). The following stochastic approximation algorithm is used to estimate the expected degree distribution of the graph from such samples.

$$\widehat{g}_{n+1} = \widehat{g}_n + \varepsilon \left(z_n - \widehat{g}_n \right). \tag{4.23}$$

Here, $\varepsilon > 0$ denote a small positive step size and $\hat{g}_0 = \mathbf{e}_1$. Therefore, (4.23) is a Hilbert-space-valued stochastic approximation algorithms. By means of the stochastic approximation (4.23), the network administrator can track the expected degree distribution of the infinite graph size increases over time.

Define

$$\widehat{g}^{\varepsilon}(t) = \widehat{g}_n$$
 for $t \in [n\varepsilon, n\varepsilon + \varepsilon)$.

Then $\hat{g}^{\varepsilon}(\cdot) \in D([0,\infty) : \ell_2)$ the space of functions defined on $[0,\infty)$ taking values in $\ell_2 = \{z \in \mathbb{R}^{\infty} : \sum_{i=0}^{\infty} ||z_i||^2 < \infty\}$ such that the functions are right continuous and have left limits endowed with the Skorohod topology. Here, we obtain a weak convergence result of the interpolated sequence of iterates. Theorem 4.5.2 below asserts that the mean square tracking error is bounded and shows that the sequnce of estimates obtained by (4.23) converge to the solution of an ODE. Before proceeding to the main theorem, we shall use the following conditions.

Theorem 4.5.2. Suppose Assumption 4.4.1 holds with the modification that M = 1, i.e., there is no Markovian dynamics. Define $\tilde{g}_n = \overline{g} - \hat{g}_n$ Then, $\lim_{n\to\infty} \mathbf{E} ||\tilde{g}_n||^2 = O(\varepsilon)$. Also, $\hat{g}^{\varepsilon}(\cdot)$ is tight in $D([0,\infty) : \ell_2)$. Any convergent subsequence has a limit $\hat{g}(\cdot)$ that is the solution of the differential equation

$$\frac{d\widehat{g}(t)}{dt} = \overline{g} - \widehat{g}(t), \ \widehat{g}(0) = \mathbf{e}_1.$$
(4.24)

Proof. The proof is presented in Appendix 4.8.8. The proof of the theorem is divided into several steps, which uses techniques in stochastic approximation [99] but with the modification that ℓ_2 is a Hilbert space (see [61, 98]).

The above result concerns $n \to \infty$ and $\varepsilon \to 0$, εn remains to be bounded. We next obtain a result with $\varepsilon \to 0$, $n \to \infty$, $\varepsilon n \to \infty$.

Corollary 4.5.1. Consider $\hat{g}^{\varepsilon}(\cdot + t_{\varepsilon})$, where $t_{\varepsilon} \to \infty$ as $\varepsilon \to 0$. Under the condition of Theorem 4.5.2, $\hat{g}^{\varepsilon}(\cdot + t_{\varepsilon}) \to \overline{g}$ in probability as $\varepsilon \to 0$.

Proof Note that $\{\widehat{g}_k\}$ is tight. Define $\widehat{g}^{\varepsilon,\text{large}}(\cdot) = \widehat{g}^{\varepsilon}(\cdot + t_{\varepsilon})$. Using the same approach, we can show that $\{\widehat{g}^{\varepsilon,\text{large}}(\cdot)\}$ is tight. We extract a weakly convergent subsequence of $(\widehat{g}^{\varepsilon,\text{large}}(\cdot), \widehat{g}^{\varepsilon,\text{large}}(\cdot - T))$ with limit denoted by $(\widehat{g}(\cdot), \widehat{g}_T(\cdot))$. We note that $\widehat{g}(0) = \widehat{g}_T(T)$ and that $\widehat{g}_T(0)$ belongs to a set that is bounded in probability. Writing it in variational form, we obtain

$$\widehat{g}_T(T) = e^{-T} \widehat{g}_T(0) + \int_0^T e^{-(T-t)} \overline{g} dt = e^{-T} \widehat{g}_T(0) + \overline{g} - \int_T^\infty e^{-t} dt$$

$$\to \overline{g} \text{ as } T \to \infty.$$

The desired result then follows.

To study the rate of variation of estimation error, we define the sequence of scaled estimation error $v_n = (\hat{g}_n - \overline{g})/\sqrt{\epsilon}$. Theorem 4.5.3 asserts that the scaled estimation error satisfy a differential equation and provides a weak convergence results for it.

Theorem 4.5.3. Suppose assumptions of Theorem 4.5.2 hold. Then, for sufficiently small ε there is an N_{ε} such that $\mathbf{E}\{\langle v_n, v_n \rangle\} = O(1)$ for all $n > N_{\varepsilon}$. Define the sequence of continuous-time interpolation of estimation error as

$$\mathbf{v}^{\varepsilon}(t) = \mathbf{v}_n \text{ for } t \in [(n - N_{\varepsilon})\varepsilon, (n - N_{\varepsilon} + 1)\varepsilon).$$

Under the assumptions of Theorem 4.5.2, $\{v^{\varepsilon}(\cdot)\}$ is tight in $D([0,\infty);\ell_2)$. Moreover, suppose that $v^{\varepsilon}(0)$ converges weakly to v(0), $v^{\varepsilon}(\cdot)$ converges weakly to $v(\cdot)$ such that $v(\cdot)$ is the solution of the following stochastic differential equation

$$d\mathbf{v}(t) = -\mathbf{v}(t)dt + dW(t). \tag{4.25}$$

Here, $W(t) = \sum_{i=0}^{\infty} W_i(t) \mathbf{e}_i$ and the covariance operator is given by

$$\mathbf{E}\langle W(t), v \rangle \langle W(t), z \rangle = t \langle z, \Gamma v \rangle = t \sum_{i=0}^{\infty} \sigma_i^2 \langle \mathbf{e}_i, v \rangle \langle \mathbf{e}_i, z \rangle \text{ for } v, z \in \ell_2,$$
(4.26)

where $W_i(\cdot)$ is a real-valued Wiener process with covariance $t\sigma_i^2$ and

$$\sigma_i^2 = \mathbf{E}[\langle z_0 - \overline{g}, \mathbf{e}_i \rangle]^2 + 2\sum_{j=1}^{\infty} \mathbf{E} \langle z_0 - \overline{g}, \mathbf{e}_i \rangle \langle z_j - \overline{g}, \mathbf{e}_i \rangle.$$

Proof. The proof is presented in Appendix 4.8.9.

Note that the covariance σ_i^2 depends on the stationary expected degree distribution \overline{g} and thus is a function of the power law exponent β .

4.5.2 Markov-modulated Probability Mass Functions with Denumerable Support

Here, we extend the above results to the problem of tracking a time-varying probability mass function with infinite support. The aim is to track a probability mass function with support on the set of non-negative integers that evolves according to a slow Markov chain θ_n with M states and initial probability distribution π_0 . The state space \mathcal{M} , and the transition probability matrix A^{ρ} of the underlying Markov chain θ_n are defined in (4.1) and (4.2), respectively. For each $\theta \in \mathcal{M}$, let

$$\overline{g}(\theta) = [\overline{g}^1(\theta), \overline{g}^2(\theta), \dots]', \tag{4.27}$$

be a probability mass function with support on the set of non-negative integers such that $\sum_{i=1}^{\infty} \overline{g}^i(\theta) = 1$ and $\overline{g}^i(\theta) \propto i^{-\beta_{\theta}}$, where $\beta_{\theta} > 1$. When the underlying Markov chain θ_n jumps from one state to another within $\mathcal{M}, \overline{g}(\theta_n)$ switches accordingly.

At each time *n*, we sample $y_n(\theta_n)$ from PMF $\overline{g}(\theta_n)$; that is $y_n(\theta_n) \sim \overline{g}(\theta_n)$. Let $z_n(\theta_n) = \mathbf{e}_{y_n(\theta_n)}$ denote the observation vector. To estimate $\overline{g}(\theta_n)$, the following constant step size stochastic approximation algorithm is deployed

$$\widehat{g}_{n+1} = \widehat{g}_n + \varepsilon(z_n(\theta_n) - \widehat{g}_n).$$
(4.28)

Here $\varepsilon > 0$ denotes a small positive step size and $\hat{g}_0 = \mathbf{e}_1$. We further assume that the Markov chain is slowly changing in that the rate of changes is an order slower than that of adaptation (4.28); that is $\rho = \varepsilon^2$.

To analyze the asymptotic properties of the stochastic approximation algorithm, we define the sequence of continuous-time interpolation $\hat{g}^{\varepsilon}(t) = \hat{g}_n$ for $t \in [n\varepsilon, n\varepsilon + \varepsilon)$. Similar to what have been obtained thus far for the non-Markovian case, with the details omitted, we obtain the following weak convergence results. Theorem 4.5.4 states that the sequence of estimates obtained via Hilbert-space-valued stochastic approximation algorithm (4.28) converges weakly to the solution of an ODE which depends on the initial distribution of the underlying Markov chain.

Theorem 4.5.4. Suppose Assumptions 4.2.1 and 4.4.1 hold. Then $\hat{g}^{\varepsilon}(\cdot)$ is tight in $D([0,\infty) : \ell_2)$. Any convergent subsequence has a limit $\hat{g}(\cdot)$ that is the solution of the differential equation

$$\frac{d\widehat{g}(t)}{dt} = \sum_{\theta=1}^{M} \overline{g}(\theta) p_{\theta} - \widehat{g}(t), \ \widehat{g}(0) = \mathbf{e}_{1},$$
(4.29)

where

$$\overline{g}(\theta) = \sum_{i=0}^{\infty} \overline{g}^i(\theta) \mathbf{e}_i, and \quad (p_{\theta}: \theta \le M) = \pi_0$$

is the initial probability distribution of Markov chain.

Proof. The proof is presented in Appendix 4.8.10.

Furthermore, we can obtain the following corollary. The proof is similar to that of Corollary 4.5.1 and thus omitted.

Corollary 4.5.2. Consider $\widehat{g}^{\varepsilon}(\cdot + t_{\varepsilon})$, where $t_{\varepsilon} \to \infty$ as $\varepsilon \to 0$. Under the condition of Theorem 4.5.4, $\widehat{g}^{\varepsilon}(\cdot + t_{\varepsilon}) \to g = \sum_{\theta=1}^{M} p_{\theta}\overline{g}(\theta)$ in probability as $\varepsilon \to 0$.

Redefine $v_n = (\widehat{g}_n - \underline{g})/\sqrt{\varepsilon}$. It can be shown that there exists N_{ε} such that the sequence $\{v_n : n \ge N_{\varepsilon}\}$ is tight. Next, redefine $v^{\varepsilon}(t) = v_n$ for $t \in [\varepsilon(n - N_{\varepsilon}), \varepsilon(n - N - \varepsilon) + \varepsilon)$. With a little more effort, we can also obtain the associated rates of convergence result, which is stated in the next theorem.

Theorem 4.5.5. Suppose Assumptions 4.2.1 and 4.4.1 hold. Then, $\{v^{\varepsilon}(\cdot)\}$ is tight in $D([0,\infty); \ell_2)$. Moreover, suppose that $v^{\varepsilon}(0)$ converges weakly to v(0), then $v^{\varepsilon}(\cdot)$ converges weakly to $v(\cdot)$ such that $v(\cdot)$ is the solution of the following stochastic differential equation (SDE)

$$du(t) = -\mathbf{v}(t)dt + \sum_{\theta=1}^{M} p_{\theta}dW(\theta, t), \qquad (4.30)$$

where for each $\theta \in \mathcal{M}$, $W(\theta, \cdot)$ is a Wiener process as given in Theorem 4.5.3.

Proof. The proof is similar to the proof of Theorem 4.5.3 with modifications similar to those of the proof of Theorem 4.5.4. \Box

4.6 Numerical Examples

In this section, numerical examples are given to illustrate the results from Section 4.2, and Section 4.4.

The main conclusions are:

- 1. The infinite duplication-deletion random graph without Markovian dynamics generated by the duplication-deletion Procedure 4.5 satisfies a power law as stated in Theorem 4.3.2; see Example 4.6.1.
- 2. The degree distribution of the fixed size duplication-deletion random graph generated by the duplication-deletion Procedure 4.5 can be computed from Theorem 4.3.1. When N_0 (the size of the random graph) is sufficiently large, numerical results show that the degree distribution satisfies a power law as well; see Example 4.6.2.
- 3. The estimates obtained by stochastic approximation algorithm (4.14) follow the expected probability distribution precisely without information about the Markovian dynamics; see Example 4.6.3.
- 4. The larger the trace of the asymptotic covariance of the scaled tracking error, the greater the average degree of nodes and the searchability of the graph. This is illustrated in Example 4.6.4 below.

Example 4.6.1. Consider an infinite duplication-deletion random graph without Markovian dynamics (so M = 1) generated by Procedure 4.5 with p = 0.5 and q = 0.1. Theorem 4.3.2 implies that the degree sequence of the resulting graph satisfies a power law with exponent computed using (4.11). Fig.4.2 displays the un-normalized degree distribution on a logarithmic scale. The linearity in Fig.4.2 (excluding the nodes with very small degree), implies that the resulting graph from duplication-deletion process satisfies a power law. As can be seen in Fig.4.2, the power law is a better approximation for the middle points compared to both ends.

Example 4.6.2. Consider the fixed size duplication-deletion random graph generated by Procedure 4.5 with r = 0, $N_0 = 10$, p = 0.4, and q = 0.1. We consider M = 1 (no Markovian dynamics) to illustrate Theorem 4.3.1. Fig. 4.3 depicts the normalized degree distribution of the fixed size duplication-deletion random graph obtained by Theorem 4.3.1. As can be seen in Fig. 4.3, the computed degree distribution is close to that obtained by simulation. The numerical results show that the degree distribution of the fixed size random graph also satisfies a power law for some values of p when the size of random graph is sufficiently large. Fig. 4.4 shows the number of nodes with specific degree for the fixed size random graph obtained by Procedure 4.5 with r = 0, $N_0 = 1000$, p = 0.4, and q = 0.1 on a logarithmic scale for both horizontal and vertical axes.

Example 4.6.3. Consider the fixed size Markov-modulated duplication-deletion random graph generated by Procedure 4.5 with r = 0 and $N_0 = 500$. Assume that the underlying Markov chain has three states, M = 3. We choose the following values for probabilities of connection and deletion: state (1): p = q = 0.05, state (2): p = 0.2 and q = 0.1, and state (3): p = 0.4, q = 0.15. The sample path of the Markov chain jumps at times n = 3000 from state (1) to state (2) and n = 6000 from state (2) to state (3). As the state of the Markov chain evolves, the expected degree distribution, $\overline{g}(\theta)$, obtained by (4.9) evolves over time. The corresponding values for the expected degree distribution for nodes of degree i = 3 are displayed in Fig.4.5 using a dotted line. The estimated probability mass function, \hat{g}_n , obtained by the stochastic approximation algorithm (4.14) is plotted in Fig.4.5 using a solid line. The figure shows that the estimates using by the stochastic approximation algorithm (4.14) follow the expected degree distribution (4.9) satisfactorily even though the algorithm has no information about the underlying Markovian dynamics.

Example 4.6.4. Consider the fixed size Markov-modulated duplication-deletion random graph obtained by Procedure 4.5 with M = 91 and r = 0 and $N_0 = 1000$. For each value of $p(\theta) = 0.04 + \theta \times 0.01, \theta \in \{1, 2, ..., 91\}$ and $q \in \{0.05, 0.1, 0.15, 0.2\}$, we compute $L(\theta)$ from (4.7) and consequently the stationary distribution, $\overline{g}(\theta)$, from (4.9). As expected, the stationary distribution does not depend on q because only the deletion step in Procedure 4.5 occurs with probability q. From $\overline{g}(\theta)$, we compute the average degree of nodes, δ . Fig.4.6 shows the average degree of nodes versus the probability of the connection in Procedure 4.5. As can be seen in Fig.4.6, with increasing the probability of connection in Procedure 4.5, the average degree of nodes in the graph (which is a measure for the connectivity of the graph, see [41]). Then for each value of $p(\theta) = 0.04 + \theta \times 0.01, \theta \in \{1, 2, ..., 91\}$ and $q \in \{0.05, 0.1, 0.15, 0.2\}$, the covariance matrix is computed using (4.6). Fig.4.7 depicts the trace of the covariance matrix, trace ($\Sigma(\theta)$), for each value of p and q versus the corresponding average degree of nodes (for each value of p). As can be seen in Fig.4.7, the trace of the covariance matrix is larger when the average degree of nodes is higher (the graph is highly connected).

Recall from Lemma 4.3.1, the order of delay in the searching problem can be computed by



Figure 4.2: Degree distribution of the duplication-deletion random graph satisfies a power law. The parameters are specified in Example 4.6.1 of Section 4.6.



Figure 4.3: Degree distribution of the fixed size duplication-deletion random graph. The parameters are specified in Example 4.6.2 of Section 4.6.



Figure 4.4: Degree distribution of the fixed size duplication-deletion random graph satisfies a power law when N_0 is sufficiently large. The parameters are specified in Example 4.6.2 of Section 4.6.



Figure 4.5: The estimates obtained by SA algorithm (4.14) follows the expected PMF precisely with no knowledge of the Markovian dynamics. The parameters are specified in Example 4.6.3



Figure 4.6: The average degree of nodes (as a measure of connectivity) of the fixed size Markovmodulated duplication-deletion random graph obtained by Procedure 4.5 for different values of the probability of connection, p, in Algorithm 4.5. The parameters are specified in Example 4.6.4 of Section 4.6.

 $\lambda(N_0) = O\left(\frac{N_0\delta}{\overline{d_2}-\delta}\right)$. Knowing the degree distribution $\overline{g}(\theta)$, δ and \overline{d}_2 can be computed for each value of $p \in \{0.05, 0.06, \dots, 0.95\}$. Fig.4.8 shows the trace of the covariance matrix versus $\left(\frac{\delta}{\overline{d_2}-\delta}\right)$ as a measure of the searchability for each value of $q \in \{0.05, 0.1, 0.15, 0.2\}$. As can be seen in Fig.4.8, the trace of covariance matrix is larger when the order of delay in the search problem in (4.10) is smaller⁴².

4.7 Closing Remarks

Markov-modulated duplication-deletion random graphs are analyzed in terms of their degree distribution. When the size of graph is fixed (r = 0) and ρ is small, the expected degree distribution of the Markov-modulated duplication-deletion random graph can be computed from (4.6) for each state of the underlying Markov chain. This result allows us to express the structure of network (degree distribution) in terms of the dynamics of the model. We also showed that, the infinite duplication-deletion random graph without Markovian dynamics generated according to Procedure 4.5 (r = 1, M = 1) satisfies a power law with component computed from (4.11). The importance of this result is that a single parameter (power law component) characterizes the structure of a possibly very large dy-

⁴²This means that the target node can be found in the search problem with smaller number of steps.



Figure 4.7: Trace of the covariance matrix of scaled tracking error, trace $(\Sigma(\theta))$, versus the average degree of nodes as a measure of connectivity of the network. The parameters are specified in Example 4.6.3 of Sec.4.6.



Figure 4.8: Trace of the covariance matrix of the scaled tracking error, trace $(\Sigma(\theta))$, versus the order of delay in the searching problem as a measure of searchability of the network. The parameters are specified in Example 4.6.3 of Sec.4.6.

namic network.

Also a stochastic approximation algorithm was presented to adaptively estimate the degree distribution of random graphs. The stochastic approximation algorithm (4.14) does not assume knowledge of the Markov-modulated dynamics of the graph. Theorem 4.4.1 showed that the tracking error of the stochastic approximation algorithm is small and is in order of $O(\varepsilon)$. As a result of this bound, we showed that the scaled tracking error weakly converges to a diffusion process. Motivated by the analysis of social networks, we presented a Hilbert-space-valued stochastic approximation algorithm to estimate the expected degree distribution of the infinite duplication-deletion random graph without Markovian dynamics. The asymptotic behaviour of such an algorithm is analyzed in terms of the power law degree distribution. Finally, we extended the analysis to a Hilbert-space-valued stochastic approximation algorithm that aims to track a Markov-modulated probability mass function with denumerable support. Using weak convergence methods, it was shown that the estimates obtained via such an algorithm converge weakly to the solution of an ordinary differential equation. It was also shown that the interpolated sequence of scaled tracking error converges weakly to the solution of a stochastic differential equation.

4.8 **Proof of Results**

4.8.1 Proof of Theorem 4.3.1

The proof is based on the proof of Lemma 4.1 in [41, Chapter 4, p79]. To compute the expected degree distribution of the Markov-modulated random graph, we find a relation between the number of nodes with specific degree at time n and the degree distribution of the graph at time n - 1. Recall that the *i*-th element of f_n , f_n^i , denotes the number of vertices with degree *i* at time *k*. Given the resulting graph at time *n*, the aim is to find the expected number of nodes with degree i + 1 at time n + 1. The following events can occur that result in a node with degree i + 1 at time n + 1:

- Degree of a node with degree i increments by one in the duplication step (Step 1 of the duplication-deletion Procedure 4.5) and remains unchanged in the deletion step (Step 2):
 - A node with degree *i* is chosen at the duplication step as a parent node and remains unchanged in the deletion step. The probability of occurrence of such an event is

$$r\left(1 - \frac{q(i+1) + q(1+pi) - q(1+pi)(i+1)/N_n}{N_n}\right)\frac{f_n^i}{N_n}$$

the probability of choosing a node with degree *i* is $\frac{f_n^i}{N_n}$ and the probability of the event

that this node remains unchanged in the deletion step is⁴³

$$1 - \frac{q(i+1) + q(1+pi) - q(1+pi)(i+1)/N_n}{N_n}.$$

- One neighbor of a node with degree *i* is selected as a parent node; the parent node connects to its neighbors (including the node with degree *i*) with probability *p* in the edge-duplication part of Step 1. The probability of such an event is

$$r\frac{f_n^i pi}{N_n} \left(1 - \frac{q(i+2) + q(1+p(i+1)) - q(1+p(i+1))(i+2)/N_n}{N_n}\right)$$

Note that the node whose degree is incremented by one in this event should remain unaffected in Step 2; the probability of being unchanged in Step 2 for such a node is

$$1 - \frac{q(i+2) + q(1+p(i+1)) - q(1+p(i+1))(i+2)/N_n}{N_n}.$$

- A node with degree i + 1 remains unchanged in both Step 1 and Step 2 of Procedure 4.5:
 - Using the same argument as above, the probability of such an event is

$$f_n^{i+1}\left(1 - q\frac{i+3 + p(i+1) - \frac{(1+p(i+1))(i+2)}{N_n}}{N_n}\right)\left(1 - r\frac{p(i+1) + 1}{N_n}\right)$$

- A new node with degree i + 1 is generated in Step 1:
 - The degree of the most recently generated node (in the vertex- duplication part of Step 1) increments to i + 1; the new node connects to "*i*" neighbors of the parent node and remains unchanged in Step 2. The probability of this scenario is

$$r\left(1-q\frac{i+3+p(i+1)-\frac{(1+p(i+1))(i+2)}{N_n}}{N_n}\right)\sum_{j\geq i}\frac{f_n^j}{N_n}\binom{j}{i}p^i(1-p)^{j-i}.$$

• Degree of a node with degree i + 2 decrements by one in Step 2:

⁴³The deletion step (Step 2 of Procedure 4.5) comprises an edge-deletion step and a duplication step. The probability that the degree of node with degree *i* changes in the edge-deletion step is $\frac{q(i+1)}{N_n}$; either this node or one of its neighbors should be selected in the edge-deletion step. Also given that the degree of this node dose not change in the edge-deletion step, if either this node or one of its neighbor is selected in the duplication step (within Step 2) then the degree of this node increments by one with probability $\frac{1+pi}{N_n}$. Therefore, the probability that the degree of a node of degree *i* remains unchanged in Step 2 is

$$1 - \frac{q(i+1) + q(1+pi) - q(1+pi)(i+1)/N_n}{N_n}$$

Note that for simplicity in our analysis, it is assumed that the nodes whose degrees change in the edge-deletion part of Step 2, remain unchanged in the duplication part of Step 2 at that time instant. Also, the new node, which is generated in the vertex-duplication step of Step 1, remains unchanged in Step 2.

- A node with degree i + 2 remains unchanged in the duplication step and one of its neighbors is eliminated in the deletion step. The probability of this event is

$$q\left(\frac{i+2}{N_n}\right)\left(1-\frac{p(i+2)+1}{N_n}\right).$$

- A node with degree i + 1 is generated in Step 2:
 - The degree of the node generated in the vertex-duplication part of duplication step within Step 2 increments to i + 1. The probability of this event is

$$q\sum_{j\geq i}\frac{1}{N_n}f_n^j\binom{j}{i}p^i(1-p)^{j-i}.$$

- Degree of a node with degree i increments by one in Step 2:
 - A node with degree *i* remains unchanged in Step 1 and its degree increments by one in the duplication part of Step 2. The corresponding probability is

$$\frac{q(1+pi)}{N_n}\left(1-\frac{1+pi}{N_n}\right).$$

Let Ω denote the set of all arbitrary graphs and \mathscr{F}_n denote the sigma algebra generated by graphs $G_{\tau}, \tau \leq n$. Considering the above events that result in a node with degree i + 1 at time n + 1, the following recurrence formula can be derived for the conditional expectation of f_{n+1}^{i+1} :

$$\begin{aligned} \mathbf{E}\{f_{n+1}^{i+1}|\mathscr{F}_{n}\} &= \\ \left(1 - q\frac{i+3+p(i+1) - \frac{(1+p(i+1))(i+2)}{N_{n}}}{N_{n}}\right) \left(1 - r\frac{p(i+1)+1}{N_{n}}\right) f_{n}^{i+1} \\ &+ r\left(1 - \frac{q(i+1)+q(1+pi) - q(1+pi)(i+1)/N_{n}}{N_{n}}\right) \left(\frac{1+pi}{N_{n}}\right) f_{n}^{i} \\ &+ r\left(1 - q\frac{i+3+p(i+1) - \frac{(1+p(i+1))(i+2)}{N_{n}}}{N_{n}}\right) \sum_{j\geq i} \frac{f_{n}^{j}}{N_{n}} {j \choose i} p^{i} (1-p)^{j-i} \\ &+ q\sum_{j\geq i} \frac{f_{n}^{j}}{N_{n}} {j \choose i} p^{i} (1-p)^{j-i} + q\left(\frac{i+2}{N_{n}}\right) \left(1 - \frac{p(i+2)+1}{N_{n}}\right) f_{n}^{i+2} \\ &+ \frac{q(1+pi)}{N_{n}} \left(1 - \frac{1+pi}{N_{n}}\right) f_{n}^{i}. \end{aligned}$$
(4.31)

Let $\overline{f}_n^i = \mathbf{E}\{f_n\}$. By taking expectation of both sides of (4.31) with respect to trivial sigma algebra $\{\Omega, \emptyset\}$, the smoothing property of conditional expectations yields:

$$\begin{split} \overline{f}_{n+1}^{i+1} &= \\ \left(1 - q \frac{i+3+p(i+1) - \frac{(1+p(i+1))(i+2)}{N_n}}{N_n}\right) \left(1 - r \frac{p(i+1)+1}{N_n}\right) \overline{f}_n^{i+1} \\ &+ r \left(1 - \frac{q(i+1)+q(1+pi) - \frac{q(1+pi)(i+1)}{N_n}}{N_n}\right) \left(\frac{1+pi}{N_n}\right) \overline{f}_n^i \\ &+ r \left(1 - q \frac{i+3+p(i+1) - \frac{(1+p(i+1))(i+2)}{N_n}}{N_n}\right) \sum_{j \ge i} \frac{\overline{f}_n^j}{N_n} {j \choose i} p^i (1-p)^{j-i} \\ &+ q \sum_{j \ge i} \frac{1}{N_n} \overline{f}_n^j {j \choose i} p^i (1-p)^{j-i} + q \left(\frac{i+2}{N_n}\right) \left(1 - \frac{p(i+2)+1}{N_n}\right) \overline{f}_n^{i+2} \\ &+ \frac{q(1+pi)}{N_n} \left(1 - \frac{1+pi}{N_n}\right) \overline{f}_n^i. \end{split}$$
(4.32)

Assuming that size of the graph is sufficiently large, each term like $\frac{\overline{f}_n^i}{N_n^2}$ can be neglected. Eq. (4.32) can be written as

$$\overline{f}_{n+1}^{i+1} = \left(1 - \frac{q(i+2) + (r+q)(p(i+1)+1)}{N_n}\right) \overline{f}_n^{i+1} + \left(\frac{(1+pi)(r+q)}{N_n}\right) \overline{f}_n^i + q\left(\frac{i+2}{N_n}\right) \overline{f}_n^{i+2} + q \sum_{j \ge i} \frac{1}{N_n} \overline{f}_n^j {j \choose i} p^i (1-p)^{j-i}.$$
(4.33)

Using (4.32), we can write the following recursion for the (i+1)-th element of \overline{g}_{n+1} :

$$\overline{g}_{n+1}^{i+1} = \left(\frac{N_n - \left(q(i+2) + (r+q)\left(p(i+1) + 1\right)\right)}{N_{n+1}}\right) \overline{g}_n^{i+1} \\ + \left(\frac{(1+pi)(r+q)}{N_{n+1}}\right) \overline{g}_n^i + q\left(\frac{i+2}{N_{n+1}}\right) \overline{g}_n^{i+2} \\ + q \sum_{j \ge i} \frac{1}{N_{n+1}} \overline{g}_n^j {j \choose i} p^i (1-p)^{j-i}.$$
(4.34)

Since the probability of duplication step r = 0, the number of vertices does not increase. Thus, $N_n = N_0$ and (4.34) can be written as

$$\overline{g}_{n+1}^{i+1} = \left(1 - \frac{1}{N_0} \left(q(i+2) + q\left(p(i+1) + 1\right)\right)\right) \overline{g}_n^{i+1} \\ + \frac{1}{N_0} \left((1+pi)q\overline{g}_n^i + \frac{1}{N_0}q(i+2)\overline{g}_n^{i+2}\right) \\ + \frac{1}{N_0} q \sum_{j \ge i} \overline{g}_n^j {j \choose i} p^i (1-p)^{j-i}.$$

$$(4.35)$$

It is clear in (4.35) that the vector \overline{g}_{n+1} depends on elements of \overline{g}_n . Using matrix notation, (4.35) can be expressed as

$$\overline{g}_{n+1} = \left(I + \frac{1}{N_0}L'\right)\overline{g}_n \tag{4.36}$$

where L is defined as (4.7).

To prove that L is a generator, we need to show that $l_{ii} < 0$ and $\sum_{i=1}^{N_0} l_{ki} = 0$. Accordingly,

$$\sum_{i=1}^{N_0} l_{ki} = -\left(q(k+1) + q(1+pk)\right) + (1+pk)q + qk + q \sum_{k \le i-1} {k \choose i-1} p^{i-1} (1-p)^{k-i+1} = -q + q \sum_{k \le i-1} {k \choose i-1} p^{i-1} (1-p)^{k-i+1}.$$
(4.37)

Let m = i - 1. Then, (4.37) can be rewritten as

$$\sum_{i=1}^{N_0} l_{ik} = -q + q \sum_{m=0}^k \binom{k}{m} p^m (1-p)^{k-m}$$
$$= -q + q(1-p)^k \sum_{m=0}^k \binom{k}{m} \left(\frac{p}{1-p}\right)^m.$$
(4.38)

Knowing that $\sum_{m=0}^{k} {k \choose m} a^m = (1+a)^k$, (4.38) can be written as

$$\sum_{i=1}^{N_0} l_{ik} = -q + q(1-p)^k \left(\frac{1}{1-p}\right)^k = 0.$$
(4.39)

Also, it can be shown that $l_{ii} < 0$. Since $p^{i-1} \le 1$, $p^{i-1} < 1 + \frac{2}{i} + p + p^i$. Consequently, $iqp^{i-1}(1 - p) - q(i+2+ip) < 0$. Therefore, $l_{ii} < 0$ and the desired result follows.

4.8.2 Proof of Theorem 4.3.2

Proof. To prove Theorem 4.3.2, we first compute the power law component, β , and then we prove that the expected degree distribution converges to the power law distribution with component β . Let $\overline{f}_n(i) = \mathbf{E}\{f_n(i)\}$. Similar to (4.31), $\overline{f}_n(\theta_n, i)$ can be written as

$$\overline{f}_{n+1}(i+1) = \left(1 - q\frac{(i+2) + (1+p(i+1))}{N_n}\right) \left(1 - \frac{p(i+1)+1}{N_n}\right) \overline{f}_n(i+1) \\
+ \left(1 - q\frac{(i+1) + (1+pi)}{N_n}\right) \left(\frac{1+pi}{N_n}\right) \overline{f}_n(i) \\
+ \left(1 - q\frac{(i+2) + (1+p(i+1))}{N_n}\right) \sum_{j \ge i} \frac{1}{N_n} \overline{f}_n(j) {j \choose i} p^i (1-p)^{j-i} \\
+ q\sum_{j \ge i} \frac{1}{N_n} \overline{f}_n(j) {j \choose i} p^i (1-p)^{j-i} \\
+ q\left(\frac{i+2}{N_n}\right) \left(1 - \frac{p(i+2)+1}{N_n}\right) \overline{f}_n(i+2) \\
+ \frac{q(1+pi)}{N_n} \left(1 - \frac{1+pi}{N_n}\right) \overline{f}_n(i) \\
+ q\left(\frac{i+2}{N_n}\right) \left(\frac{p((i+1)+1)}{N_n}\right) \overline{f}_n(i+1).$$
(4.40)

To compute the power law component, we can heuristically assume that $\overline{f}_n(i) = a_i t$ as $N_n = n$ goes to infinity (we will prove this precisely later on this section). Therefore, each term like $\frac{\overline{f}_n(i')}{N_n^2}$ can be neglected as *n* approaches infinity. So (4.40) can be re-written as

$$\overline{f}_{n+1}(i+1) = \left(1 - \frac{q(i+2) + (1+q)\left(p(i+1) + 1\right)}{N_n}\right)\overline{f}_n(i+1) + \left(\frac{(1+pi)(1+q)}{N_n}\right)\overline{f}_n(i) + q\left(\frac{i+2}{N_n}\right)\overline{f}_n(i+2) + (1+q)\sum_{j\geq i}\frac{1}{N_n}\overline{f}_n(j)\binom{j}{i}p^i(1-p)^{j-i}.$$
(4.41)

Substituting $\overline{f}_{\tau}(j) = a_j \tau$ and $N_n = n$ in (4.98) yields

$$a_{i+1}(n+1) = a_{i+1}n - a_{i+1}\left(\left(1 + p(i+1)\right)(1+q) + q(i+2)\right) + (1+q)(1+pi)a_i + q(i+2)a_{i+2} + (1+q)\sum_{j\geq i}a_j\binom{j}{i}p^i(1-p)^{j-i}.$$
(4.42)

Taking all terms with a_{i+1} to the left hand side, we have

$$a_{i+1}\left(1+(1+q)\left(1+p(i+1)\right)+q(i+2)\right) = (1+q)\left((1+pi)a_i + \sum_{j\geq i}a_j\binom{j}{i}p^i(1-p)^{j-i}\right) + q(i+2)a_{i+2}.$$
(4.43)

Dividing both sides of (4.43) by a_i yields

$$\frac{a_{i+1}}{a_i} \Big(1 + (1+q) \big(1 + p(i+1) \big) + q(i+2) \big) \Big) = (1+q) \left((1+pi) + \sum_{j \ge i} \frac{a_j}{a_i} {j \choose i} p^i (1-p)^{j-i} \right) + q(i+2) \frac{a_{i+2}}{a_i}.$$
(4.44)

Solving Equation (4.43) for a_i , we can complete the proof of Theorem 4.3.2 The following lemma whose proof can be found in [42] is used to solve the recurrence relation for a_i .

Lemma 4.8.1.

$$\sum_{j\geq i} \frac{a_j}{a_i} {j \choose i} p^i (1-p)^{j-i} = p^{\beta-1} + O\left(\frac{1}{i}\right).$$
(4.45)

Proof. The proof is presented in Appendix 4.8.3.

To solve (4.43) for a_i , we can further assume that $a_i = Ci^{-\beta}$ [41]. Therefore, $\frac{a_{i+\alpha}}{a_i} = \left(\frac{i+\alpha}{i}\right)^{-\beta}$

$$\left(1 - \frac{\beta}{i}\right) \left(1 + (1+q)\left(1 + p(i+1)\right) + q(i+2)\right) = (1+q)(1+pi+p^{\beta-1}) + O\left(\frac{1}{i}\right) + q(i+2)\left(1 - \frac{2\beta}{i}\right).$$
(4.46)

Neglecting the O $\left(\frac{1}{i}\right)$ terms, yields

$$(1+q)(p^{\beta-1}+p\beta-p) = 1+\beta q.$$
(4.47)

Note that the proof presented above depends on few assumptions. To give a rigorous proof, the succeeding steps should be followed as described in [41]:

- First, we need to show that the limit $\lim_{n\to\infty} \frac{1}{n} \mathbf{E} \{f_n(i)\}$ exists.
- Let a_i be the solution of (4.43) such that $\sum_{i=1}^{\infty} a_i = 1$ and $a_0 = 0$, then it is needed to show that

$$\lim_{n \to \infty} \frac{1}{n} \mathbf{E} \{ f_n(i) \} = a_i.$$
(4.48)

• Finally, we should show that a_i is proportional to $i^{-\beta}$, where β is the root of (4.47).

To complete the proof we define new function as follows $h_n(i) = \frac{1}{n} \sum_{k=1}^{i} \mathbf{E} \{f_n(k)\}$ which can be described as CDF of degree of each node in random graph. It is sufficient to show that for all i > 0,

$$\lim_{n \to \infty} h_n(i) = \sum_{k=1}^i a_k \tag{4.49}$$

where a_i is the solution of (4.43). It is obvious if (4.49) holds, $h_n(i) - h_n(i-1) = a_i$ and thus

$$\lim_{n\to\infty}\frac{1}{n}\mathbf{E}\left\{f_n(i)\right\}=a_i$$

(as presented in (4.48)). The following lemma gives a recurrence formula to compute the value of h(n+1,i).

Lemma 4.8.2.

$$h_{n+1}(i) = D_{n+1}(i)h_n(i) + B_{n+1}(i)h_n(i-1) + C_{n+1}(i)h_n(i+1) + \frac{1+q}{n+1}\sum_{j\ge i-1}h_n(j)F(j,i-1,p),$$
(4.50)

where

$$D_{n+1}(i) = \left(\frac{n - \left(q(i+2) + (1+q)\left(pi+1\right)\right)}{n+1}\right),$$

$$B_{n+1}(i) = \frac{(1+q)(1+pi)}{n+1},$$

$$C_{n+1}(i) = \frac{q(i+1)}{n+1},$$

$$F(j,i,p) = \sum_{k=0}^{i} {j \choose k} p^{k}(1-p)^{j-k} - \sum_{k=0}^{i} {j+1 \choose k} p^{k}(1-p)^{j+1-k}.$$

This lemma can be proved by induction. The complete proof can be found in Appendix 4.8.4. The recursive equation presented in Lemma 4.8.2 is used later to prove that the degree distribution converges to a power law.

Lemma 4.8.3. *Let* $s_i = \sum_{k=1}^{i} a_i$ *and*

$$\boldsymbol{\omega}(n) = \sup_{i \ge 1} \frac{h_n(i)}{s_i},\tag{4.51}$$

where $h_n(i)$ satisfies (4.50). Then the limit $\lim_{n\to\infty} \omega(n)$ exists and we have $\lim_{n\to\infty} \omega(n) = 1$.

Sketch of the proof Knowing that $h_n(i)$ satisfies the recurrence formula (4.50), the proof is similar to [41]. Plugging i = n in (4.51) yields $\omega(n) \ge \frac{h_n(n)}{s_n} \ge \frac{1}{s_n} \ge 1$. Using the Lemma 4.8.2 and similar to [41], it can be shown that $\omega(n+1) \le \omega(n)$. $\omega(n)$ is bounded and decreasing, so the limit of $\lim_{n\to\infty} \omega(n)$ exists. To show $\lim_{n\to\infty} \omega(n) = 1$, we assume that $\lim_{n\to\infty} \omega(n) = c$. It can be shown that if $c \ne 1$, $\omega(n) \le 1$ is violated. Thus c = 1 and the proof is complete.

4.8.3 **Proof of Lemma 4.8.1**

Proof.

$$\begin{split} \sum_{j \ge i} \frac{a_j}{a_i} {j \choose i} p^i (1-p)^{j-i} &= \sum_{j \ge i} (\frac{i}{j})^{\beta} {j \choose i} p^i (1-p)^{j-i} \\ &= \sum_{j \ge i} (\frac{i}{j})^{\beta} {j \choose j-i} p^i (1-p)^{j-i} \\ &= \left(1 + O(\frac{1}{i})\right) \sum_{j \ge i} {j-\beta \choose j-i} p^i (1-p)^{j-i} \\ &= \left(1 + O(\frac{1}{i})\right) p^i \sum_{k=0} {k-i-\beta \choose k} (1-p)^k \\ &= \left(1 + O(\frac{1}{i})\right) p^i \sum_{k=0} {\beta-i-1 \choose k} (-1)^k (1-p)^k \\ &= \left(1 + O(\frac{1}{i})\right) p^i p^{\beta-i-1} = \left(1 + O(\frac{1}{i})\right) p^{\beta-1}. \end{split}$$
(4.52)

4.8.4 **Proof of Lemma 4.8.2**

Proof. We prove the lemma by induction on *i*:

For i = 1 It is sufficient to show that: $h(n+1,1) = D_{n+1}(1)h(n,1) + C_{n+1}(1)h(n,2) + \frac{1}{n+1}\sum_{j\geq 1}h(n,j)F(j,0,p)$. Also using the definition of F(j,i,p), we can rewrite F(j,0,p) as $(1-p)^j - (1-p)^{j+1}$. The number of nodes with degree one at time n + 1 can be written as following

$$\mathbf{E}\{f(n+1,1)\} = \left(1 - \frac{(1+q)(1+p)+q}{n}\right) \mathbf{E}\{f_n(1)\} + \frac{2q}{n} \mathbf{E}\{f_n(2)\} + (1+q) \sum_{j \ge 1} \frac{1}{n} \mathbf{E}\{f_n(j)\}(1-p)^j.$$
(4.53)

Note that (4.53) is slightly different from the general equation for each *i*, (4.98). Because as described in Section 4.1, neighbors of a node with degree one cannot be eliminated from the graph to maintain the connectivity in the graph. Therefore, a node with degree one can change in the deletion step if that node is selected in the deletion step (with probability *q*). Using (4.53), h(n+1,1) can be written as

$$h(n+1,1) = \frac{1}{n+1} \mathbf{E} \{ f(n+1,1) \}$$

= $\frac{1}{n+1} \left(\left(1 - \frac{(1+q)(1+p)+q}{n} \right) \mathbf{E} \{ f_n(1) \} + \frac{2q}{n} \mathbf{E} \{ f_n(2) \} \right)$
+ $\frac{1}{n+1} \sum_{j \ge 1} \frac{1+q}{n} \mathbf{E} \{ f_n(j) \} (1-p)^j.$ (4.54)

We know that h(n,0) = 0 for all *n*. Using the definition of $h(\cdot, \cdot)$ and (4.53), (4.54) can be re-arranged as follows

$$h(n+1,1) = \frac{1}{n+1} \left(\left(n - \left((1+q)(1+p) + q \right) \right) h(n,1) + \frac{2q}{n} \left(h(n,2) - h(n,1) \right) + (1+q) \sum_{j \ge 1} (h(n,j) - h(n,j-1))(1-p)^j \right) \\ = \frac{1}{n+1} \left(\left(n - \left(3q + (1+q)(1+p) \right) \right) h(n,1) + \frac{2q}{n} h(n,2) \right) + \frac{1+q}{n+1} \sum_{j \ge 1} (h(n,j) - h(n,j-1))(1-p)^j$$

$$(4.55)$$

 $\sum_{j\geq 1}(h(n,j)-h(n,j-1))(1-p)^j$ can be written in terms of the F(j,i,p).

$$\begin{split} \sum_{j\geq 1} (h(n,j) - h(n,j-1))(1-p)^j &= \sum_{j\geq 1} h(n,j)(1-p)^j - \sum_{j\geq 1} (h(n,j-1)(1-p)^j \\ &= \sum_{j\geq 1} h(n,j)(1-p)^j - \sum_{j\geq 1} (h(n,j)(1-p)^j + 1 \\ &= \sum_{j\geq 1} h(n,j) \left((1-p)^j - (1-p)^{j+1} \right) \\ &= \sum_{j\geq 1} h(n,j) F(j,0,p). \end{split}$$
(4.56)

Substituting (4.56) in (4.55) yields

$$h(n+1,1) = \frac{1}{n+1} \left(\left(n - \left((1+q)(1+p) + 3q \right) \right) h(n,1) + \frac{2q}{n} h(n,2) + (1+q) \sum_{j \ge 1} h(n,j) F(j,0,p) \right) \\ = D_{n+1}(1)h(n,1) + C_{n+1}(1)h(n,2) + \frac{1+q}{n+1} \sum_{j \ge 1} h(n,j) F(j,0,p).$$
(4.57)

Thus (4.50) holds for i = 1. Now it is assumed that (4.50) holds for i = k, we want to show that it also holds for i = k + 1.

$$\mathbf{E}\{f(n+1,k+1)\} = \left(1 - \frac{q(k+2) + (1+q)(p(k+1)+1)}{n}\right) \mathbf{E}\{f(n,k+1)\} \\ + \left(\frac{(1+q)(1+pk)}{n}\right) \mathbf{E}\{f_n(k)\} + \left(\frac{q(k+2)}{n}\right) \mathbf{E}\{f_n(k+2)\} \\ + (1+q)\sum_{j\leq k} \frac{f_n(j)}{n} \binom{j}{k} p^k (1-p)^{j-k}.$$
(4.58)

from definition of h(n,k), we have : $\mathbf{E}\{f_n(k)\} = n(h(n,k) - h(n,k-1))$. Eq. (4.58) can be rewritten as follows

$$\mathbf{E}\{f(n+1,k+1)\} = \left(n - \left(q(k+2) + (1+q)\left(p(k+1) + 1\right)\right)\right) \left(h(n,k+1) - h(n,k)\right) \\ + (1+q)(1+pk)\left(h(n,k) - h(n,k-1)\right) + q(k+2)\left(h(n,k+2) - h(n,k+1)\right) \\ + (1+q)\sum_{j \le k} \left(h(n,j) - h(n,j-1)\right) \binom{j}{k} p^k (1-p)^{j-k}.$$

$$(4.59)$$

Using the Abel summation identity, and knowing that

$$F(j,k,p) = \sum_{k=0}^{k} {j \choose k} p^{k} (1-p)^{j-k} - \sum_{k=0}^{k} {j+1 \choose k} p^{k} (1-p)^{j+1-k},$$

the last term can be written as

$$\sum_{j \le k} \left(h(n,j) - h(n,j-1) \right) {j \choose k} p^k (1-p)^{j-k}$$

$$= \sum_{j \ge k} \left({j \choose k} p^k (1-p)^{j-k} - {j+1 \choose k} p^k (1-p)^{j+1-k} \right) - p^k h(n,k-1)$$

$$= -p^k h(n,k-1) + \sum_{j \ge k} h(n,j) \left(F(j,k,p) - F(j,k-1,p) \right).$$
(4.60)
(4.61)

Substituting (4.60) in (4.59) yields

$$\mathbf{E}\{f(n+1,k+1)\} = h(n,k+2)(q(k+2)) + h(n,k+1)\left(n - \left(2q(k+2) + (1+q)(p(k+1)+1)\right)\right) \\ + h(n,k)\left((1+q)\left(2+p(2k+1)\right) + q(k+2) - n\right) + h(n,k-1)(1+q)(-1-pk-p^k) \\ + (1+q)\sum_{j\geq k} h(n,j)\left(F(j,k,p) - F(j,k-1,p)\right).$$

$$(4.62)$$

The value of h(n+1,k+1) can be computed using h(n,k+1) and $\mathbf{E}{f_n(k+1)}$ as follows

$$h(n+1,k+1) = h(n+1,k) + \frac{1}{n+1} \mathbf{E} \{ f(n+1,k+1) \}.$$
(4.63)

Eq.(4.62) gives an expression for $\mathbf{E}\{f(n+1,k+1)\}$ in terms of the value of $h(\cdot,\cdot)$ at time *n*. Substituting (4.62) in (4.63) gives a recursive equation for computing h(n+1,k+1):

$$h(n+1,k+1) = h(n+1,k) + \frac{1}{n+1} \mathbf{E} \{ f(n+1,k+1) \}$$

= $D_{n+1}(k)h(n,k) + B_{n+1}(k)h(n,k-1) + C_{n+1}h(n,k+1)$
+ $\frac{1+q}{n+1} \sum_{j \ge k-1} h(n,j)F(j,k-1,p)$
+ $\frac{1}{n+1} \left(h(n,k+2)(q(k+2)) + h(n,k+1) \right)$
 $\left(n - \left(2q(k+2) + (1+q)(p(k+1)+1) \right) \right)$
+ $h(n,k) \left((1+q) \left(2 + p(2k+1) \right) + h(n,k-1)(1+q)(-1-pk-p^k) \right)$
+ $(1+q) \sum_{j \ge k} h(n,j) \left(F(j,k,p) - F(j,k-1,p) \right) \right).$ (4.64)

We assume that (4.98) holds for i = k so substituting the values for $D_{n+1}(k)$, $B_{n+1}(k)$, and $C_{n+1}(k)$ from (4.98) in (4.64) yields

$$h(n+1,k+1) = h(n,k+2) \left(\frac{q(k+2)}{n+1}\right) + h(n,k+1) \left(\frac{n - \left(q(k+3) + (1+q)\left(p(k+1) + 1\right)\right)}{n+1}\right) + h(n,k) \left(\frac{(1+q)\left(1+p(k+1)\right)}{n+1}\right) + \frac{1+q}{n+1} \sum_{j \ge k} h(n,j) \left(F(j,k,p)\right).$$
(4.65)

(4.65)can be written as follows

$$h(n+1,k+1) = D_{n+1}(k+1)h(n,k+1) + B_{n+1}(k+1)h(n,k) + C_{n+1}(k+1)h(n,k+2) + \frac{1+q}{n+1} \sum_{j \ge k} h(n,j)F(j,k,p).$$
(4.66)

Thus, (4.98) holds for i = k + 1 and the proof is completed by induction.

4.8.5 **Proof of Theorem 4.4.1**

Proof. Define the Lyapunov function V(x) = (x'x)/2 for $x \in \mathbb{R}^{N_0}$. Use \mathbf{E}_n to denote the conditional expectation with respect to the σ -algebra \mathscr{H}_n generated by $\{z_j(\theta_j), \theta_j, j \leq n\}$. Then,

$$\mathbf{E}_{n}\{V(\widetilde{g}_{n+1}) - V(\widetilde{g}_{n})\} = \mathbf{E}_{n}\left\{\widetilde{g}_{n}'[-\varepsilon\widetilde{g}_{n} + \varepsilon(z_{n}(\theta_{n}) - \overline{g}(\theta_{n})) + \overline{g}(\theta_{n}) - \overline{g}(\theta_{n+1})]\right\} + \mathbf{E}_{n}\left\{\|-\varepsilon\widetilde{g}_{n} + \varepsilon(z_{n}(\theta_{n}) - \overline{g}(\theta_{n})) + \overline{g}(\theta_{n}) - \overline{g}(\theta_{n+1})\|^{2}\right\}$$
(4.67)

where $z_n(\theta_n)$ and $\overline{g}(\theta_n)$ are vectors in \mathbb{R}^{N_0} with elements $z_n^i(\theta_n)$ and $\overline{g}(\theta_n)^i$, $1 \le i \le N_0$, respectively. Due to the Markovian assumption and the structure of the transition matrix of θ_n , defined in (4.2),

$$\mathbf{E}_{n}\{\overline{g}(\theta_{n}) - \overline{g}(\theta_{n+1})\} = \mathbf{E}\{\overline{g}(\theta_{n}) - \overline{g}(\theta_{n+1})|\theta_{n}\} = \sum_{i=1}^{M} \mathbf{E}\{\overline{g}(i) - \overline{g}(\theta_{n+1})|\theta_{n} = i\}I\{\theta_{n} = i\}$$
$$= \sum_{i=1}^{M} \left[\overline{g}(i) - \sum_{j=1}^{M} \overline{g}(j)A_{ij}^{\rho}\right]I\{\theta_{n} = i\} = -\rho \sum_{i=1}^{M} \sum_{j=1}^{M} \overline{g}(j)q_{ij}I\{\theta_{n} = i\} = O(\rho)$$
(4.68)

where $I\{\cdot\}$ denotes the indicator function. Similarly, it is easily seen that

$$\mathbf{E}_{n}\{\|\overline{g}(\theta_{n}) - \overline{g}(\theta_{n+1})\|^{2}\} = O(\rho).$$
(4.69)

Using *K* to denote a generic positive value (with the notation KK = K and K + K = K), a familiar inequality $ab \le \frac{a^2+b^2}{2}$ yields

$$O(\varepsilon \rho) = O(\varepsilon^2 + \rho^2). \tag{4.70}$$

Moreover, we have $\|\widetilde{g}_n\| = \|\widetilde{g}_n\| \cdot 1 \le (\|\widetilde{g}_n\|^2 + 1)/2$. Thus,

$$O(\rho)\|\widetilde{g}_n\| \le O(\rho) \left(V(\widetilde{g}_n) + 1 \right). \tag{4.71}$$

Then, detailed estimates lead to

$$\mathbf{E}_{n}\left\{\left\|-\varepsilon\widetilde{g}_{n}+\varepsilon\left(z_{n}(\theta_{n})-\overline{g}(\theta_{n})\right)+\overline{g}(\theta_{n})-\overline{g}(\theta_{n+1})\right\|^{2}\right\} \\ \leq K\mathbf{E}_{n}\left\{\varepsilon^{2}\|\widetilde{g}_{n}\|^{2}+\varepsilon^{2}\|(z_{n}(\theta_{n})-\overline{g}(\theta_{n})\|^{2}+\varepsilon^{2}\|\widetilde{g}_{n}'(z_{n}(\theta_{n})-\overline{g}(\theta_{n+1}))\| \\ +\varepsilon\|\widetilde{g}_{n}'(\overline{g}(\theta_{n})-\overline{g}(\theta_{n+1}))\|+\varepsilon\|(z_{n}(\theta_{n})-\overline{g}(\theta_{n}))'(\overline{g}(\theta_{n})-\overline{g}(\theta_{n+1}))\|\right\} \\ +\mathbf{E}_{n}\left\{\|\overline{g}(\theta_{n})-\overline{g}(\theta_{n+1})\|\right\}^{2}.$$
(4.72)

It follows that

$$\mathbf{E}_{n}\left\{\left\|-\varepsilon\widetilde{g}_{n}+\varepsilon\left(z_{n}(\theta_{n})-\overline{g}(\theta_{n})\right)+\overline{g}(\theta_{n})-\overline{g}(\theta_{n+1})\right\|^{2}\right\}=O(\varepsilon^{2}+\rho^{2})(V(\widetilde{g}_{n})+1).$$
(4.73)

Furthermore,

$$\mathbf{E}_{n}\{V(\widetilde{g}_{n+1}) - V(\widetilde{g}_{n})\} = -2\varepsilon V(\widetilde{g}_{n}) + \varepsilon \mathbf{E}_{n}\{\widetilde{g}'_{n}[z_{n}(\theta_{n}) - \overline{g}(\theta_{n})]\} + \mathbf{E}_{n}\{\widetilde{g}'_{n}[\overline{g}(\theta_{n+1}) - \overline{g}(\theta_{n})]\} + O(\varepsilon^{2} + \rho^{2})(V(\widetilde{g}_{n}) + 1).$$

$$(4.74)$$

To obtain the desired bound, define V_1^{ρ} and V_2^{ρ} as follows:

$$V_{1}^{\rho}(\widetilde{g},n) = \varepsilon \sum_{j=n}^{\infty} \widetilde{g}' \mathbf{E}_{n} \{ z_{j}(\theta_{j}) - \overline{g}(\theta_{j}) \},$$

$$V_{2}^{\rho}(\widetilde{g},n) = \sum_{j=n}^{\infty} \widetilde{g}' \mathbf{E}_{n} \{ \overline{g}(\theta_{j}) - \overline{g}(\theta_{j+1}) \}.$$
 (4.75)

It can be shown that

$$|V_1^{\rho}(\widetilde{g},n)| = O(\varepsilon)(V(\widetilde{g})+1),$$

$$|V_2^{\rho}(\widetilde{g},n)| = O(\rho)(V(\widetilde{g})+1).$$
(4.76)

Define $W(\tilde{g}, n)$ as

$$W(\widetilde{g},n) = V(\widetilde{g}) + V_1^{\rho}(\widetilde{g},n) + V_2^{\rho}(\widetilde{g},n).$$
(4.77)

This leads to

$$\mathbf{E}_{n}\{W(\widetilde{g}_{n+1}, n+1) - W(\widetilde{g}_{n}, n)\} = \mathbf{E}_{n}\{V_{1}^{\rho}(\widetilde{g}_{n+1}, n+1) - V_{1}^{\rho}(\widetilde{g}_{n}, n)\} + \mathbf{E}_{n}\{V(\widetilde{g}_{n+1}) - V(\widetilde{g}_{n})\} + \mathbf{E}_{n}\{V_{2}^{\rho}(\widetilde{g}_{n+1}, n+1) - V_{2}^{\rho}(\widetilde{g}_{n}, n)\}.$$
(4.78)

Moreover,

$$\mathbf{E}_n\{W(\widetilde{g}_{n+1},n+1)-W(\widetilde{g}_n,n)\}=-2\varepsilon V(\widetilde{g}_n)+O(\varepsilon^2+\rho^2)(V(\widetilde{g}_n)+1).$$
(4.79)

Equation (4.79) can be rewritten as

$$\mathbf{E}_{n}\{W(\widetilde{g}_{n+1},n+1)-W(\widetilde{g}_{n},n)\} \le O(\varepsilon^{2}+\rho^{2})(W(\widetilde{g}_{n},n)+1)-2\varepsilon W(\widetilde{g}_{n},n).$$
(4.80)

If ε and ρ are chosen small enough, then there exists a small λ such that $-2\varepsilon + O(\rho^2) + O(\varepsilon^2) \le -\lambda\varepsilon$. Therefore, (4.80) can be rearranged as

$$\mathbf{E}_{n}\{W(\widetilde{g}_{n+1}, n+1)\} \leq (1-\lambda\varepsilon)W(\widetilde{g}_{n}, n) + O(\varepsilon^{2}+\rho^{2}).$$
(4.81)

Taking expectation of both sides yields

$$\mathbf{E}\{W(\widetilde{g}_{n+1}, n+1)\} \le (1-\lambda\varepsilon)\mathbf{E}\{W(\widetilde{g}_n, n)\} + O(\varepsilon^2 + \rho^2).$$
(4.82)

Iterating on (4.82) then results

$$\mathbf{E}\{W(\widetilde{g}_{n+1},n+1)\} \le (1-\lambda\varepsilon)^{n-N_{\rho}} \mathbf{E}\{W(\widetilde{g}_{N_{\rho}},N_{\rho})\} + \sum_{j=N_{\rho}}^{n} O(\varepsilon^{2}+\rho^{2})(1-\lambda\varepsilon)^{j-N_{\rho}}.$$
 (4.83)

As the result,

$$\mathbf{E}\{W(\widetilde{g}_{n+1}, n+1)\} \le (1-\lambda\varepsilon)^{n-N_{\rho}} \mathbf{E}\{W(\widetilde{g}_{N_{\rho}}, N_{\rho})\} + O\left(\varepsilon + \rho^{2}/\varepsilon\right).$$
(4.84)

If *n* is large enough, one can approximate $(1 - \lambda \varepsilon)^{n-N_{\rho}} = O(\varepsilon)$. Therefore,

$$\mathbf{E}\{W(\widetilde{g}_{n+1}, n+1)\} \le O\left(\varepsilon + \frac{\rho^2}{\varepsilon}\right)$$
(4.85)

Finally, using (4.76) and replacing $W(\tilde{g}_{n+1}, n+1)$ with $V(\tilde{g}_{n+1})$, we obtain

$$\mathbf{E}\{V(\widetilde{g}_{n+1})\} \le O\left(\rho + \varepsilon + \frac{\rho^2}{\varepsilon}\right).$$
(4.86)

4.8.6 Sketch of the Proof of Theorem 4.4.2

Proof. Since the proof is similar to [154, Theorem 4.5], we only indicate the main steps in what follows and omit most of the verbatim details.

Step 1: First, we show that the two component process $(\hat{g}^{\varepsilon}(\cdot), \theta^{\varepsilon}(\cdot))$ is tight in $D([0,T] : \mathbb{R}^{N_0} \times M)$. Using techniques similar to [156, Theorem 4.3], it can be shown that $\theta^{\varepsilon}(\cdot)$ converges weakly to a continuous-time Markov chain generated by Q. Thus, we mainly need to consider $\hat{g}^{\varepsilon}(\cdot)$. We show that

$$\lim_{\Delta \to 0} \limsup_{\varepsilon \to 0} \mathbf{E} \left[\sup_{0 \le s \le \Delta} \mathbf{E}_t^{\varepsilon} \| \widehat{g}^{\varepsilon}(t+s) - \widehat{g}^{\varepsilon}(t) \|^2 \right] = 0$$
(4.87)

where $\mathbf{E}_t^{\varepsilon}$ denotes the conditioning on the past information up to *t*. Then, the tightness follows from the criterion [97, p. 47].

Step 2: Since $(\hat{g}^{\varepsilon}(\cdot), \theta^{\varepsilon}(\cdot))$ is tight, we can extract weakly convergent subsequence according to the Prohorov theorem; see [99]. To figure out the limit, we show that $(\hat{g}^{\varepsilon}(\cdot), \theta^{\varepsilon}(\cdot))$ is a solution of the martingale problem with operator L_0 . For each $i \in \mathcal{M}$ and continuously differential function with compact support $f(\cdot, i)$, the operator is given by

$$L_0 f(\widehat{g}, i) = \nabla f'(\widehat{g}, i) [-\widehat{g} + \overline{g}(i)] + \sum_{j \in \mathscr{M}} q_{ij} f(\widehat{g}, j), \ i \in \mathscr{M}.$$

$$(4.88)$$

We can further demonstrate the martingale problem with operator L_0 has a unique solution (in the sense of in distribution). Thus, the desired convergence property follows.

4.8.7 Sketch of the Proof of Theorem 4.4.3

Proof. The proof comprises of four steps as described below:

Step 1: First, note

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \varepsilon \mathbf{v}_n + \sqrt{\varepsilon} (y_{n+1} - \mathbf{E}\overline{g}(\boldsymbol{\theta}_n)) + \frac{\mathbf{E}[\overline{g}(\boldsymbol{\theta}_n) - \overline{g}(\boldsymbol{\theta}_{n+1}]]}{\sqrt{\varepsilon}}.$$
(4.89)

The approach is similar to that of [154, Theorem 5.6]. Therefore, we will be brief.

Step 2: Define an operator

$$\mathscr{L}f(\mathbf{v},i) = -\nabla f'(\mathbf{v},i)\mathbf{v} + \frac{1}{2}\operatorname{tr}[\nabla^2 f(\mathbf{v},i)\Sigma(i)] + \sum_{j\in\mathscr{M}} q_{ij}f(\mathbf{v},j), \ i\in\mathscr{M},$$
(4.90)

for function $f(\cdot, i)$ with compact support that has continuous partial derivatives with respect to v up to the second order. It can be shown that the associated martingale problem has a unique solution (in the sense of in distribution).

Step 3: It is natural now to work with a truncated process. For a fixed, but otherwise arbitrary $r_1 > 0$, define a truncation function

$$q^{r_1}(x) = \begin{cases} 1, & \text{if } x \in S^{r_1}, \\ 0, & \text{if } x \in \mathbb{R}^{N_0} - S^{r_1} \end{cases}$$

where $S^{r_1} = \{x \in \mathbb{R}^{N_0} : ||x|| \le r_1\}$. Then, we obtain the truncated iterates

$$\mathbf{v}_{n+1}^{r_1} = \mathbf{v}_n^{r_1} - \varepsilon \mathbf{v}_n^{r_1} q^{r_1}(\mathbf{v}_n^{r_1}) + \sqrt{\varepsilon} (\mathbf{y}_{n+1} - \mathbf{E}\overline{g}(\theta_n)) + \frac{\mathbf{E}[\overline{g}(\theta_n) - \overline{g}(\theta_{n+1})]}{\sqrt{\varepsilon}} q^{r_1}(\mathbf{v}_n^{r_1}).$$
(4.91)

Define $v^{\varepsilon,r_1}(t) = v_n^{r_1}$ for $t \in [\varepsilon n, \varepsilon n + \varepsilon)$. Then, $v^{\varepsilon,r_1}(\cdot)$ is an *r*-truncation of $v^{\varepsilon}(\cdot)$; see [99, p. 284] for a definition. We then show the truncated process $(v^{\varepsilon,r_1}(\cdot), \theta^{\varepsilon}(\cdot))$ is tight. Moreover, by Prohorov's theorem, we can extract a convergent subsequence with limit $(v^{r_1}(\cdot), \theta(\cdot))$ such that the limit $(v^{r_1}(\cdot), \theta(\cdot))$ is the solution of the martingale problem with operator \mathscr{L}^{r_1} defined by

$$\mathscr{L}^{r_1} f^{r_1}(\mathbf{v}, i) = -\nabla \ell f^{r_1}(\mathbf{v}, i) \mathbf{v} + \frac{1}{2} \operatorname{tr}[\nabla^2 f^{r_1}(\mathbf{v}, i) \Sigma(i)] + \sum_{j \in \mathscr{M}} q_{ij} f^{r_1}(\mathbf{v}, j)$$
(4.92)

for $i \in \mathcal{M}$, where $f^{r_1}(\mathbf{v}, i) = f(\mathbf{v}, i)q^{r_1}(\mathbf{v})$.

Step 4: Letting $r_1 \to \infty$, we show that the un-truncated process also converges and the limit, denoted by $(\mathbf{v}(\cdot), \theta(\cdot))$, is precisely the martingale problem with operator \mathscr{L} defined in (4.90). The limit covariance can further be evaluated as in [154, Lemma 5.2].

4.8.8 Proof of Theorem 4.5.2

Proof. The proof of the theorem is divided into several steps and uses techniques in stochastic approximation [99] but with the modification that ℓ_2 is a Hilbert space (see [61, 98]). Whenever possible, we only indicate the main idea and refer to the literature of stochastic approximation.

Step 0: Note that (4.24) has a unique solution for each initial condition since it is linear in $\hat{g}(\cdot)$.

Step 1: Preliminary estimates. From (4.23), we obtain that for $0 < \varepsilon < 1$, the elements of \hat{g}_n are non-negative and add up to one. Thus, \hat{g}_n is bounded.

In addition, define $V(\hat{g}) = \frac{1}{2} \langle \hat{g} - \overline{g}, \hat{g} - \overline{g} \rangle$, which can be thought of as a Lyapunov function. Then using perturbed Lyapunov function argument [99], it can be shown

$$\mathbf{E}V(\widehat{g}_n) = O(\varepsilon). \tag{4.93}$$

Step 2: Tightness of $\{\widehat{g}^{\varepsilon}(\cdot)\}$. Henceforth, we often use t/ε and $(t+s)/\varepsilon$ to denote $\lfloor t/\varepsilon \rfloor$ and $\lfloor (t+s)/\varepsilon \rfloor$, the integer parts of t/ε and $(t+s)/\varepsilon$, respectively. By using the boundedness of $\{\widehat{g}_n\}$ established in the first step together with the Hölder inequality, we have for each $0 < T < \infty$, any

 $t \ge 0$, any $0 < \delta$, any $0 < s \le \delta$, and $\varepsilon > 0$,

$$\begin{split} \mathbf{E}_{t}^{\varepsilon} \|\widehat{g}^{\varepsilon}(t+s) - \widehat{g}^{\varepsilon}(t)\|^{2} \\ &\leq \mathbf{E}_{t}^{\varepsilon} \left| \varepsilon \sum_{\substack{j=t/\varepsilon \\ \varepsilon}}^{(t+s)/\varepsilon - 1} [z_{j} - \widehat{g}_{j}] \right|^{2} \\ &\leq K\varepsilon \left(\frac{t+s}{\varepsilon} - \frac{t}{\varepsilon} \right) \leq Ks, \end{split}$$

where K > 0 is independent of ε and $\mathbf{E}_t^{\varepsilon}$ denotes the conditional expectation with respect to $\mathscr{F}_t^{\varepsilon}$. Thus

$$\mathbf{E}_{t}^{\varepsilon} \|\widehat{g}^{\varepsilon}(t+s) - \widehat{g}^{\varepsilon}(t)\|^{2} \leq Ks, \\ \lim_{\delta \to 0} \limsup_{\varepsilon \to 0} \mathbf{E}[\sup_{0 < s \leq \delta} \mathbf{E}_{t}^{\varepsilon} \|\widehat{g}^{\varepsilon}(t+s) - \widehat{g}^{\varepsilon}(t)\|^{2}] = 0.$$

$$(4.94)$$

The tightness criterion (see [97, Theorem 3, p. 47] with \mathbb{R}^r replaced by ℓ_2 ; see also [61]) enables us to conclude that $\{\widehat{g}^{\varepsilon}(\cdot)\}$ is tight in $D([0,\infty):\ell_2)$.

Step 3: Characterization of the limit process. Since $\{\hat{g}^{\varepsilon}(\cdot)\}\$ is tight, by Prohorov's theorem, we can extract a convergent subsequence. Select such a sequence and still denote it by $\hat{g}^{\varepsilon}(\cdot)$ with limit denoted by $\hat{g}(\cdot)$. By using the Skorohod representation, with a slight abuse of notation, we may assume that $\hat{g}^{\varepsilon}(\cdot)$ converges to $\hat{g}(\cdot)$ w.p.1 and the convergence is uniform on any bounded time interval. We shall show that $\hat{g}(\cdot)$ is a solution of the martingale problem with operator

$$\mathscr{L}f(\widehat{g}) = \left\langle \nabla f(\widehat{g}), [\overline{g} - \widehat{g}] \right\rangle$$

for any $f(\cdot) \in C_0^1(\ell_2 : \mathbb{R})$ (collection of real-valued C^1 functions defined on ℓ_2 with compact support). We need to show that

$$f(\widehat{g}(t)) - f(\widehat{g}(0)) - \int_0^t \mathscr{L}f(\widehat{g}(\tau))d\tau$$
 is a marginale.

To prove the martingale property, we pick out any bounded and continuous function $h(\cdot)$ defined on ℓ_2 , any $T < \infty$, any $0 < t, s \le T$, any positive integer κ , and $t_{l_1} \le t$ for any $l \le \kappa$. To derive the desired property, we need only show that

$$\mathbf{E}h(\widehat{g}(t_{l_1}): l_1 \le \kappa) \left(f(\widehat{g}(t+s)) - f(\widehat{g}(t)) - \int_t^{t+s} \mathscr{L}f(\widehat{g}(\tau)) d\tau \right) = 0.$$
(4.95)

To prove (4.95), we work with the process indexed by ε . First, by the weak convergence and the Skorohod representation,

$$\lim_{\varepsilon \to 0} \mathbf{E}h(\widehat{g}^{\varepsilon}(t_{l_1}) : l_1 \le \kappa) [f(\widehat{g}^{\varepsilon}(t+s)) - f(\widehat{g}^{\varepsilon}(t))] = \mathbf{E}h(\widehat{g}(t_{l_1}) : l_1 \le \kappa) [f(\widehat{g}(t+s)) - f(\widehat{g}(t))].$$
(4.96)

Choose a sequence of integers $\{m_{\varepsilon}\}$ such that $m_{\varepsilon} \to \infty$ as $\varepsilon \to 0$ but $\Delta_{\varepsilon} = \varepsilon m_{\varepsilon} \to 0$. Next, we note

$$\begin{split} f(\widehat{g}^{\varepsilon}(t+s)) - f(\widehat{g}^{\varepsilon}(t)) &= f(\widehat{g}_{(t+s)/\varepsilon}) - f(\widehat{g}_{t/\varepsilon}) \\ &= \sum_{\substack{(t+s)/\varepsilon-1\\lm_{\varepsilon}=t/\varepsilon}}^{(t+s)/\varepsilon-1} [f(\widehat{g}_{lm_{\varepsilon}+m_{\varepsilon}}) - f(\widehat{g}_{lm_{\varepsilon}})] \\ &= \varepsilon \sum_{\substack{(t+s)/\varepsilon-1\\lm_{\varepsilon}=t/\varepsilon}}^{(t+s)/\varepsilon-1} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} [z_{j} - \widehat{g}_{j}] \rangle + o(1) \\ &= \sum_{\substack{(t+s)/\varepsilon-1\\lm_{\varepsilon}=t/\varepsilon}}^{(t+s)/\varepsilon-1} \Delta_{\varepsilon} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \frac{1}{m_{\varepsilon}} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} [z_{j} - \widehat{g}_{j}] \rangle + o(1), \end{split}$$

where $o(1) \rightarrow 0$ in probability as $\varepsilon \rightarrow 0$. The stationarity and the mixing condition imply that

$$\frac{1}{m_{\varepsilon}}\sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1}\mathbf{E}_{lm_{\varepsilon}}z_{j} \to \mathbf{E}z_{0} = \sum_{i=0}^{\infty}\mathbf{e}_{i}P(y_{0}=i) = \overline{g} \text{ in probability.}$$

Therefore,

$$\mathbf{E}h(\widehat{g}^{\varepsilon}(t_{l_{1}}):t_{1} \leq \kappa) \left[\sum_{lm_{\varepsilon}=t/\varepsilon}^{(t+s)/\varepsilon-1} \Delta_{\varepsilon} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \frac{1}{m_{\varepsilon}} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} z_{j} \rangle \right] \\
= \mathbf{E}h(\widehat{g}^{\varepsilon}(t_{l_{1}}):t_{1} \leq \kappa) \left[\sum_{lm_{\varepsilon}=t/\varepsilon}^{(t+s)/\varepsilon-1} \Delta_{\varepsilon} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \frac{1}{m_{\varepsilon}} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} \mathbf{E}_{lm_{\varepsilon}} z_{j} \rangle \right] \\
\rightarrow \mathbf{E}h(\widehat{g}(t_{l_{1}}):t_{1} \leq \kappa) \left(\int_{t}^{t+s} \langle \nabla f(\widehat{g}(\tau)), \overline{g} \rangle d\tau \right) \text{ as } \varepsilon \to 0.$$
(4.97)

Likewise,

$$\mathbf{E}h(\widehat{g}^{\varepsilon}(t_{l_{1}}):t_{1}\leq\kappa)\left[-\sum_{lm_{\varepsilon}=t/\varepsilon}^{(t+s)/\varepsilon-1}\Delta_{\varepsilon}\langle\nabla f(\widehat{g}_{lm_{\varepsilon}}),\frac{1}{m_{\varepsilon}}\sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1}\widehat{g}_{j}\rangle\right] \\ \rightarrow \mathbf{E}h(\widehat{g}(t_{l_{1}}):t_{1}\leq\kappa)\left[-\int_{t}^{t+s}\langle\nabla f(\widehat{g}(\tau)),\widehat{g}(\tau)\rangle d\tau\right].$$

$$(4.98)$$

Combing (4.96)–(4.98), (4.95) follows.

4.8.9 Proof of Theorem 4.5.3

Proof. In the proof of Theorem 4.5.3, we use several lemmas and propositions described below. From (4.23),

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \varepsilon \mathbf{v}_n + \sqrt{\varepsilon}(z_n - \overline{g}). \tag{4.99}$$

Lemma 4.8.1. Under assumption Theorem 4.5.2, for sufficiently small ε , there is an N_{ε} such that $EV(v_n) = O(1)$ for all $n \ge N_{\varepsilon}$.

Proof The proof uses a perturbed Lyapunov function argument.

To proceed, recall the definition of covariance operator and Wiener process [46, 98] on ℓ_2 . A covariance Γ of an ℓ_2 -valued random variable *y* is an operator from ℓ_2 to ℓ_2 defined by $\Gamma v = EY \langle v, y \rangle$ for any $v \in \ell_2$. A process $W(\cdot)$ is a zero mean (stationary increment) ℓ_2 -valued Wiener process if there are mutually independent real-valued, zero mean, Wiener processes $\{W_i(\cdot)\}$ with covariances $t\rho_i$ satisfying $\sum_{i=0}^{\infty} \rho_i < \infty$ and there is an orthonormal sequence $\{\beta_i\}$ with $\beta_i \in \ell_2$ such that $W(t) = \sum_{i=0}^{\infty} W_i(t)\beta_i$. For $v, z \in \ell_2$, the covariance operator of W(t) is defined by

$$\mathbf{E}\langle W(t), v \rangle \langle W(t), z \rangle = t \langle z, \Gamma v \rangle = t \sum_{i=0}^{\infty} \rho_i \langle \beta_i, v \rangle \langle \beta_i, z \rangle.$$
(4.100)

Lemma 4.8.2. Assume the conditions of Theorem 4.5.2. For any natural number $i \in \mathbb{N}$, define

$$W_i^{\varepsilon}(t) = \sqrt{\varepsilon} \sum_{j=0}^{t/\varepsilon - 1} \left\langle z_j - \overline{g}, \mathbf{e}_i \right\rangle$$

Then $W_i^{\varepsilon}(\cdot)$ converges weakly to a real-valued Wiener process $W_i(\cdot)$ with covariance $t\sigma_i^2$, where

$$\sigma_i^2 = \mathbf{E}[\langle z_0 - \overline{g}, \mathbf{e}_i \rangle]^2 + 2\sum_{j=1}^{\infty} \mathbf{E} \langle z_0 - \overline{g}, \mathbf{e}_i \rangle \langle z_j - \overline{g}, \mathbf{e}_i \rangle.$$
(4.101)

Proof Note that with the use of inner product in ℓ_2 , $\{\langle z_n - \overline{g}, \mathbf{e}_i \rangle\}$ is a real-valued mixing sequence with mean 0. The desired convergence follows from the functional invariance principle for mixing process; see [99, Chapter 7] (see also [25, 61]).

Lemma 4.8.3. Under the conditions of Lemma 4.8.2, for $i \neq l$, $EW_i^{\varepsilon}(t)W_l^{\varepsilon}(t) = 0$. As a result, the limit Wiener processes $W_i(\cdot)$ and $W_l(\cdot)$ are independent.

Proof It is straightforward that

$$\mathbf{E}W_{l}^{\varepsilon}(t)W_{l}^{\varepsilon}(t) = \varepsilon \mathbf{E} \sum_{k=0}^{t/\varepsilon-1} \sum_{j=0}^{t/\varepsilon-1} \langle z_{j} - \overline{g}, \mathbf{e}_{i} \rangle \langle z_{k} - \overline{g}, \mathbf{e}_{l} \rangle$$
$$= \varepsilon \mathbf{E} \sum_{k=0}^{t/\varepsilon-1} \sum_{j=0}^{t/\varepsilon-1} \langle z_{j} - \overline{g}, \mathbf{e}_{i} \mathbf{e}_{l}'(z_{k} - \overline{g}) \rangle$$
$$= 0 \text{ since } \mathbf{e}_{i} \mathbf{e}_{l}' = 0 \in \mathbb{R}^{\infty \times \infty}.$$

Since $\mathbf{E}W_i^{\varepsilon}(t) = 0$, we conclude that $\Sigma(W_i^{\varepsilon}(t), W_l^{\varepsilon}(t)) = 0$. Consequently, $\Sigma(W_i(t), W_l(t)) = 0$, and as a result $W_i(t)$ and $W_l(t)$ are independent Wiener processes.

Proposition 4.8.4. Under the conditions of Lemma 4.8.2, define

$$W^{\varepsilon}(t) = \sqrt{\varepsilon} \sum_{j=0}^{t/\varepsilon - 1} [z_j - \overline{g}].$$
(4.102)

Then $W^{\varepsilon}(\cdot)$ converges weakly to $W(\cdot)$ such that

$$W(t) = \sum_{i=0}^{\infty} W_i(t) \mathbf{e}_i, \qquad (4.103)$$

and the covariance operator is given by

$$\mathbf{E}\langle W(t), v \rangle \langle W(t), z \rangle = t \langle z, \Gamma v \rangle = t \sum_{i=0}^{\infty} \sigma_i^2 \langle \mathbf{e}_i, v \rangle \langle \mathbf{e}_i, z \rangle \text{ for } v, z \in \ell_2,$$
(4.104)

where σ_i^2 is defined in (4.101).

Proof In view of the definition of (4.102), for any $\delta > 0$, t > 0, $0 < s \le \delta$, with $\mathbf{E}_t^{\varepsilon}$ denotes the conditional expectation with respect to $\mathscr{F}_t^{\varepsilon}$, using the mixing properties, we can show that

$$\lim_{\delta \to 0} \limsup_{\varepsilon \to 0} [\sup_{0 \le \delta \le s} \mathbf{E}_t^{\varepsilon} \langle W^{\varepsilon}(t+s) - W^{\varepsilon}(t), W^{\varepsilon}(t+s) - W^{\varepsilon}(t) \rangle] = 0$$

Thus $W^{\varepsilon}(\cdot)$ is tight in $D([0,\infty); \ell_2)$. We can extract any weakly convergent subsequence and denote the limit by $W(\cdot)$. We next characterize its limit.

Again, using (4.102)

$$W^{\varepsilon}(t) = \sum_{i=0}^{\infty} W_i^{\varepsilon}(t) \mathbf{e}_i = \sqrt{\varepsilon} \sum_{i=0}^{\infty} \sum_{j=0}^{t/\varepsilon-1} \langle z_j - \overline{g}, \mathbf{e}_i \rangle \mathbf{e}_i.$$

Therefore, for each $l \in \mathbb{N}$,

$$\mathbf{E}[\langle W^{\varepsilon}(t), \mathbf{e}_l \rangle]^2 = \mathbf{E} ||W_l^{\varepsilon}(t)||^2 = t \sigma_l^2.$$

By virtue of exponential decay property of $\overline{g}^i \propto i^{-\beta}$, $\sum_{l=0}^{\infty} \sigma_l^2 < \infty$. By Lemma 4.8.2, $W_i^{\varepsilon}(\cdot)$ converges weakly to $W_i(\cdot)$. By virtue of Lemma 4.8.3, $W_i(\cdot)$ are independent Wiener processes. In view of the definition of Wiener process on ℓ_2 , we conclude that $W^{\varepsilon}(\cdot)$ converges weakly to $W(\cdot)$ such that (4.103) holds. In addition, the structure of the covariance operator (4.104) is obtained.

We proceed to obtain the desired weak convergence of $v^{\varepsilon}(\cdot)$. Since the stochastic differential equation (4.25) is linear, there is a unique solution for each initial condition. The rest of the proof is similar to the finite dimensional counter part with necessary modifications similar to that of the proof of Theorem 4.5.2.

4.8.10 Proof of Theorem 4.5.4

Proof. Before proceeding to the main proof, we first state a preliminary result. The proofs of the assertions below can be found in [156, Theorem 3.6 and Theorem 4.3, respectively] and are thus omitted.

Lemma 4.8.5. Under Assumption 4.2.1, the following claims hold:

(a) Denote $p_n^{\rho} = [P(\theta_n^{\rho} = 1), \dots, P(\theta_n^{\rho} = M)]$ and the n-step transition probability by $(A^{\rho})^n$ with A^{ρ} given in (4.2) with $\rho = \varepsilon^2$. Then

$$p_n^{\rho} = p(\rho n) + O(\rho + \rho^{-k_0 t/\rho}),$$

$$(A^{\rho})^{n-n_0} = \Xi(\rho n, \rho n_0) + O(\rho + e^{-k_0 (t-t_0)/\rho}),$$
(4.105)

where $p(t) \in \mathbb{R}^{1 \times M}$ and $\Xi(t,t_0) \in \mathbb{R}^{M \times M}$ are the continuous-time probability vector and transition matrix satisfying

$$\frac{\frac{dp(t)}{dt} = p(t)Q, \ p(0) = p_0,}{\frac{d\Xi(t,t_0)}{dt} = \Xi(t,t_0)Q, \ \Xi(t_0,t_0) = I,}$$
(4.106)

with $t_0 = \rho n_0$ and $t = \rho n$.

(b) $\theta^{\rho}(\cdot)$ converges weakly to $\theta(\cdot)$, a continuous-time Markov chain generated by Q.

To analyze the algorithm, the techniques developed in the proof of Theorem 4.5.2 are used along with the ideas and methods developed in [155]. The developments are similar in the approach and the results, but are more complex due to the added switching process. For example, with modifications, Step 1 in the proof of Theorem 4.5.2 can still be carried out. Also Step 2 can be proved. So the sequence $\{\hat{g}^{\varepsilon}(\cdot)\}$ is tight.

To characterize the limit, we still use martingale averaging techniques. We shall only highlight the main difference here. In carrying out the analysis similar to that of Step 3 in the proof of Theorem 4.5.2, we will encounter the following term

$$\mathbf{E}h(\widehat{g}^{\varepsilon}(t_{l_{1}}):t_{1} \leq \kappa) \left[\sum_{lm_{\varepsilon}=t/\varepsilon}^{(t+s)/\varepsilon-1} \Delta_{\varepsilon} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \frac{1}{m_{\varepsilon}} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} Y_{j}(\theta_{j}) \rangle \right] \\
= \mathbf{E}h(\widehat{g}^{\varepsilon}(t_{l_{1}}):t_{1} \leq \kappa) \left[\sum_{lm_{\varepsilon}=t/\varepsilon}^{(t+s)/\varepsilon-1} \Delta_{\varepsilon} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \frac{1}{m_{\varepsilon}} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} \mathbf{E}_{lm_{\varepsilon}} Y_{j}(\theta_{j}) \rangle \right] \\
= \mathbf{E}h(\widehat{g}^{\varepsilon}(t_{l_{1}}):t_{1} \leq \kappa) \left[\sum_{lm_{\varepsilon}=t/\varepsilon}^{(t+s)/\varepsilon-1} \Delta_{\varepsilon} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \frac{1}{m_{\varepsilon}} \sum_{\theta=1}^{M} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} \mathbf{E}_{lm_{\varepsilon}} Y_{j}(\theta) I_{\{\theta_{j}=\theta\}} \rangle \right].$$
(4.107)

Since $Y_j(\theta)$ and θ_j are independent, we have

$$\frac{1}{m_{\varepsilon}} \sum_{\theta=1}^{M} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} \mathbf{E}_{lm_{\varepsilon}} Y_{j}(\theta) I_{\{\theta_{j}=\theta\}}
= \frac{1}{m_{\varepsilon}} \sum_{\iota_{0}=1}^{M} \sum_{\theta=1}^{M} \sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} \mathbf{E}_{lm_{\varepsilon}} Y_{j}(\theta) P(\theta_{j}=\theta|\theta_{lm_{\varepsilon}}=\iota_{0}) I_{\{\theta_{lm_{\varepsilon}}=\iota_{0}\}}.$$
(4.108)

For each $\theta \in \mathcal{M}$, the averaging of $Y_j(\theta)$ can be carried out as in Case 1. We concentrate on the term involving Markov chain. By virtue of Lemma 4.8.5, noting $\rho = \varepsilon^2$ and using (4.105), we have

$$[A^{\rho}]^{j-lm_{\varepsilon}} = \Xi(\varepsilon^2 j, \varepsilon^2 lm_{\varepsilon}) + O(\varepsilon^2 + e^{-k_0(\varepsilon^2 j - \varepsilon^2 lm_{\varepsilon})/\varepsilon^2}).$$

Because we are working with (4.28) and the stepsize is ε . In the interval $[l\Delta_{\varepsilon}, l\Delta_{\varepsilon} + \Delta_{\varepsilon})$ with $\Delta_{\varepsilon} = \varepsilon m_{\varepsilon}$, it is readily seen that $\Xi(\varepsilon^2 j, \varepsilon^2 l m_{\varepsilon}) \to \Xi(0,0) = I$ as $\varepsilon \to 0$. As a result,

$$P(\theta_j = \theta | \theta_{lm_{\varepsilon}} = \iota_0) + o_{\varepsilon}(1) = \delta_{\iota_0, \theta} + o_{\varepsilon}(1) = \begin{cases} 1, & \text{if } \iota_0 = \theta \\ 0, & \text{otherwise} \end{cases} + o_{\varepsilon}(1),$$

where $o(1) \to 0$ as $o_{\varepsilon}(1) \to 0$ as $\varepsilon \to 0$. Putting the above estimates in (4.108), we obtain the limit in probability of

$$\frac{1}{m_{\varepsilon}}\sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1}\mathbf{E}_{lm_{\varepsilon}}Y_{j}(\theta)P(\theta_{j}=\theta|\theta_{lm_{\varepsilon}}=\iota_{0})I_{\{\theta_{lm_{\varepsilon}}=\iota_{0}\}}$$

is the same as that of

$$\sum_{i=1}^{\infty} \mathbf{e}_i \theta_i(\theta) \delta_{\iota_0 \theta} I_{\{\theta^{\varepsilon^2}(\varepsilon^2 l m_{\varepsilon}) = \theta_0\}}$$

This further leads to that as $\varepsilon \to 0$,

$$\begin{split} \mathbf{E}h(\widehat{g}^{\boldsymbol{\varepsilon}}(t_{l_{1}}):t_{1}\leq\kappa) & \left[\sum_{lm_{\varepsilon}=t/\varepsilon}^{(t+s)/\varepsilon-1} \Delta_{\varepsilon} \langle \nabla f(\widehat{g}_{lm_{\varepsilon}}), \frac{1}{m_{\varepsilon}}\sum_{j=lm_{\varepsilon}}^{lm_{\varepsilon}+m_{\varepsilon}-1} Y_{j}(\boldsymbol{\theta}_{j}) \rangle\right] \\ \rightarrow \mathbf{E}h(\widehat{g}(t_{l_{1}}):t_{1}\leq\kappa) & \left[\int_{t}^{t+s} \langle \nabla f(\widehat{g}(\tau)), \boldsymbol{\theta}_{i}(\boldsymbol{\theta}) P(\boldsymbol{\theta}(0)=\boldsymbol{\theta}) \rangle d\tau\right] \\ = \mathbf{E}h(\widehat{g}(t_{l_{1}}):t_{1}\leq\kappa) & \left[\int_{t}^{t+s} \langle \nabla f(\widehat{g}(\tau)), \boldsymbol{\theta}_{i}(\boldsymbol{\theta}) \mathbf{e}_{i} p_{\boldsymbol{\theta}} \rangle d\tau\right]. \end{split}$$

Conclusions

The unifying theme of this thesis was to devise a set of theories and methods for statistical signal processing on graphs (possibly random) which involves multi-agent Bayesian estimation, social learning, stochastic approximation algorithms and adaptive filtering, and dynamics of random graphs to understand the effects of the interactions among agents on the estimation/tracking problem. Part I of this dissertation was devoted to mis-information management problem in multi-agent state estimation over social networks. Part II of this thesis deals with tracking a time-varying degree distribution of a dynamic social network using noisy observations. This chapter concludes this work and presents a summary of findings along with some some directions for future research and development.

5.1 Summary of Findings in Part I

Over the last decade there has been a growing interest in social networks which facilitate our dayto-day activities such as our decision-makings, social communications, and sharing news or stories. Many of these activities involve estimation, learning or decision making using social networks such as rating and review systems, micro-blogging platforms, and online social networks. In these estimation problems, structure of the underlying social network imposes a communication constraint and dictates who talks to whom. First part of this thesis was motivated by such social networks that comprises of a set of agents (social sensors) that seek to estimate an underlying state of nature interactively. Part I dealt with mis-information management problem in two different information exchange protocols:

- Chapter 2 considered an information exchange protocol where agents broadcast their (private) beliefs over the network. In such a protocol, each agent records its (private) observations, then, it combines it with the information received from other agents to form its belief about state of nature. Finally, it transmits the updated belief over the network. Note that this is not social learning, since (private) beliefs of agents are transmitted.
- Chapter 3 used social learning to model the interactions among agents. In this model, each agent computes its private belief using the local observation and the information received from the network. Then, based on its private belief, it chooses an action from a finite set such

that a local cost function is minimized. In the social learning context considered in Chapter 3, as opposed to the one in Chapter 3, private beliefs of agents which depend on the private observations are not communicated to other agents. Instead, public beliefs of them, which can be computed directly from their actions, are broadcasted over the network.

In both of the above protocols, mis-information propagation arises as a result of the correlation introduced by the loops in the communication graph and the recursive nature of Bayesian models. In Chapter 2, we present an optimal information aggregation scheme that completely removes the mis-information associated with estimates of agents under some conditions on the topology of the communication graph. The optimal mis-information removal algorithm proposed in Chapter 2 requires knowledge of transitive closure matrix of the communication graph. For the scenarios where due to a random delay in communications among agents the transitive closure matrix is not completely known, a sub-optimal algorithm is proposed to mitigate "double counting" events which are more likely to happen.

The social learning model considered in Chapter 3 results in two interesting phenomena: (i) herding where all rational agents end up choosing the same action and (ii) mis-information propagation which produces a bias in the public belief and results in overconfidence. Inspired by online rating and review systems, Chapter 3 presented a 5-step protocol to mimic the interactions among agents (social sensors) that aim to estimate an underlying state of nature. It then introduced a fair protocol that prevents mis-information propagation and was used as a benchmark. Using that, an algorithm is invented for the administrator of the rating system to deploy such that it maintains fair ratings. The results of a psychology experiment on a group which has been conducted by our colleagues at the University of British Columbia on a group of undergraduate students to study the learning behavior of humans in a society. The experiment showed that the interactions of the agents can be modeled using a social learning model. We further showed that, as a result of the information exchange protocol (between individuals within a group) and the recursive nature of decision making process, data incest arises in a large fraction of trials in the experiment.

Tools

Graph theory first started 250 years ago with a paper written by Leonhard Eüler on the Seven Bridges of Knigsberg published in 1763 [23]. Since then, it has became a powerful tool to model several networks. Each vertex denotes an agent (or group of individuals) in the social network and each edge depicts a relationship between different agents in the social network. In this work, graph theoretic tools and definitions are used to model the flow of information through the network. Also, the necessary and sufficient condition for complete mis-information removal is presented in terms of adjacency and transitive closure matrices of the underlying communication graph.

Social learning is another mathematical abstraction which is used in this work to model the

interactions among agents (social sensors) in social networks.

5.2 Summary of Findings in Part II

The second part of this thesis was motivated by the importance of the degree distribution in analysis of social networks. The interaction between nodes in dynamic social networks is not always fixed and may evolve over time. An example of such time-varying dynamics is the seasonal variations in friendships among college students. Chapter 4 considered social networks where dynamics of the underlying graph is evolving according to realization of a Markov chain. The Markov-modulated random graph generated by Algorithm 4.5 mimics such networks where the dynamics (the connection/deletion probabilities p,q) depend on the state of nature and evolve over time. Algorithm 4.5 models these time variations as a finite state Markov chain $\{\theta_n\}$. This model forms our basis for analysis of social networks.

Markov-modulated duplication-deletion random graphs are analyzed in terms of degree distribution. When the size of graph is fixed (r = 0) and ρ is small, the expected degree distribution of the Markov-modulated duplication-deletion random graph can be uniquely computed from (4.6) for each state of the underlying Markov chain. This result allows us to express the structure of network (degree distribution) in terms of the dynamics of the model. We also showed that, when the size of the graph is fixed and there is no Markovian dynamics, the random graph generated according to Algorithm 4.5 satisfies a power law with exponent computed from (4.11). The importance of this result is that a single parameter (power law exponent) characterizes the structure of a possibly very large dynamic network.

Moreover, a stochastic approximation algorithm is presented to estimate the empirical degree distribution of the finite duplication-deletion random graph using noisy observations. The proposed stochastic approximation algorithm (4.14) does not assume any knowledge of the Markov-modulated dynamics of the graph (state of nature). Since the expected degree distribution can be uniquely computed for each state of underlying Markov chain, a social sensor can be designed based on (4.14) to track the state of nature using the noisy observations of nodes' degrees. These noisy observations can be samples of the degree sequence of each node; that is, some nodes are randomly chosen and inquired about the number of connections that they have. Using perturbed Lyapunov function, we showed that the tracking error of the stochastic approximation algorithm is small and bounded.

Then, in Chapter 4 a Hilbert-space-valued stochastic approximation algorithm is presented to track the expected degree distribution of the infinite duplication-deletion random graph without Markovian dynamics. The asymptotic behavior of such an algorithm is analyzed in terms of the power law degree distribution. Finally, we extended the analysis to a Hilbert-space-valued stochastic approximation algorithm that aims to track a Markov-modulated probability mass function with
denumerable support. Using weak convergence methods, it was shown that the estimates obtained via such an algorithm converge weakly to the solution of an ordinary differential equation. It was also shown that the interpolated sequence of scaled tracking error converges weakly to the solution of a stochastic differential equation.

Tools

We borrowed techniques from graph theory to perform the degree distribution analysis provided in Chapter 4. Such techniques were previously used in analysis of complex networks in the social and economic networks literature, see [41, 80]. Adaptive filtering, stochastic approximation, and Markov-switched systems are among the tools which are employed in this chapter in order to estimate the expected degree distribution and consequently the state of underlying Markov chain. Weak convergence analysis and functional central limit theorem are used as mathematical abstractions to analyze the performance of such tracking algorithms.

5.3 Directions for Future Research and Development

There is clearly much work to be done in the area of signal processing on complex (and possibly random) networks to understand the behavior of agents in social networks. In this section we present some of the immediate extensions of the work presented in this dissertation.

Mis-information removal algorithms in Bayesian quickest-time change detection:

In *change detection* problem, the objective is to detect a random change time by optimizing the trade-off between number of observations and delay penalty [124, 134]. This problem is very similar to the sensing problems that have been considered in the first part of this dissertation and involve interactive sensing with the goal of detecting a random change in state of nature. Multi-agent Bayesian change detection involves a set of agents (sensors) where each agent estimates an underlying state and then, using Bayesian models, updates the posterior distribution of the change. Then it sends this updated posterior distribution (or a myopic action obtained based on a local cost optimization [87]) over the network. This process repeats until a global decision maker detects a change and all agents stops making observations. In other words, using all the local information (decisions or distributions), the goal in the quickest-time change detection is to detect a change and make a global decision. Because of the recursive nature of the Bayesian estimators and the information exchange protocol, mis-information propagation can arise in such system. One immediate extension of the work presented in this dissertation is to investigate the effect of mis-information propagation in such interactive sensing scenarios and possibly to devise a mis-information removal algorithm

for each agent to employ, such that the decision of the global decision maker is not affected by the mis-information propagation.

Analyzing spread of contagious disease through network

Diffusion of information and disease through society has been studied extensively in the literature of social network analysis, see [80, 107, 108, 122, 146]. One of the models that used to model the diffusion of information in the social network is called the Susceptible-Infected-Susceptible (SIS) model [122]. Consider a social network where agents interacts with other nodes that are dictated by the structure of the network. Each agent in such a social network can be in two states: (i) infected, or (ii) not infected but susceptible to becoming infected. In SIS model, infected nodes can recover and become susceptible again. In the model considered in [107, 108], the assumption is that the degree of nodes remain fixed. One extension or the work presented in the second part of this thesis, is to study the diffusion of information in dynamic graphs (where the graph is not fixed and is evolving according to Procedure 4.5 presented in Chapter 4). Investigating contagion in such dynamic networks is non-standard in two ways: First, the spread of disease⁴⁴ is dynamic and depends on the number and the distribution of infected nodes over the network. Second, the structure of the underlying graph (which dictates who talks to whom) is evolving over time. Analyzing diffusion process in such networks (both in stationary state and in transient phase) can be a possible avenue for further research and development.

Mis-information removal algorithms in non-Bayesian models

In our work, we addressed mis-information management problem in two scenarios: (i) constrained filtering and (ii) social learning over social networks. In both of these scenarios, Bayesian models for information aggregation have been used. There exists a large body of works in the literature that consider Bayesian models to formulate the learning behavior of humans [1, 13, 16, 24, 136]. On the other hand, a body of social learning literature focuses on non-Bayesian social learning models, see [3, 14, 15, 56, 81]. In these models, agents use a simple rule-of-thumb to update their beliefs from their private information about state of nature and those received from other agents. Similar to the learning problems considered in Part I of this dissertation, as a result of the recursive nature of the information aggregation schemes and possible loops in communication graph, mis-information propagation may arise in such non-Bayesian learnings over social networks. Mis-information management in learning with non-Bayesian models over social networks can be a possible direction for further research and development.

⁴⁴Instead of spread of disease or information, we can also consider a scenario where nodes are interactively deciding whether to adapt a technology (infected) or not (remain susceptible). The aim now is to study the rate of adaptation of new technology in the network.

Mis-information propagation mitigation for specific network structures

In the proposed mis-information algorithm presented in this thesis, we used a combination of the previous estimates in order to completely mitigate data incest. However, the optimality of this approach demands a topological constraint to hold. An extension to the work presented in this dissertation is to investigate mis-information management problem in graphs with specific structures that do not necessarily satisfy such a constraint. An alternative approach is to "censor" the estimates of some nodes that are most likely to be polluted with mis-information. This can be viewed as deliberately cutting some edges in order to lower the risk of mis-information propagation. Cutting these edges can be useful for some specific graph structures (like bi-bipartite graphs) specially when the topological constraints is not satisfied and, thus, optimal mis-information removal is not possible.

Bibliography

- [1] D. Acemoglu, M. A. Dahleh, I. Lobel, and A. Ozdaglar. Bayesian learning in social networks. *The Review of Economic Studies*, 78(4):1201–1236, 2011.
- [2] D. Acemoglu and A. Ozdaglar. Opinion dynamics and learning in social networks. *Dynamic Games and Applications*, 1(1):3–49, Mar. 2011.
- [3] D. Acemoglu, A. Ozdaglar, and A. ParandehGheibi. Spread of (mis) information in social networks. *Games and Economic Behavior*, 70(2):194–227, 2010.
- [4] L. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, Jul. 2005.
- [5] T. Aeppel. 50 million users: The making of an angry birds internet meme. *Wall Street Journal*, Mar. 2015.
- [6] C. C. Aggarwal and T. Abdelzaher. Integrating sensors and social networks. Social Network Data Analytics, pages 397–412, 2011.
- [7] D. Agrawal, C. Budak, and A. El Abbadi. Information diffusion in social networks: Observing and affecting what society cares about. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 2609–2610, New York, NY, USA, 2011. ACM.
- [8] Y. Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *Proceedings of 16th International Conference* on World Wide Web, pages 835–844, Banff, AB, Canada, May 2007.
- [9] M. Albakour, C. Macdonald, and I. Ounis. Identifying local events by using microblogs as social sensors. In *Proceedings of 10th Conference on Open Research Areas in Information Retrieval*, pages 173–180, Lisbon, Portugal, May 2013.
- [10] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, Jan. 2002.

- [11] S. Asur and B. A. Huberman. Predicting the future with social media. In 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, volume 1, pages 492–499, Toronto, ON, Canada, 2010.
- [12] R. J. Aumann. Agreeing to disagree. The Annals of Statistics, 4(6):1236–1239, Nov. 1976.
- [13] R. J. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1–18, Jan. 1987.
- [14] V. Bala and S. Goyal. Learning from neighbours. *The review of economic studies*, 65(3):595–621, 1998.
- [15] V. Bala and S. Goyal. Conformism and diversity under social learning. *Economic theory*, 17(1):101–120, 2001.
- [16] A. Banerjee. A simple model of herd behavior. *Quaterly Journal of Economics*, 107(3):797–817, Aug. 1992.
- [17] A.-L. Barabási and A. Reka. Emergence of scaling in random networks. *Science*, 286(5439):509, 1999.
- [18] T. Bayes. An essay toward solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53, 1764.
- [19] A. Benveniste, M. Métivier, and P. P. Priouret. Adaptive algorithms and stochastic approximations. Springer-Verlag, 1990.
- [20] A. Benveniste, M. Métivier, and P. Priouret. Adaptive Algorithms and Stochastic Approximations. Springer, 2012.
- [21] J. M. Bernardo and A. Smith. *Bayesian theory*, volume 405. John Wiley & Sons, 2009.
- [22] P. Biernacki and D. Waldorf. Snowball sampling: Problems and techniques of chain referral sampling. *Sociological Methods & Research*, 10(2):141–163, Nov. 1981.
- [23] N. L. Biggs, E. K. Lloyd, and R. J. Wilson. *Graph Theory*. Oxford University Press, 1986.
- [24] S. Bikchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change as information cascades. *Journal of Political Economy*, 100(5):992–1026, Oct. 1992.
- [25] P. Billingsley. Convergence of probability measures, volume 493. John Wiley & Sons, 2009.
- [26] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Compu*tational Science, 2(1):1–8, Mar. 2011.

- [27] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnády. The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, 18(3):279–290, 2001.
- [28] P. Bonhard and M. A. Sasse. knowing me, knowing you using profiles and social networking to improve recommender systems. *BT Technology Journal*, 24(3):84–98, 2006.
- [29] T. Brehard and V. Krishnamurthy. Optimal data incest removal in Bayesian decentralized estimation over a sensor network. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 173–176, Honolulu, Hawaii, Apr. 2007.
- [30] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 665–674. ACM, 2011.
- [31] D. Budak and A. El Abbadi. Information diffusion in social networks: Observing and influencing societal interests. *Proceedings of the VLDB Endowment*, 4(12), 2011.
- [32] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava. Participatory sensing. In Workshop on World-Sensor-Web: Mobile Device Centric Sensor Networks and Applications, Boulder, CO, 2006.
- [33] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: Percolation on random graphs. *Physical Review Letters*, 85(25):5468–5471, Dec. 2000.
- [34] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. Peterson, H. Lu, X. Hong, X. Zheng, M. Musolesi, K. Fodor, and G. Ahn. The rise of people-centric sensing. *IEEE Internet Computing*, 12(4):12–21, 2008.
- [35] C. Chamley. Rational Herds: Economic Models of Social Learning. Cambridge University Press, 2004.
- [36] C. Chamley, A. Scaglione, and L. Li. Models for the diffusion of beliefs in social networks: An overview. *IEEE Signal Processing Magazine*, 30(3):16–29, May 2013.
- [37] C. P. Chamley. *Rational Herds: Economic Models of Social Learning*. Cambridge University Press, 2003.
- [38] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based approach to geo-locating twitter users. In *Proceedings of 19th ACM International Conference on Information and Knowledge Management*, pages 759–768, Toronto, ON, Canada, Oct. 2010.

- [39] W. H. Chin, D. B. Ward, and A.G. Constantinides. Semi-blind mimo channel tracking using auxiliary particle filtering. In *Global Telecommunications Conference*, 2002. GLOBE-COM'02. IEEE, volume 1, pages 322–325. IEEE, 2002.
- [40] N. A. Christakis and J. H. Fowler. Social network sensors for early detection of contagious outbreaks. *PLoS ONE*, 5(9), Sep. 2010.
- [41] F. Chung and L. Lu. *Complex Graphs and Networks*. Conference Board of the Mathematical Sciences, National Science Foundation (U.S.), 2006.
- [42] F. Chung, L. Lu, T. G. Dewey, and D. G. Galas. Duplication models for biological networks. *Journal of Computational Biology*, 10(5):677–687, Oct. 2003.
- [43] T. Conley and C. Udry. Social learning through networks: The adoption of new agricultural technologies in Ghana. *American Journal of Agricultural Economics*, 83(3):668–673, Aug. 2001.
- [44] C. Cooper and A. Frieze. A general model of web graphs. *Random Structures and Algorithms*, 22(3):311–335, 2003.
- [45] R. Corten. Composition and structure of a large online social network in the netherlands. *PLoS ONE*, 7(4), Apr. 2012.
- [46] R. F. Curtain and A. J. Pritchard. Infinite dimensional linear systems theory. *Lecture notes in control and information sciences*, 8, 1978.
- [47] B. Debatin, J. P. Lovejoy, A. K. Horn, and B. N. Hughes. Facebook and online privacy: Attitudes, behaviors, and unintended consequences. *Journal of Computer-Mediated Communication*, 15(1):83–108, 2009.
- [48] M. H. DeGroot. Reaching a consensus. Journal of the American Statistical Association, 69(345):118–121, 1974.
- [49] P. Dodin and V. Nimier. Distributed tracking systems and their optimal inference topology. In Proceedings of 5th International Conference on Information Fusion, volume 1, pages 585– 592, Annapolis, MD, Jul. 2002.
- [50] X. G. Doukopoulos and G. V. Moustakides. Adaptive power techniques for blind channel estimation in cdma systems. *IEEE Transactions on Signal Processing*, 53(3):1110–1120, Mar. 2005.
- [51] X. G. Doukopoulos and G. V. Moustakides. Blind adaptive channel estimation in OFDM systems. *IEEE Transactions on Wireless Communications*, 5(7):1716–1725, Jul. 2006.

- [52] X. G. Doukopoulos and G. V. Moustakides. Fast and stable subspace tracking. *IEEE Trans*actions on Signal Processing, 56(4):1452–1465, Apr. 2008.
- [53] R. Durrett. Random Graph Dynamics. Cambridge Series on Statistical and Probabilistic Mathematics. Cambridge University Press, 2007.
- [54] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, May 2006.
- [55] H. Ebel, L. I. Mielsch, and S. Bornholdt. Scale-free topology of e-mail networks. *Physical Review E*, 66, 2002.
- [56] G. Ellison and D. Fudenberg. Rules of thumb for social learning. *Journal of Political Econ*omy, 101(4):612–643, Aug. 1993.
- [57] G. Ellison and D. Fudenberg. Word-of-mouth communication and social learning. *The Quar*terly Journal of Economics, 110(1):93–125, 1995.
- [58] Y. Ephraim. A bayesian estimation approach for speech enhancement using hidden markov models. *IEEE Transactions on Signal Processing*, 40(4):725–735, 1992.
- [59] P. Erdös and T. Gallai. Graphs with given degrees of vertices. *Mat Lapok*, 11:264–274, 1960.
- [60] P. Erdös and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5:17–61, 1960.
- [61] S. N. Ethier and T. G. Kurtz. *Markov processes: characterization and convergence*, volume 282. John Wiley & Sons, 2009.
- [62] S. Eubank, H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429:180– 184, May 2004.
- [63] R. Evans, V. Krishnamurthy, G. Nair, and L. Sciacca. Networked sensor management and data rate control for tracking maneuvering targets. *IEEE Transactions on Signal Processing*, 53(6):1979–1991, Jun. 2005.
- [64] D. Gale and S. Kariv. Bayesian learning in social networks. *Games and Economic Behavior*, 45(2):329–346, Nov. 2003.
- [65] A. G.Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, Nov. 2010.

- [66] J. D. Geanakoplos and H. M. Polemarchakis. We can't disagree forever. *Journal of Economic Theory*, 28(1):192–200, Oct. 1982.
- [67] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. Proceedings of the National Academy of Sciences, 99(12):7821–7826, 2002.
- [68] S. Goel and M. J. Salganik. Respondent-driven sampling as markov chain Monte Carlo. *Statistics in Medicine*, 28(17):2209–2229, Jul 2009.
- [69] L. Guo, L. Ljung, and G. J. Wang. Necessary and sufficient conditions for stability of lms. *IEEE Transactions on Automatic Control*, 42(6):761–770, 1997.
- [70] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph.I. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.
- [71] M. Hamdi and V. Krishnamurthy. Removal of data incest in multi-agent social learning in social networks. arXiv:1309.6687 [cs.SI], Sep. 2013.
- [72] M. Hamdi, V. Krishnamurthy, and G. Yin. Tracking the empirical distribution of a Markovmodulated duplication-deletion random graph. *arXiv*:1303.0050 [cs.IT], Feb. 2013.
- [73] V. Havel. A remark on the existence of finite graphs. *Časopis Pěst Mat* 80:477, 1955.
- [74] D. D. Heckathorn. Respondent-driven sampling: A new approach to the study of hidden populations. *Social Problems*, 44(2):174–199, May 1997.
- [75] Y.C. Ho and RCKA Lee. A bayesian approach to problems in stochastic estimation and control. *Automatic Control, IEEE Transactions on*, 9(4):333–339, 1964.
- [76] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Physical Review E*, 65(5), May 2002.
- [77] B. Huang and T. Jebara. Maximum likelihood graph structure estimation with degree distributions. In *Analyzing Graphs: Theory and Applications, NIPS Workshop*, volume 14, 2008.
- [78] M. B. Hurley. An information theoretic justification for covariance intersection and its generalization. In *Proceedings of the 5th International Conference on Information Fusion*, volume 1, pages 505–511, Annapolis, MD, Jul. 2002.
- [79] B. Ifrach, C. Maglaras, and M. Scarsini. Monopoly pricing in the presence of social learning. *NET Institute Working Paper No. 12-01*, 2011.
- [80] M. O. Jackson. Social and Economic Networks. Princeton University Press, 2008.

- [81] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi. Non-bayesian social learning. *Games and Economic Behavior*, 76(1):210–225, 2012.
- [82] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.
- [83] Y. Kanoria and O. Tamuz. Tractable bayesian social learning on trees. In Proceedings of IEEE International Symposium on Information Theory, pages 2721 –2725, Cambridge, MA, Jul. 2012.
- [84] H. Kim, Z. Toroczkai, P. L. Erdös, I. Miklós, and L. A. Székely. Degree-based graph construction. *Journal of Physics A: Mathematical and Theoretical*, 42(39), Oct. 2009.
- [85] G. Kossinets and D. J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88–90, Jan. 2006.
- [86] P. L. Krapivsky, G. J. Rodgers, and S. Redner. Degree distributions of growing networks. *Physical Review Letters*, 86(23):5401–5404, Jun. 2001.
- [87] V. Krishnamurthy. Quickest detection POMDPs with social learning: Interaction of local and global decision makers. *IEEE Transactions on Information Theory*, 58(8):5563–5587, 2012.
- [88] V. Krishnamurthy, O. Namvar Gharehshiran, and M. Hamdi. Interactive sensing and decision making in social networks. *Foundations and Trends in Signal Processing*, 7(1-2):1–196, 2013.
- [89] V. Krishnamurthy and M. Hamdi. Data fusion and mis-information removal in social networks. In *Proceedings of 15th International Conference on Information Fusion*, pages 1142– 1149, Singapore, Jul. 2012.
- [90] V. Krishnamurthy and M. Hamdi. Mis-information removal in social networks: Constrained estimation on dynamic directed acyclic graphs. *IEEE Journal of Selected Topics in Signal Processing*, 7(2):333–346, Apr. 2013.
- [91] V. Krishnamurthy and H. V. Poor. Social learning and Bayesian games in multiagent signal processing: How do local and global decision makers interact? *IEEE Signal Processing Magazine*, 30(3):43–57, May 2013.
- [92] V. Krishnamurthy and H. V. Poor. A tutorial on interactive sensing in social networks. *IEEE Transactions on Computational Social Systems*, 1(1), 2014.
- [93] V. Krishnamurthy and T. Ryden. Consistent estimation of linear and non-linear autoregressive models with Markov regime. *Journal of Time Series Analysis*, 19(3):291–307, May 1998.

- [94] V. Krishnamurthy, K. Topely, and G. Yin. Consensus formation in a two-time-scale markovian system. SIAM Journal on Multiscale Modeling and Simulation, 7(4):1898–1927, 2009.
- [95] Vikram Krishnamurthy and Sujay Bhatt. Sequential detection of market shocks using riskaverse agent based models. *arXiv preprint arXiv:1511.01965*, 2015.
- [96] Vikram Krishnamurthy and William Hoiles. Online reputation and polling systems: Data incest, social learning, and revealed preferences. *Computational Social Systems, IEEE Transactions on*, 1(3):164–179, 2014.
- [97] H. J. Kushner. Approximation and Weak Convergence Methods for Random Processes, with Applications to Stochastic Systems Theory. MIT Press, Cambridge, MA, 1984.
- [98] H. J. Kushner and A. Shwartz. Stochastic approximation in hilbert space: Identification and optimization of linear continuous parameter systems. *SIAM journal on control and optimization*, 23(5):774–793, 1985.
- [99] H. J. Kushner and G. Yin. Stochastic Approximation and Recursive Algorithms and Applications, volume 37 of Stochastic Modeling and Applied Probability. Springer-Verlag, 2nd edition, 2003.
- [100] P. J. Lamberson. Social learning in social networks. *The BE Journal of Theoretical Economics*, 10(1), 2010.
- [101] R. Lee and K. Sumiya. Measuring geographical regularities of crowd behaviors for twitterbased geo-social event detection. In *Proceedings of 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 1–10, San Jose, CA, Nov. 2010.
- [102] S. H. Lee, P.-J. Kim, and H. Jeong. Statistical properties of sampled networks. *Physical Review E*, 73(1), Jan. 2006.
- [103] J. Leskovec. SNAP library. http://snap.stanford.edu/data/index.html.
- [104] E. Lieberman, C. Hauert, and M. A. Nowak. Evolutionary dynamics on graphs. *Nature*, 433(7023):312–316, Jan. 2005.
- [105] T. T. Lin and S. S. Yau. Bayesian approach to the optimization of adaptive systems. *Systems Science and Cybernetics, IEEE Transactions on*, 3(2):77–85, 1967.
- [106] C. Lingji, P. O. Arambel, and R. K. Mehra. Estimation under unknown correlation: covariance intersection revisited. *IEEE Transactions on Automatic Control*, 47(11):1879–1882, Nov. 2002.

- [107] D. López-Pintado. Contagion and coordination in random networks. *International Journal of Game Theory*, 34(3):371–381, Oct. 2006.
- [108] D. López-Pintado. Diffusion in complex social networks. *Games and Economic Behavior*, 62(2):573–590, Mar. 2008.
- [109] M. Luca. Reviews, reputation, and revenue: The case of Yelp.com, Technical Report 12-016. Harvard Business School, 2011.
- [110] R. Maben, G. V. Moustakides, and J. S. Baras. Adaptive sampling for linear state estimation. SIAM Journal on Control and Optimization, 50(2):672–702, 2012.
- [111] G. V. Moustakides. Exponential convergence of products of random matrices: Application to adaptive algorithms. *International Journal of Adaptive Control and Signal Processing*, 12(7):579–597, 1998.
- [112] K. P. Murphy. Machine learning: a probabilistic perspective. MIT press, 2012.
- [113] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.
- [114] M. E. J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.
- [115] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89, Oct. 2002.
- [116] M. E. J. Newman. The structure and function of complex networks. SIAM Review, 45(2):167– 256, 2003.
- [117] M. E. J. Newman. Power laws, pareto distributions and zipf's law. Contemporary physics, 46(5):323–351, 2005.
- [118] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(S1):2566–2572, 2002.
- [119] N. P. Nguyen, G. Yan, M. T. Thai, and S. Eidenbenz. Containment of misinformation spread in online social networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, WebSci '12, pages 213–222. ACM, 2012.
- [120] T. D. Nielsen and F. V. Jensen. Bayesian networks and decision graphs. Information Science and Statistics. Springer, 2007.

- [121] R. Pastor-Satorras, E. Smith, and R. V. Solé. Evolving protein interaction networks through gene duplication. *Journal of Theoretical Biology*, 222(2):199–210, May 2003.
- [122] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86(14):3200, 2001.
- [123] J. Pearl. Fusion, propagation, and structuring in belief networks. Artificial Intelligence, 29(3):241–288, Sep. 1986.
- [124] H. V. Poor and O. Hadjiliadis. *Quickest Detection*. Cambridge University Press, 2008.
- [125] S. J. Press. Subjective and objective Bayesian statistics: principles, models, and applications, volume 590. John Wiley & Sons, 2009.
- [126] Naren Ramakrishnan, Benjamin J Keller, Batul J Mirza, Ananth Y Grama, and George Karypis. Privacy risks in recommender systems. *IEEE Internet Computing*, 5(6):54–63, 2001.
- [127] J. Reason and W. Ren. Estimating the optimal adaptation gain for the lms algorithm. In Proceedings of the 32nd IEEE Conference on Decision and Control,, pages 1587–1588. IEEE, 1993.
- [128] S. Reece and S. Roberts. Robust, low-bandwidth, multi-vehicle mapping. In *Proceedings of* 8th International Conference on Information Fusion, volume 2, Philadelphia, PA, Jul. 2005.
- [129] F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.
- [130] C. Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation.* Springer Science & Business Media, 2007.
- [131] A. Rosi, M. Mamei, F. Zambonelli, S. Dobson, G. Stevenson, and Y. Juan. Social sensors and pervasive services: Approaches and perspectives. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 525–530, Seattle, WA, Mar. 2011.
- [132] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: Real-time event detection by social sensors. In *Proceedings of 19th International Conference on World Wide Web*, pages 851–860, Raleigh, NC, Apr. 2010.
- [133] L. Scharf. Statistical signal processing, volume 98. Addison-Wesley Reading, MA, 1991.
- [134] A. N. Shiryaev. On optimum methods in quickest detection problems. *Theory of Probability and its Applications*, 8(1):22–46, 1963.

- [135] J. Shrager, T. Hogg, and B. A. Huberman. A graph-dynamic model of the power law of practice and the problem-solving fan-effect. *Science*, 242(4877):414–416, 1988.
- [136] L. Smith and P. Sørensen. Pathological outcomes of observational learning. *Econometrica*, pages 371–398, 2000.
- [137] V. Solo and X. Kong. *Adaptive signal processing algorithms: stability and performance*. Prentice-Hall information and system sciences series. Prentice Hall, 1995.
- [138] S. M. Stigler. The true title of bayess essay. Statistical Science, 28(3):283-288, 2013.
- [139] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268, 2001.
- [140] R. Taylor. Contrained switchings in graphs. In K. L. McAvaney, editor, *Combinatorial Mathematics VIII*, volume 884 of *Lecture Notes in Mathematics*, pages 314–336. Springer Berlin Heidelberg, 1981.
- [141] H. L. Van Trees. Detection, estimation, and modulation theory. John Wiley & Sons, 2004.
- [142] Anne M Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136, 1980.
- [143] A. Tripathi and S. Vijay. A note on a theorem of erdös and gallai. Discrete Mathematics, 265(1–3):417–420, Apr. 2003.
- [144] M. Trusov, A. V. Bodapati, and R. E. Bucklin. Determining influentual users in internet social networks. *Journal of Marketing Research*, XLVII:643–658, Aug. 2010.
- [145] B. Tuttle. Fact-checking the crowds: How to get the most out of hotel-review sites. *Time magazine*, Jul. 2013.
- [146] F. Vega-Redendo. Complex Social Networks. Economic Society Monographs. Cambridge University Press, 2007.
- [147] A. Wagner. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Molecular Biology and Evolution*, 18(7):1283–1292, 2001.
- [148] T. Wang and H. Krim. Statistical classification of social networks. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pages 3977–3980, Kyoto, Japan, Mar. 2012.
- [149] T. Wang, H. Krim, and Y. Viniotis. A generalized Markov graph model: Application to social network analysis. *IEEE journal of Selected Topics in Signal Processing*, 7(2):318–332, Apr. 2013.

- [150] X. Xie and R. J. Evans. Multiple target tracking and multiple frequency line tracking using hidden markov models. *IEEE Transactions on Signal Processing*, 39(12):2659–2676, Dec. 1991.
- [151] Y. Xin and T. Jaakkola. Controlling privacy in recommender systems. In Advances in Neural Information Processing Systems, pages 2618–2626, 2014.
- [152] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, Jul. 2005.
- [153] G. Yin and V. Krishnamurthy. LMS algorithms for tracking slow Markov chains with applications to hidden Markov estimation and adaptive multiuser detection. *IEEE Transactions* on Information Theory, 51(7):2475–2490, Jul. 2005.
- [154] G. Yin, V. Krishnamurthy, and C. Ion. Regime switching stochastic approximation algorithms with application to adaptive discrete stochastic optimization. *SIAM Journal on Optimization*, 14(4):1187–1215, 2004.
- [155] G. Yin, Y. Sun, and L. Y. Wang. Asymptotic properties of consensus-type algorithms for networked systems with regime-switching topologies. *Automatica*, 47(7):1366–1378, 2011.
- [156] G. Yin and Q. Zhang. *Discrete-Time Markov Chains: Two-Time-Scale Methods and Applications*, volume 55 of *Applications of Mathematics*. Srpinger-Verlag, New York, NY, 2005.
- [157] G. Yin and C. Zhu. Hybrid Switching Diffusions: Properties and Applications, volume 63 of Stochastic Modeling and Applied Probability. Springer, 2010.

Appendix A

Some Graph Theoretic Definitions

Graph, Directed Graph, Path and Directed Acyclic Graph (DAG):

- A graph G_N comprising of N nodes is a pair (V, E), where $V = \{v_1, \dots, v_N\}$ is a set of nodes (also called vertices), and $E \subset V \times V$ is a set of edges between the nodes.
- Graph G_N is an undirected graph if for any (v_i, v_j) ∈ E then (v_j, v_i) ∈ E and a graph is said to be directed if (v_j, v_i) ∈ E is not a consequence of (v_i, v_j) ∈ E.
- A *path* is an alternating sequence of nodes and edges, beginning and ending with an edge, in which each node is incident to the two edges that precede and follow it in the sequence.
- A *Directed Acyclic Graph (DAG)* is a directed graph with no path that starts and ends at the same node.
- A family of DAGs \mathscr{G}_N is defined as a set of DAGs $\{G_1, \ldots, G_N\}$ where G_n is the sub-graph of G_{n+1} such that for $n = 1, \ldots, N-1$

$$\begin{cases} V_n = V_{n+1}/v_{n+1}, \\ E_n = E_{n+1}/\{(v_i, v_{n+1}) \in E_{n+1} | v_i \in V_{n+1}\}. \end{cases}$$
(A.1)

Adjacency and Transitive Closure matrices:

Let $G_N = (V, E)$ denote a graph with N nodes $V = \{v_1, \dots, v_N\}$.

• The Adjacency Matrix \overline{A} of G_N is an $N \times N$ matrix whose elements $\overline{A}(i, j)$ are given by

$$\overline{A}(i,j) = \begin{cases} 1 & \text{if } (v_j, v_i) \in E ,\\ 0 & \text{otherwise} \end{cases} \quad A(i,i) = 0.$$
(A.2)

• The Transitive Closure Matrix *T* of *G_N* is an *N*×*N* matrix whose elements *T*(*i*, *j*) are given by *T*(*i*, *i*) = 1, and

$$T(i,j) = \begin{cases} 1 & \text{if there is a path between } v_j \text{ and } v_i, \\ 0 & \text{otherwise} \end{cases}$$
(A.3)

The following shows the special form of the adjacency matrix of the directed acyclic graph and provides a closed form expression to compute the transitive closure matrix from the adjacency matrix of a directed acyclic graph.

Lemma A.1. A sufficient condition for a graph G_N to be a DAG is that the Adjacency matrix A is an upper triangular matrix. For a DAG G_N , the Transitive Closure Matrix T is related to the Adjacency matrix by

$$T = Q(\{\mathbf{I}_N - A\}^{-1}).$$
(A.4)

Here, \mathbf{I}_N is the $N \times N$ identity matrix, and Q denote the matrix valued "quantization" function so that for any $N \times N$ -matrix B, Q(B) is the $N \times N$ matrix with elements

$$Q(B)(i,j) = \begin{cases} 0 & \text{if } B(i,j) = 0 , \\ 1 & \text{if } B(i,j) \neq 0 \end{cases}.$$
 (A.5)

Proof: This result is derived from the classical interpretation of matrix $\{\mathbf{I}_N - A\}^{-1}$. The entry in row *i* and column *j* of this matrix gives the number of paths from node *i* to node *j*.

To deal with information flow in a social network, we now introduce the concept of a *family* of DAGs.

Remark A.1. For the sake of simplicity in notations, let us define two vector representatives of adjacency and transitive closure matrices of directed acyclic graph. For each graph $G_n \in \mathcal{G}_N$, let the $n \times n$ matrices \overline{A}_n and T_n , respectively, denote the adjacency matrix and transitive closure matrix. Define the following:

$$\begin{cases} t_n \in \{0,1\}^{1 \times (n-1)}: \text{ transpose of first } n-1 \text{ elements of nth column of } T_n, \\ b_n \in \{0,1\}^{1 \times (n-1)}: \text{ transpose of first } n-1 \text{ elements of nth column of } \overline{A}_n. \end{cases}$$
(A.6)

Remark A.2. As can be straightforwardly followed from the construction of adjacency and transitive closure matrices in (A.1), for a family of DAGs $\mathscr{G}_N = \{G_1, \ldots, G_N\}$, for any $n \in \{1, \ldots, N-1\}$, the adjacency matrix \overline{A}_n and transitive closure matrix T_n of graph G_n are respectively the $n \times n$ left upper matrices of the adjacency matrix A_{n+1} and transitive closure matrix T_{n+1} of graph G_{n+1} .

Appendix B

A Note on Degree-based Graph Construction

The first step in numerical studies of social networks is the graphical modeling of such networks. A graph can be uniquely determined by the adjacency matrix (also known as the connectivity matrix) of the graph. However, in the graphical modeling of social networks (specially when the size of the network is relatively large), the only available information is the degree sequence of nodes, and not the adjacency matrix of the network.

Definition B.1. *The degree sequence, denoted by* **d***, is a non-increasing sequence comprising of the vertex degrees of the graph vertices.*

The degree sequence, in general, does not specify the graph uniquely; there can be a large number of graphs that realize a given degree sequence. It is straightforward to show that not all integer sequences represent a true degree sequence of a graph. For example, sequence $\mathbf{d} = \{2, 1, 1\}$ represents a tree with two edges, but $\mathbf{d} = \{3, 2, 1\}$ cannot be realized as the degree sequence of a simple graph. Motivated by social network applications, this section addresses the following two questions given a degree sequence \mathbf{d} :

- Existence Problem: Is there any simple graph that realizes d?
- Construction Problem: How can we construct all simple graphs that realize a true degree sequence **d**?

There are two well-known results that address the existence problem: (i) the Erdös-Gallai theorem [59] and the Havel-Hakimi theorem [70, 73]. These theorems provide necessary and sufficient conditions for a sequence of non-negative integers to be a true degree sequence of a simple graph. Here, we recall these results without proofs.

Theorem B.1 (Erdös-Gallai, [59]). Let $d_1 \ge d_2 \ge \cdots \ge d_n > 0$ be integers. Then, the degree sequence $\mathbf{d} = \{d_1, \cdots, d_n\}$ is graphical if and only if

1. $\sum_{i=1}^{n} d_i$ is even;

2. *for all* $1 \le k < n$:

$$\sum_{i=1}^{k} d_i \le k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}.$$
(B.1)

It is shown in [143] that there is no need to check (B.1) for all $1 \le k \le n-1$; it suffices to check (B.1) for $1 \le k \le s$, where *s* is chosen such that $d_s \ge s$ and $d_{s+1} < s+1$. Note that, in degree-based graph construction, we only care about nodes of degree greater than zero; zero-degree nodes are isolated nodes which can be added to the graph consisting of nodes of degree greater than zero.

The Havel-Hakimi theorem also provides necessary and sufficient conditions for a degree sequence to be graphical. It also gives a greedy algorithm to construct a graph from a given graphical degree sequence.

Theorem B.2 (Havel-Hakimi, [70, 73]). Let $d_1 \ge d_2 \ge \cdots \ge d_n > 0$ be integers. Then, the degree sequence $\mathbf{d} = \{d_1, \cdots, d_n\}$ is graphical if and only if the degree sequence $\mathbf{d}' = \{d_2 - 1, d_3 - 1, \cdots, d_{d_1+1} - 1, d_{d_1+2}, \cdots, d_n\}$ is graphical.

In the following, we provide algorithms to construct a simple graph from a true degree sequence. In the construction problem, the degree sequence is treated as a collection of *half-edges*; a node with degree d_i has d_i half-edges. One end of these half-edges are fixed at node *i*, but the other ends are free. An edge between node *i* and node *j* is formed by connecting a half-edge from node *i* to a half-edge from node *j*. The aim is to connect all these half edges such that no free half-edge is left. The Havel-Hakimi theorem provides a recursive procedure to construct a graph from a graphical degree sequence. This procedure is presented in Algorithm B.6

Using Algorithm B.6, one can sample from graphical realizations of a given degree sequence. In this algorithm, each vertex is first connected to nodes with lower degrees. Therefore, Algorithm B.6 generates graphs where high-degree nodes tend to connect to low-degree nodes; the resulting graph has assortative property [84, 115]. To overcome this problem, one way is to perform *edge swapping* repeatedly such that the final graph looses its assortative property. In the edge swapping method, two edges (for example (1,2) and (3,4)) can be swapped (to (1,4) and (2,3)) without changing the degree sequence. Edge swapping method is also used to generate all samples from a given degree sequence; one sample is generated via Algorithm B.6 and then, by use of Markov chain Monte-Carlo algorithm based on edge swapping [140], other samples from the graphical realizations of the degree sequence are obtained.

In [84] a swap-free algorithm is proposed to generate all graphical realizations of a true degree sequence. Before proceeding to Algorithm B.7, we first provide definitions which will be used in this algorithm.

Definition B.2. Let $\mathbf{d} = \{d_1, \dots, d_n\}$ be a degree sequence of a simple graph and N(i) be the set of adjacent nodes of node *i*. Then, the degree sequence reduced by N(i) is denoted by $\mathbf{d}|_{N(i)} =$

Algorithm B.6 Creating a sample graph from a given degree sequence Given a graphical sequence $d_1 \ge d_2 \ge \cdots \ge d_n > 0$: Start from i = 1

- (i) Initialize k = n.
- (ii) Connect (one half-edge of) node *i* to (a half-edge of) node *k*
- (iii) Check that the resulting degree sequence is graphical

- if Yes:

- 1. Let k = k 1.
- 2. Repeat (i).
- if No:
 - 1. Save the connection between node i and node k
 - 2. If node *i* has any half-edges left, let k = k 1 and repeat (i)

(iv) If i < n, then, $i \leftarrow i + 1$ and repeat (i).

 $\{d_1|_{N(i)}, \dots, d_n|_{N(i)}\}$ with elements defined as follows

$$d_k|_{N(i)} = \begin{cases} d_k - 1, & \text{if } k \in N(i), \\ 0, & \text{if } k = i, \\ d_k, & \text{otherwise.} \end{cases}$$
(B.2)

Definition B.3. Let $(a_1, a_2, ..., a_n)$ and $(b_1, b_2, ..., b_n)$ be two sequences. Then, $(a_1, a_2, ..., a_n) <_{CR}$ $(b_1, b_2, ..., b_n)$ if and only if there exists an index m such that $1 \le m \le n$ and $a_m < b_m$ and $a_i = b_i$ for all $m < i \le n$.

Let **d** be a non-increasing graphical degree sequence. In order to construct the graph, we need to find all possible neighbors N(i) ("allowed set") of each node *i* such that if we connect this node to its allowed set, then the resulting reduced degree sequence $\mathbf{d}|_{N(i)}$ is also graphical, i.e., the graphicality is preserved. Algorithm B.7 provides a systematic way (swap-free) to generate all graphical realizations of a true degree sequence (by means of finding all possible neighbors of each node).

Algorithm B.7 Constructing all graphs from a graphical degree sequence [84]

Given a graphical sequence $d_1 \ge d_2 \ge \cdots \ge d_n > 0$

Start from i = 1

Step 1: Find neighbors with highes index of node *i*

The aim is to find $A_R(i)$:

- (i) Initialize k = n.
- (ii) Connect node i to node k
- (iii) Check that the resulting degree sequence is graphical

– if Yes:

1. Let
$$k = k - 1$$

2. Repeat (i).

- if No:
 - 1. Save the connection between node i and node k
 - 2. If node *i* has any stubs left, let k = k 1 and repeat (i)

Step 2: Find all possible neighbors of node *i*

With $<_{CR}$ defined in (B.3), the aim is to find

 $\mathscr{A}(i) = \{N(i) = \{v_1, \cdots, v_{d_i}\}; N(i) <_{CR} A_R(i) \text{ and } \mathbf{d}|_{N(i)} \text{ is graphical}\}$

- (i) Find all sets of nodes who are colexicographically smaller than $A_R(1)$ (prospective neighbor sets).
- (ii) Connect node *i* to those neighbors and check if the resulting degree sequence is graphical.

Step 3: For every $N(i) \in \mathscr{A}(i)$:

- Connect node *i* to *N*(*i*)
- Discard node *i* and compute the reduced degree sequence $\mathbf{d}|_{N(i)}$
- Create all graphs from degree sequence $\mathbf{d}|_{N(i)}$ using this algorithm