University of Miami
Scholarly Repository

Open Access Dissertations | Electronic Theses and Dissertations

2016-12-06

Who's Shaping Who's Opinions? Estimating the Degree of Influence Exerted by Each Contact in a Social Network

Luis Eduardo Castro Abril
University of Miami, luiscastroabril@gmail.com

Follow this and additional works at: https://scholarlyrepository.miami.edu/oa_dissertations

Recommended Citation
Castro Abril, Luis Eduardo, "Who's Shaping Who's Opinions? Estimating the Degree of Influence Exerted by Each Contact in a Social Network" (2016). *Open Access Dissertations*. 1762.
https://scholarlyrepository.miami.edu/oa_dissertations/1762

This Embargoed is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarly Repository. It has been accepted for inclusion in Open Access Dissertations by an authorized administrator of Scholarly Repository. For more information, please contact repository.library@miami.edu.

UNIVERSITY OF MIAMI

WHO'S SHAPING WHO'S OPINIONS? ESTIMATING THE DEGREE OF INFLUENCE EXERTED BY EACH CONTACT IN A SOCIAL NETWORK

By
Luis Eduardo Castro Abril

A DISSERTATION

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

Coral Gables, Florida

December 2016

© 2016
Luis Eduardo Castro Abril
All Rights Reserved

UNIVERSITY OF MIAMI

A dissertation submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

WHO'S SHAPING WHO'S OPINIONS? ESTIMATING THE DEGREE OF INFLUENCE EXERTED BY EACH CONTACT IN A SOCIAL NETWORK

Luis Eduardo Castro Abril

Approved:

_____ Nazrul I Shaikh, Ph.D. Assistant Professor of Industrial Engineering	_____ Murat Erkoc, Ph.D. Associate Professor of Industrial Engineering
_____ Shihab S. Asfour, Ph.D. Professor of Industrial Engineering	_____ Guillermo Prado, Ph.D. Dean of Graduate School
_____ Kamal Premaratne, Ph.D. Professor of Electrical and Computer Engineering Department	

CASTRO ABRIL, LUIS EDUARDO (Ph.D., Industrial Engineering)
Who's Shaping Who's Opinions? Estimating the Degree of Influence Exerted by Each Contact in a Social Network (December 2016)

Abstract of a dissertation at the University of Miami

Dissertation supervised by Professor Nazrul I. Shaikh.
No. of pages in text. (191)

It is important in domains such as organizational behavior, politics, marketing, sociology, psychology, engineering, and economics to study how people that belong to a network form their opinions on a specific topic, how these opinions evolve, and how a consensus is reached. Researchers have mainly focused on estimating hidden opinions, identifying opinion trends, opinion leaders, consensus points, and the impact of the structure of the network on consensus. This dissertation proposes a model that (a) enables the tracking of the opinions of every member of a network, and (b) helps identifying who is influencing who's opinions and to what degree when the opinions and connections of only a small subset of the people in the network is observed.

This dissertation has three phases. Phase I provides the theoretical background of opinion dynamics and the philosophical concepts that surround individuals and opinions. Phase II develops the infrastructure and mathematical framework required to model opinions, opinion dynamics, and consensus over large networks. Phase III uses the mathematical constructs of the previous phase to present a statistical framework for the estimation of the influence that each person has on the opinions of others (influence network). The estimation problem is solved for the scenarios of complete information (when we know who is connected to whom and we have opinion measurements for all the

agents); and when we have incomplete information (only part of the network and measurements are observed).

The intellectual contribution of this dissertation are: (i) a large scale network simulation R-engine tool based on parallel computing and efficient memory constructs, (ii) a mathematical model where the opinion dynamics process is a functional system of stochastic equations that extends the concepts of opinion, update rule, local and global consensus; (iii) an expected value opinion model that transforms of our functional system of stochastic equations into a mean opinion model, enabling us to estimate the influence parameters and track the mean opinion distributions under complete information; and (iv) an online particle filter algorithm that estimates the influence parameters and latent opinion distributions of each agent as new information becomes available. The proposed algorithm handles effectively incomplete information and large scale estimation.

The outcome of the proposed research has numerous applications. For example, it could be used to track and obtain consensus opinions and information about opinion leaders in trending news on Twitter, or it could be used to estimate the existence of an influence pathway (ghost links) between two selected individuals that have no obvious connection. Other applications include tracking opinions about new products from reviews and fashion trends from blogs; and early identification of trends and/or deviant behavior for homeland security reasons.

DEDICATION

This work is dedicated to my wife and parents.

My wife unmeasurable love and incredible support throughout my stay at UM kept me always motivated and ready for every single challenge that I had to face. I could not have done it without you.

My parents love and example since I was a little kid set the initial but solid foundations of my love for science and research. Their amazing work on me always keep me asking and wanting to learn more.

The three of them are always teaching me how I can be not only a better scientist, but the best possible human being that I can be.

ACKNOWLEDGEMENTS

First and foremost I want to thank my advisor Dr. Nazrul I. Shaikh for his immense support (moral, intellectual and financial) and time. It has been an honor to work with him as his Ph.D. student. During my years at UM, he has taught me not only how rigorous and formal research needs to be, but also that a great scientist needs to be also a great human being. The joy and passion that he has for his research was contagious and motivational every step all the way. I am thankful to him for all the productive and stimulating research that we were able to conduct.

I also want to thank the members of my thesis committee. Dr. Asfour for his support not only at the final stage but during all these years. His support, as the leader of the Industrial Engineering department, was always present not only at a student level, but also at a personal level. Dr. Erkoc classes and useful comments for my dissertation were always a key note during my years of research. Dr. Premaratne for his useful comments on the computational formality of my research and the philosophical motivations for my models.

In general, I would like to thank all the professors from the Industrial Engineering department for their knowledge, useful comments and support during these years.

Finally, I want to make explicit my gratitude to Dr. Shaikh's and Dr. Premaratne's funding through grant N00014-10-1-0140 from the US Office of Naval Research (ONR). This endowment made possible my financial support while in UM.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

The ability to track how opinions about people, places, issues or things are evolving over time is important in a multitude of domains such as engineering, social sciences, and management science. This ability enables firms to identify emerging trends and dominant opinions, opinion leaders, influencers and stubborn agents, and take proactive/predictive actions. The objective of this dissertation is to build a model that (a) enables the tracking of the opinions of every member of a network, and (b) helps identifying who is influencing whose opinions and to what degree.

The complexity of this research problem is determined by (i) the size of the network; (ii) whether who knows who is known or unknown to the analyst; (iii) richness of the observational data in terms of volume, velocity, and veracity; and (iv) the number and location of the agents. Unlike prior research, the proposed research does not assume that opinions are directly observed, and the information about who influences who (network connectivity) is completely known.

The current dissertation has seven chapters. The second chapter focuses on the understanding of current theoretical basis and developments in opinion dynamics so key limitations and insights can be pointed out. In addition, we identify computational requirements for the analysis of opinion dynamics.

We present a computationally robust approach for mid-scale and large scale network simulation and metrics calculation in chapters 3 and 4. We describe in detail the simulation and metric algorithms, and the parallel processing approach used. Our main

result is an R-based library capable of simulating and analyzing large scale networks in regular laptop and PC computers.

On chapter five we develop a functional stochastic opinion dynamics model and a concept of consensus that is conducive for large studies. We model an individuals' opinion at any point of time as a unique stochastic process for each agent. Though computationally more expensive, this concept: (a) it is better aligned with recent research in the areas of economics, psychology, and sociology which shows that an opinion in not a fixed point (Budescu and Rantilla 2000, Budescu and Yu 2007, Jackson and López-Pintado 2013, Li, Myaeng, and Kim 2007, Miao, Li, and Zeng 2010, Nakata 2003) and recently developments in opinion dynamics models under Dempster-Shafer Theory (Dabarera et al. 2016, Wickramarathne et al. 2014); (b) it allows for a less restrictive definition of consensus as compared to the notion of strong/weak consensus that is used in extant literature (Hegselmann and Krause 2002, Li, Scaglione, et al. 2013, Li, Braunstein, et al. 2013, Olfati-Saber and Murray 2004, Ren, Beard, and Atkins 2005). We use mathematical proofs and simulation based studies to highlight the nuances that the opinion model and the group structure (network topology) together have on whether, when, and where a consensus would be reached.

On chapter six, we build upon the opinion model and the notion of consensus to (a) track the opinions of each individual[1] in the network and (b) estimate the influence that each individual has on everyone she is connected to. The proposed research differentiates between a contact network and an influence network. The contact network captures the information about whether individual i is connected to individual j whereas the influence

[1] The terms individuals, agents, and nodes will be used interchangeably.

network captures information about the strength of the influence that individual i exerts on j. We assume that all the information from the contact network is known while the influence network is fully unknown and needs to be estimated. This chapter follows our theoretical functional stochastic opinion dynamics model and the classical estimation setting of restricted maximum likelihood (Wooldridge 2010) . We provide a complete analysis of our estimates analyzing the stability, identifiability, and asymptotic properties of the model. We outline the estimation algorithm and provide the estimation results and insights.

The assumption that the contact network is completely known is relaxed in chapter 7. We address the problem of uncovering the opinion distributions of each individual in the network, and the estimation of the influence that each individual has on everyone she is connected to when: (i) the contact network is partially observed, (ii) opinion measurements are partially observed, (iii) both cases. This lack of complete information on the network structure increases the amount of information that needs to be extracted from the data. The proposed research solves the information problem using an online particle filter based learning algorithm (Carvalho, Johannes, et al. 2010, Lopes and Carvalho 2013) that can learn the influence parameters online as new data becomes available. We test the sample size requirements for tracking the opinions of all individuals in the network while measuring the opinions of only a select few.

Finally, we provide a complete references section with all the cited works that served as theoretical base for this research.

The outcome of this dissertation has numerous applications. For example, it could be used to track and obtain consensus opinions and information about opinion leaders in a trending news item on twitter, or it could be used to estimate the existence of an influence

pathway (ghost links) between two select individuals who have no obvious connections. Other applications may include opinions tracking about new products from reviews and fashion trends from blogs; or early identification of trends and/or deviant behavior for homeland security purposes.

Chapter 2: Current Literature on Opinion Dynamics

Overview

Opinion dynamics is a multidisciplinary subject that models the inception of opinions[2], their evolution, and ultimately the consensus that might be formed from these opinions. These interaction occurs among agents in a social networks. These structure can be viewed as the underlying configuration that governs (restricts) the interaction amongst these agents. The structure of the social network therefore influences how opinions are formed and if, when, and how a consensus will be reached.

The study of opinion dynamics is relevant for multiple purposes. From an economical and business point of view, opinion dynamics models two apparently different processes. First, it seeks to explain why some products or services are persistently chosen and adopted among a social network (Bass 1969). However, in a broader concept, opinion dynamics in economics deals with the diffusion of ideas as part of the innovation process in a society. Social networks are seen as the natural abstract representation of society (Golub and Jackson 2010). From this point of view, the formation of opinion, evolution, and their final configuration could explain not only products but economic actions and incentives that social –rational- agents have (Acemoğlu et al. 2013, Yildiz et al. 2011a). With this respect, the literature is entirely theoretical using simulation as a mean of example, not evidence.

From a psychology and sociological point of view, opinions dynamics try to understand under which structures an idea can be spread, shared, modified and adopted in society (Rogers 1962). This fact is mathematically modelled using the sociophysics approach – (Sobkowicz 2009). Under this view, the evolution of discrete and continuous opinions is

[2] It is important to mention that the words idea and concept are used as synonyms for opinion across the opinion dynamics literature – Nowak et.al (1990), Sobkowicz (2007), Jackson (2010).

modelled using models and measures from physics, mostly spin models and mean field theory models (Galam 1997, Kacperski 1999, Lewenstein, Nowak, and Latané 1992, Nowak and Lewenstein 1996); and mixtures of these models with linear update rules (Deffuant et al. 2000, Hegselmann and Krause 2002, Sznajd-Weron and Sznajd 2000). Ultimately, the final configuration of opinions into consensus or groups is characterized based on the modelling rules. Most of these studies focus their results on evidence from theory and simulation, not in real life data or experimentation.

From the engineering perspective, opinion and consensus is an important concept in transmission, automation, signal processing and fusion of hard and soft information. For several authors, opinions can be modelled as incoming and outgoing signals whose analysis helps in the solution of coordination problems and optimization tasks (Olfati-Saber, Fax, and Murray 2007, Ren, Beard, and Atkins 2005). In this framework, the social network is a key element that allows the diffusion of opinions and ultimately the rise of consensus. Recent literature has focused on the correct identification of signals and the true value of consensus when the real final state of consensus may be hidden for the whole network (Molavi et al. 2013). Opinions can also be modelled using as main framework DS theory. In this context, opinions are represented by prepositions (hypothesis) and belief functions that represent the probability of occurrence of those prepositions. Agents update and share information based on their location in a network; these information exchange process has as key objective the estimation of a ground truth (consensus) that may not be well known to all the agents, but that can be reached (Dabarera et al. 2016, Wickramarathne et al. 2014).

Opinion dynamics is a theoretical problem with multiple perspectives and applied solutions. This research brings to light these three potentially different visions and unify

them. The rest of the document has 4 sections. Section 2 provides a detailed literature review. First, definitions for opinion, sentiment, belief, consensus and social networks are provided. The opinion modelling process is approached by analyzing 3 updating methods common in the literature. Later, the relevance of the network structure is brought into the analysis highlighting its role and interplay in the opinion process. A summary and classification table with the most relevant works in opinions dynamics in social networks is provided. At the end, philosophical background and reasons for opinion update and opinion pool are discussed. Section 3 summarizes the missing links of the actual literature on opinion dynamics. Finally, Section 4 outlines our research propositions upon which we build the conceptual, theoretical, mathematical and statistical ideas of the next chapters of the dissertation.

Literature Review

Social Network

A social network expresses the underlining correlations, interactions and structures of a set of agents- nodes. Inside this setting, social agents interact with each other sharing information. This interaction whether it may seem limited to a set of neighbors or local vicinity; in fact, depends on the interactions that take place along the whole network structure (Newman 2003).

A social network is formed by a set of nodes V and arcs E. Each individual arc connects uniquely a pair of nodes (i, j). The network is represented by an adjacency matrix D. This

is a square matrix that indicates whether pairs of vertices are adjacent or not in the graph. Given this initial structure, the following particularizations are possible:

- The arcs could be directed or undirected. When the arcs are undirected, D is symmetric matrix where $d_{ij} = d_{ji}$. This basically means that i and j has a mutual relation and influence. In the case of a directed network, $d_{ij} \neq d_{ji}$. This expresses the fact that i can be connected j while j has no connection with j.

- If the graph is weighted, there exists a W matrix where each entry represents the weight that node i has on node j. This matrix does not need to be symmetric, thus $w_{ij} \neq w_{ji}$ since each neighbor of i can have its own particular weight.

- The network could be static or dynamic. In the static setting, the topology of the network does not change in time. When the network is dynamic, for each period there exists a set of new nodes V' and arcs A' which are added to the network. In this context, a dynamic network can be viewed as a pair $Nt_t(V_t, A_t)$ at time t with nodes V and arcs A. In a dynamic setting, new nodes are allowed to emerge through time while old nodes are allowed to disappear (Goldenberg et al. 2010,

Newman 2003). This process could be guided by randomness (Erdős and Simonovits 1965) or preferential attachment (Barabási and Albert 1999).

The analysis of real life social networks can be mainly characterized by (Barabási and Albert 1999, Goldenberg et al. 2010, Newman 2003):

- Having power law distribution
- Allowing the formation of hubs (nodes that exceed the average degree of the network)
- Having the small world effect
- Allowing the formation of a core
- Looping

These real life structures are modelled using two general network models:

- The first model is the random graph. This structure is also known as the Erdős–Rényi model. This model assigns equal probability to all graphs with exactly E edges. The degree distribution of any particular vertex follows a Binomial distribution. In addition, its structure reproduces well the small-world effect. However there are several properties of real world networks that are not satisfied by this model: (i) the clustering coefficient is always low; (ii) the degree distribution is not exponential; (iii) there is no correlation between neighbor vertices; and (iv) there exists no community structure. Random graphs can be undirected and directed. However, due to the randomness in the creation of arcs between nodes, the common features do not change. (Erdős and Simonovits 1965, Newman 2003).
- In (de Solla Price 1965) the authors described the first example of what is now known as scale-free networks. Studying the citation context in research, he

discovered the patterns of in and out connections and the persistently presence of a power law distribution for this type of networks. In this model the rate of new connections from a vertex is proportional to the number of connections that an arc has. A few years later with the advances of simulation, (Barabási and Albert 1999) were able to fully develop this concept into the well-known scale-free model. This representation for social networks relies on the basis of preferential attachment and randomness. The first development of this model is undirected. The bases for its development were the citation and indexing structure of the web pages on the internet. The most relevant characteristics of this model are: (i) the formation of hubs –nodes which degree exceeds the average degree of the network, (ii) the clustering coefficient decreases as the node degree increases, (iii) power law distribution, (iv) reproduction of the small world effect, and (v) the formation of a core based on nodes with the highest degree values. In the case of scale-free graphs, the generation process can also be viewed as the collection of all the graphs of the scale free process of generation. The undirected network structure is the most controversial feature of this first model. On a directed graph setting, the number of out connections from a node is set fixed, while the number of inner connections is allowed to change with time. The addition of new vertexes occur at a variable rate. In the dynamic case this limitation is overcome allowing new nodes to have direct connections with highly connected node –out degree- while the opposite may not be true. The in-degree is modelled as being proportional to the connections and location of the new node (Newman 2003). Since scale-free graphs are still abstraction of real social networks, this model has the following limitations: (i) if a

directed network is proposed, the evolution of the graph will generate acyclic representations, a fact that is not true in social networks, (ii) the out-degree will end up being constant. Despite these limitations, scale-free networks are a strong theoretical model to represent a social network due to the many resemblances with real networks and advantages of its easy-formation and evolving process (Barabási and Albert 1999, Newman and Watts 1999, Newman 2003).

Opinion dynamics

Opinion dynamics analyzes the process of how opinions evolve, and whether after some given time these opinions will become just one, a group of opinions or will totally differ one from another. This analysis based its foundations in the concept of consensus.

There are three main constituents of opinion dynamics (a) opinions, (b) how opinions evolve in time, and (c) when and under which conditions consensus is achieved.

Opinion

An opinion is defined as an individual's preferences about a specific topic, issue or belief (Deffuant et al. 2000, Hegselmann and Krause 2002, Jackson and Rogers 2007, Nowak, Szamrej, and Latané 1990). An opinions is generally represented as $x_{i,t}$ where x is either a scalar or a vector (depending on whether the opinion pertains to only one thing or multiple attributes of an issue) and usually takes continuous values. The opinion of an agent is generally initialized by taking a draw from a common probability distribution that captures the heterogeneity in the population (Deffuant et al. 2000, Fortunato 2004, Hegselmann and Krause 2002, Lewenstein, Nowak, and Latané 1992, Olfati-Saber and Murray 2004, Sznajd-Weron and Sznajd 2000, Yildiz et al. 2011a). The sub-script i represents the individual agents while t indexes the time.

Opinions are represented using scalars or vector values from the real numbers. This mathematic expression summarizes the opinion(s) or belief(s) of an agent.

From the views and works of (Deffuant et al. 2000, Golub and Jackson 2010, Hegselmann and Krause 2002, Olfati-Saber and Murray 2004, Sznajd-Weron and Sznajd 2000) opinions could be modelled as being single attribute values. Under this framework, opinions are real numbers at period t collecting the opinions of all individuals in a network or group. Opinions can be discrete and continuous. In the first case, $x_{i,t}$ assumes discrete values, for instance, it could be used to model Yes/No decisions in the sense of (Deffuant et al. 2000, Lewenstein, Nowak, and Latané 1992). Also, discrete opinions are useful to model coloring or voting problems with more than 2 alternatives (Stocker and Cornforth 2002).

Opinions can also be multiple attribute. In this case, opinions are random vectors that collect for each period t the information of all individuals in a network or group about k attributes. However, the literature has only few works on this line (Sobkowicz 2009). Early formulations of this type have only considered discrete cases (Lanchier 2012, Weisbuch et al. 2002).

Sentiment

Agents hold opinions about a topic. This opinion can have different degrees or levels of sentiment, or have no sentiment at all. A sentiment can be defined as an implicit or explicit expression of an agent expressing an opinion where this expression can either have a positive, negative or neutral connotation (Baccianella, Esuli, and Sebastiani 2010, Liu 2012, Nasukawa and Yi 2003, Pak and Paroubek 2010) . Sentiments involve the agent's

emotions, desires and potential attitudes towards an issue someone or something (Baccianella, Esuli, and Sebastiani 2010).

In opinion dynamics, sentiments are modelled through statistical classification models. In other words, opinions are collected and transformed into numerical values, and these values are related to continuous or discrete indexes that allow the researches to identify the degree of connotation of the opinion of an individual (Baccianella, Esuli, and Sebastiani 2010, Liu 2012, Nasukawa and Yi 2003). Current literature focuses on how we can measure sentiment from real life opinions; how we can model its dynamics and track it is merely a job of opinion models. These concept is discussed later in the present chapter.

Belief

Agents have to confront and evaluate their beliefs period after period based on the inputs they received from a social network (i.e. contacts, external information) (Icard, Pacuit, and Shoham 2010). In this context, belief can be defined as the probability dimension of the opinion that an agent has (Campbell 1967, Fishbein and Raven 1962, Grandy 1973, Douglass, Fishbein, and Ajzen 1977). The belief of an agent is affected by the communication channels, the reliability and validity of the counter-opinion that she gets and observes (Douglass, Fishbein, and Ajzen 1977). In opinion dynamics, beliefs are merely associated with the initial probability distribution of the opinions; then they have not been treated as separate concepts (Acemoğlu et al. 2013, Benczik et al. 2008, Hegselmann and Krause 2002, Sznajd-Weron and Sznajd 2000, Weisbuch, Deffuant, and Amblard 2005) . In addition, in opinion dynamics the attitude concept is not present since we cannot observe the actions or decisions that the agents are making based on their opinions. Nonetheless, it is useful to point out that attitude toward a given opinion,

individual or thing is the result from a change in belief about that same opinion, individual or thing.

Opinion Update Models.

Opinions are usually seen to be dynamic as most agents update their opinions when they receive new information over time. Several opinion update models have been proposed by researchers; the key opinion update models are:

The probability threshold method: In the first case, each individual agent i has a probability p or accepting a new idea or belief. The agent compares the actual value of p to an acceptance threshold h (Hegselmann and Krause 2002, Sznajd-Weron and Sznajd 2000):

$$x_{i,t} = \begin{cases} 1, & p \geq h \\ 0, & otherwise \end{cases} \quad (1)$$

This framework require that opinions be discrete. A more complex variety of the model occurs when the selection can only occur among close neighbors and/or p and/or h varies in time. The Ising model over a quadratic grid is one of the best examples of this rule (Galam 1997, Lewenstein, Nowak, and Latané 1992, Nowak and Lewenstein 1996).

Linear update method (LUM): The LUM is the most popular approach to modeling opinion update. (Acemoğlu et al. 2013, Deffuant and Amblard 2002, Olfati-Saber and Murray 2004, Deffuant et al. 2002). This update rule is given by:

$$x_{i,t} = Ax_{i,t-1} \quad (2)$$

where $x_{i,t}$ is the vector of opinions at time t for agent i, A is a stochastic matrix, where each element a_{ij} summarizes the effect that individual j has over the opinion formation of individual i at time t. The matrix A reflects the network structure and in some sense the concept of homophily from (Jackson and López-Pintado 2013). This means that an agent i

can only be influenced by his neighbors. The opinions are initialized by taking a random draw from a single common distributions for all the agents.

The confidence bound rule: is a hybrid of the linear rule where agents only update their opinion if the value for it is at most at a distance d (Hegselmann and Krause 2002). The mathematical formulations is given by:

$$x_{i,t} = \begin{cases} Ax_{i,t-1} & \text{if } |x_{i,t-1} - x_{j,t-1}| < \text{d} \\ x_{i,t-1} & otherwise \end{cases} \quad (3)$$

Game theory and payoffs: This update rule relies on the payoffs or utility that an agent can obtain when updating his opinion. In general, this reward can be summarized by a utility function $U_{i,t}(.)$ that depends on the actions or beliefs of agent i at time t with respect to the other agents. The best response dynamics (Acemoğlu et al. 2013, Olfati-Saber 2007) for this rule is given by:

$$x_{i,t} = \begin{cases} x^*_{i,t} & \text{if } U^*_{i,t-1} > U_{i,t-1} \\ x_{i,t-1} & otherwise \end{cases} \quad (4)$$

where $x_{i,t}$ is the opinion value or vector associated to the utility level $U_{i,t-1}$. In this sense, Eq.(4) expresses a maximization behavior of the agent when adopting an opinion $\text{x}_{i,t}{}^*$.

Consensus

There are two broad conditions that are used to define consensus:

In a strong sense, consensus is a fixed point x_{sc} regardless of the updating rule. This value is reached when $|\text{x}_{\text{i,t}} - \text{x}_{\text{j,t}}| \to 0$ as $\text{t} \to \infty$ for all i agents in a network (Olfati-Saber and Murray 2004, Ren, Beard, and Atkins 2005).

A weaker condition for consensus (Hegselmann and Krause 2002) states that consensus can be said to achieved when $|\text{x}_{\text{i,t}} - \text{x}_{\text{j,t}}| < \delta$ with $\delta > 0$ when $\text{t} > \text{T}$ only for agents i

and j in a group. The authors in (Li, Braunstein, et al. 2013) define that individual clusters may meet this condition and reached an agreement in opinions inside them.

Networks and Opinion Dynamics

Opinion dynamics is related to the study and exploration of how ideas are spread over a large network. This process occurs over a specific social network setting. This process has its roots in anthropology, economics, sociology, geography and marketing (Bailey 1975, Bass 1969, Gerard and Orive 1987, Robertson 1971). In a general sense, these models have been adapted –consciously or unconsciously- from epidemiology models such as SIR models and later extensions for different phases and immunities (Kermack and McKendrick 1927). The study of these models seeks to understand reasons for adoption of ideas/beliefs among a targeted population.

Most of the pioneer works were developed in the late 1940s. The authors in (Ryan and Gross 1943) showed that social influences play a key role in adoption, rather than only economic factors. (Rogers 1962) developed a first model to approach the diffusion of innovation among individuals. (Bass 1969) formalized diffusion model for adoption of new durable goods. This model can be used to predict future levels and rate of diffusion and analyze the role of external and internal influences. However, the Bass model imposes perfect interaction among individuals.

The first formal works on opinion dynamics go back to (Lewenstein, Nowak, and Latané 1992). This research proposes an approach using the mean field theory of social impact. His model uses a binary opinion state, allowing interactions between individuals with different influential values. Simulations are run for a sparse network, a hierarchical

static network and a lattice. Later, (Galam 1997) analyzed an Ising model for consensus. Individuals are locally connected over a grid, interacting and updating discrete opinions only with their closest neighbors. Consensus is achieved under a condition named as minimization of individual conflicts (achieved at the minimum entropy level). (Kacperski 1999) uses the mean field approach to analyze a binary model for opinion formation. He shows the role of leaders and their opinions over the global opinion and transition within different opinions. This early works started the collaboration of sociology, psychology and physics into a more formal mathematical framework that is known as sociophysics.

The model in (Sznajd-Weron and Sznajd 2000) considers a population of agents with discrete (+1/–1) opinions. Each neighbor is influenced only by its pair of neighbors. Opinions are updated using a ferromagnetic rule. In this model, the opinion flows out from a group of agreeing agents rather than in, from external influences to the agents.

The authors in (Deffuant et al. 2000) uses a population of agents with continuous, bounded range of opinions. A complete connectivity among agents is assumed, so any two randomly chosen agents can share opinions. Opinions are updated after each encounter by the average of those opinions if the two opinions do not differ, given a specific threshold.

In the (Hegselmann and Krause 2002) model the concept of continuous of opinions is extended to not only opinions but influences on the updating rule of opinions. Each agent has a weight vector, so a stochastic influence matrix can be imposed. The original setting considers a fully connected network. This model is the well-known bounded confidence interval opinion model.

These three models have been expanded and test on practical contexts. (Stocker and Cornforth 2002) used simulation in a Random Boolean Network to address the problem of

opinion propagation and consensus. Individuals are model as nodes, which have only a vector value of opinions of length 2 (1 and 0). Starting most of the nodes at 0, idea developers (with an opinion value of 1) start to spread his beliefs among the network. The interaction occurs only among connected nodes. Each time, a pair of nodes meet, based on the generation of a random number, directly update of the states (1 or 0) takes place. The result shows that connectivity in this type of networks assures the transmission of ideas (or contagion). The topology of the network does not vary over time. The problem of propagation of idea is considered under a static network view.

The authors of (Martins, Pereira, and Vicente) studied opinions dynamics through simulation under the setting of an Ising Model. Each agent is connected only to his/her neighbors through a grid (static setting) and updates his/her beliefs based on a Bayes rule. The results show that new ideas spread easily, however, the eventually rise supporters and non-supporters. Data from the introduction of a new medical procedure in Denmark is used to test the model finding a statically significant match between the model and the data.

The theoretical views on opinion dynamics were also developed in the applied sciences. In the field of electric and computation engineering, opinion dynamics and consensus are important topics for automation, control and optimization. In addition, different network setting are discussed, using the ideas of random and free scale networks. In (Ren, Beard, and Atkins 2005) several important definitions for consensus are addressed. One the most important results states that if a spanning tree exists for the network, the final consensus

value is equal to a weighted average of the members of the spanning set. The consensus value is also equal to the average of the initial opinion of all the agents of the network.

A special case is the replicator-imitator dynamics proposed by (Olfati-Saber, Fax, and Murray 2007). In this framework, an agent is placed in a complete network. The agent updates his beliefs and selects a behavior –opinion- that leaves him with the highest payoff. Under this framework, if the network topology is altered, the existence of consensus depends on the initial number of opinions and connections among the network. A wide variety of opinions without interconnections of groups can result in collapse – or non-consensus. Another special case is a modification from the epidemiological model Susceptible Infected Susceptible -SIS-. In (Jackson and López-Pintado 2013) the authors propose a model where an agent updates and adopts a new opinion when at least one infectious agent –with the new opinions- is met.

The social sciences have also several important contributions to the diffusion literature. In (Nowak and Lewenstein 1996, Nowak, Szamrej, and Latané 1990) the authors use a binary model for opinion dynamics to test persuasiveness and supportiveness. He uses a conceptual framework from psychology to explain that the strength of interaction decreases with distance. This process characterizes the opinion phenomena of society. (Banerjee 1992, Bikhchandani, Hirshleifer, and Welch 1992) propose the first models for social learning. A game theory model of sequential games is used to understand how agents make choices. An agent a time t can base his adoption (ideas or beliefs) on the choices that other agent (neighbor) did in t-1. They results show that with infinite memory society will herd –reach a consensus- for the option which yields maximum discounted payoffs. The works of (Acemoğlu et al. 2013, Acemoglu, Dahleh, and Lobel 2009, Ellison and Fudenberg

1995, Gale and Kariv 2003, Jadbabaie et al. 2012, Rosenberg, Solan, and Vieille 2007) present the game theory and opinions in social networks. A key inside from game theory is that if agents do not update their beliefs after each time, no consensus is reached. Consensus will be reached at some point if there is homogeneity in the payoffs across agents.

Other relevant works from the social science involve the use of game theory combined with networks topology and its dynamics. (Jackson and Yariv 2007) discuss the concepts of diffusion over random graphs. Using a game theory approach, the authors show that incentives –or payoffs- matter when a belief or idea needs to be propagated. (Golub and Jackson 2012) address the question of agents adopting an idea based on shortest path communication, random walks and linear updating processes. The main results show that homophily (the concept that people associate with others of his/her similar type) shorter the times of contagion on a network. (Jackson and López-Pintado 2013) proposes a diffusion and contagion model in networks with heterogeneous agents. A theoretical model of disease propagation on a dynamic network is solved (generalizing the SIS model), concluding that homophily fosters propagation.

Other contributions from innovation literature are: (Ehrhardt, Marsili, and Vega-Redondo 2006) show that changes in a network facilitates the diffusion process. The paper shows through simulations that nonlinear dynamics can arise. The model involves learning and diffusion based only on neighbors dynamics spreading through a small network. (Lee et al. 2013) use an agent-based simulation model for the concept of small word (caveman randomly rewired) network model to explain the introduction, purchasing time and market share of the Korean notebook product. A random network setting and agent's behavior

based on neighbors is modeled. The simulation results show that even though the network is sparse, diffusion happens very early.

An interesting practical view is modelled by (Apolloni et al. 2009). Using national survey data for a county in Virginia, profiling population by age group, economic activity and household income, a SIR model for conversation of ideas is simulated. This approach over a real data framework accounts for network formation and spread of ideas subject to time and space restrictions based on the daily activities (For example, only teenagers can belong to the school network from 9 a.m to 3 p.m). The main findings suggest that youths play a significant role in spreading information through a community rapidly, mainly through interactions in schools and recreational activities.

The rest of the authors combine these 3 different perspectives to simulate different processes in opinion dynamics. The authors in (Chen and Yu 2008, Suo and Chen 2008) model opinion dynamics based on a utility framework and dynamic topology for free scale networks. Their findings suggest that adoption leading to consensus depends on the internal and external influence over agents. Their framework is a mixture from the confidence bound model for continuous opinion, with an updating rule based on utility of a Cobb-Douglas function. (Yildiz et al. 2011a) combines the notions of payoff of game theory and consensus from (Olfati-Saber 2007) to show that if in a social network there is a group of influential nodes that do not update their beliefs/opinions over time, no consensus can be reached. Using a stochastic gossip model based on game theory and control theory, the study shows that oscillation of opinions arises in time. Formal proofs and simulation results are provided for random networks and scale free networks.

The authors in (Li, Braunstein, et al. 2013) propose an interesting variation for modeling consensus. Using two parallel networks to model social interaction and opinion, a non-consensus model for random graphs is introduced. This variant allows two different opinions to coexist in a stable relationship for a given threshold. Under extreme cases (i) only one opinion survives, (ii) no consensus is reached. In their research a variation of the voting model is used to analyze opinions dynamics in a social group. Convergence properties are studied using the framework of consensus and unifying criteria's from the electric engineering view. The results show that reaching consensus depends on the threshold selection of the update rule.

(Askari-Sichani and Jalili 2013) studies the evolution of continuous opinions in a modified version of a bounded confidence model. Agents have their own continuous opinions and have different characteristics, model through individual weights. Their work defines a specific function for opinions' update based on the best matching neighbor condition. The best condition is defined to be highest value of a given function on the closets neighbor for a given task. A formal proof shows that consensus can always be reached if the topology of the network does not change, proving the algorithm useful for coordination problems. This model is simulated over a random network of up to 2000 nodes.

In the case of dynamic networks the literature is more recent. Dynamic network and opinion processes have an impact of whether strong consensus can be reached. (Olfati-Saber 2007, Olfati-Saber, Fax, and Murray 2007, Olfati-Saber and Murray 2004) addressed the concept of reaching consensus on a dynamic network. In the case of switching networks (a dynamic graph parameterize by a switching signal) consensus is asymptotically reached

if the network is periodically connected (have a common spanning tree over time). Several issues are still presented, being the most important: (i) that the consensus reached in dynamic networks is not equal to the average consensus results of static networks, (ii) the connectivity notion for reaching consensus could be too restrictive in terms of the time (that it may take a consensus to be attained). A direct application to social networks is proposed from these researches using a variation of the mutation model that lies on the biology replicator-imitator dynamics. This analysis studies opinion dynamics through social choices. In this model, under a starting fixed set of behaviors represented by each node of the network, a vector of incentives for changing behavior per each node (represented by a weight matrix in the network), 4 potential states (flocking, cohesion, collapse and complete collapse) are possible. The characterization of the mutation matrix is given by its decomposition, based on the incentive matrix, the Laplace matrix of the graph and a fixed mutation parameter. Important implications and results of this theoretical model for social networks are: (i) the four states can appear depending in the mutation rate, (ii) a slow rate of mutation leads towards few dominant social trends. In line with consensus, this certifies that topology and variety of opinion matters across time.

Using a Kalman filter update, (Ren, Beard, and Kingston 2005) addresses the problem of nodes entering and leaving the network. In this model, the topology changes due to disconnections and failures in communication. The final result shows that consensus is delayed due to the fact that members keep entering and leaving the network. In addition,

the Kalman Filter rule is the best linear update rule if the agents do not actually know the true and final value of consensus.

(Iñiguez et al. 2009) proposes a dynamic random network for a discrete opinion process. The sense of rewiring in the network is used to join agents with similar opinions and to disconnect those who held a different one. Opinions are update for a period t, and then when reaching time T $(t < T)$ the network is updated based on the rewiring process. Using this model, it is shown that rewiring connectivity only with neighbors that share the same opinions leads to clusters and consensus only among the members of the cluster.

(Lanchier 2012) formalizes the results of the general case of the Axelrod model. A network based on N vertex which represents N agents with a vector of F cultural (ideas) features and potential S states converges almost surely to a monocultural state if F<=s. In the language of consensus, F cultural/idea groups are formed, where the vertexes on each one share most or all the characteristics of its neighbors. Notice that under this model, vertexes do not only agree or disagree on features, but adopt new position –states- in the network. An interesting result of this proof is also that if $F > s$ fragmentation among all possible features will occur.

The literature on opinions dynamics based most of its models on undirected networks. In the case of (Hegselmann and Krause 2002) since the network structure itself is the opinion structure, directed and undirected graphs may be formed while opinion evolution. In this model, if the bounded confidence interval between two opinions is set symmetric, the network will be undirected. On the other hand, if the bounded confidence interval is defines in an asymmetric manner, directed networks will arise. However, since the network changes directly with opinion the structure of the network itself plays no role. (Sousa 2005)

proposes a model for evolution of opinion and network formation. However, the network still is undirected. In addition, (Olfati-Saber, Fax, and Murray 2007, Ren, Beard, and Atkins 2005) briefly comment that directionality of networks will not affect consensus as long as the common spanning tree condition is met in the network.

In the case of the economic models, (Jackson and Rogers 2007) argue that since homophily is the characteristic that leads to network formation, directed networks may be the case. However, no empirical evidence is provided on how directionality of a network may change the rise and timing of consensus. Even though (Yildiz et al. 2011a) do not directly mention directionality of edges in a network, this issue is present when opinions fluctuate due to a stubborn agent. In this case, this agent's opinion does affect the rest of the opinions while his opinion is never affected. Consensus is not reached if this is the case. Unfortunately, only the provided evidence addresses this extreme case.

The latest developments of diffusion and opinion by (Jackson and López-Pintado 2013, Jalili 2013, Lee et al. 2013, Li, Scaglione, et al. 2013, Li, Braunstein, et al. 2013, Molavi et al. 2013) still leave an open question on the effect of having a directed network on the evolution and formation of opinion.

The majority of the literature on opinion dynamics deals with theoretical models and simulations. Just a few cases from psychology and sociology have developed in a close group environment (Nowak, Szamrej, and Latané 1990) . In those cases, consensus has always been reached not only due to the small network of the group, but due to the structure. In addition, as (Acemoglu, Dahleh, and Lobel 2009, Yildiz et al. 2011a) have

shown there are cases, as the stubborn agents, were regardless of the group-network- size consensus cannot be reached.

From the cluster formation and membership grouping, the research on opinion dynamics is richer. In the case of the of the bounded confidence model, once a cluster is formed this group does not change in time. In addition, the number of clusters will depend on the size of the confidence interval chosen to equate for differences among opinions (Hegselmann and Krause 2002). (Ren, Beard, and Atkins 2005) points out that the coming and going of members in a network has implications and delays for consensus. The stubborn agent model shows that influential agents create clusters or groups of opinions around them (Acemoglu, Dahleh, and Lobel 2009, Yildiz et al. 2011a). Finally, modelling diffusion of opinions in multilayer networks may lead to cluster formation in opinion (Li, Scaglione, et al. 2013, Li, Braunstein, et al. 2013).

The findings of researchers studying the opinion dynamics over networks is summarized in Table 1.

Table 1 Opinion Dynamics literature (1990-2013)

Reference	Network type	Opinion definition	Update method	Important result/characteristics
Nowak et al. (1990) [1]	static (a)	discrete - single attribute	rule 1	The strength of opinions decreases with less interaction and more distance between agents
Banerjee (1992) [1]	static (a)	discrete-single attribute	rule 3	In sequential games opinions are linked to learning since they are the based for decision making
Lewenstein et al. (1992) [2]	static (a,b,c,d)	discrete - single attribute	rule 1	In an Ising model, ideas spread faster if nodes can easily change their current opinion position
Nowak et al. (1996) [1]	static (e)	discrete - single attribute	rule 1	Consensus and influence decreases with geographical distance in the grid
Galam et al. (1997) [1]	static (e)	discrete - single attribute	rule 1	Opinions may differ based on close neighbors
Deffuant et al. (2000) [1]	static (a)	continuous - single attribute and multiple	rule 1	In a voter's model, interaction with all neighbors but one at a time leads to consensus.
Kacperski (2000) [2]	static (a)	continuous - single attribute	rule 2	There is the presence of a universal phase transition when the agents reach consensus
Schweitzer (2000) [1]	static (e)	discrete - single attribute	Other	An agent may move to minimize opinion pressure.
Sznadj et al. (2000) [1]	static (e)	discrete - single attribute	rule 1	There is a chain effect for adoption of new ideas for agents in a network
Stocker et al. (2001) [2]	static (f)	discrete - single attribute	rule 1	Contagion is assured by connectivity in a network.
Stocker et al. (2002) [2]	static(g,h,i)	discrete - 3 choices	rule 1	Under a complete, random and free scale networks, the opinion process is stable and reaches consensus.
Bernardes et al. (2002) [1]	static(i)	discrete - single attribute	rule 1	Opinion in a consensus status only fluctuate if the opinion regime (agents and influence) changes.
Elgazzar (2002) [2]	static (j)	discrete - single attribute	rule 1	Similar conclusions to Sznadj et al. (2000) but under a different network topology
Hegselman et al. (2002) [2]	static (a)	continuous - single attribute	rule 2	The author proposes an opinion model based on threshold, and that not always reaches consensus
Deffuant et al. (2002) [2]	static (a)	continuous - single attribute	rule 1	Consensus is always reached if an extreme opinion influences the rest. If communication occurs only with nodes of similar opinion, the results follow the bounded confidence interval consensus formation.
Weisbuch (2002) [2,*]	static (e)	continuous - multi attribute	rule 2	Local interaction matters the most for reaching consensus.
Behera (2003) [1,*]	static (e)	discrete - single attribute	rule 1	If opinions are bias or not true, the consensus value will depart from truth.

Reference	Network type	Opinion definition	Update method	Important result/characteristics
Sobkowicz (2003) [2]	static (e)	discrete - strong leaders	rule 1	Strategies define how each agent uses finite information to convince another agent
Stauffer et al. (2003) [2]	static (e)	continuous - single attribute	rule 2	Opinion process has a similar evolution on random and scale free networks
Deffuant et al. (2004) [2]	static (a)	continuous - single attribute	rule 2	Special extreme agents are radical cases of networks with agents where opinions will not reach consensus.
Fortunato (2004) [2]	static (d,g)	continuous - single attribute	rule 2	Consensus is always reached in network topologies with connections among all their agents
Fortunato (2004) [2]	static (a,i)	continuous - single attribute	rule 1	Under continuous opinions and a bounded confidence interval model, if all the agents have the mean opinion as the initial value, consensus will be reached. If the interval for acceptance of an opinion is small, opinion clusters start to appear.
Olfati-Saber et al. (2004) [1]	static and dynamic	continuous - single attribute	rule 2	Consensus is always reached if the network is statically or dynamically always linked by a spanning tree
Schulze (2004) [2]	static (e)	discrete - single attribute with n options	rule 1	Advertising and external information introduces external factors that make the opinion of the agents shift
Sousa (2004) [2]	static (i)	Discrete	rule 1	The evolution of opinion changes when agents form triangles in the network
Sousa (2005) [2,*]	dynamic - static (i)	continuous - single attribute	rule 2	If the network grows at the same time as opinions are adjusted, consensus is reached if the new connections are directly attached to the network
Stauffer et al. (2004) [2, *]	static (h)	continuous - single attribute	rule 1	Advertising impact the opinion of agents in a network
Weisbuch (2004) [2,*]	static (i)	discrete - single attribute	rule 2	Consensus is always reach under a bounded confidence interval with low bounds
Caruso et al. (2005) [2,*]	static (a)	discrete - single attribute + fixed opinion agents	rule 1	Agents that form a coalition can influence opinion
Fortunato (2005) [2,*]	static (i)	continuous - single attribute	rule 2	Consensus can be delayed by agents with extreme opinions
Pluchino et al. (2005) [2]	static (a)	continuous - single attribute	rule 2	When differences are stressed in initial opinion profiles, opinions vary over time
Ren et al. (2005) [1]	static and dynamic	continuous - single attribute	rule 2	Survey of consensus problems in Multi-Agent coordination. Consensus requires the existence of a spanning tree.

Reference	Network type	Opinion definition	Update method	Important result/characteristics
Ren et al. (2005) [1]	static	continuous - single attribute	rule 2	When all the agents can communicate to each other, if the consensus value is unknown in the systems, and all the agents are required to reach it; the usage of a Kalman filter rule for opinion update help the agents to uncover the hidden consensus state faster.
Weisbuch et al. (2005) [2,*]	static (i)	discrete - single attribute	rule 2	A group of agents have a different adoption threshold based on their position (degree) on the network. Nonetheless, the result shows that this hardly matters and consensus is only contingent on whether the confidence bounds are small or large. In addition, there is no big difference between the dynamics of a complete network and a free scales.
Ehrhardt et al. (2006) [1]	dynamic(g)	continuous - single attribute	rule 3	In a game theory framework where agents spread knowledge among them and agents create connections among them based on this knowledge, only a unique stationary state is possible if the network is sparse and agents do not spread knowledge at a fast pace. As soon as the spreading rate increases, a chaotic dynamic of knowledge and link creation appears.
Fortunato et al. (2007) [2]	static (c)	continuous - single attribute/unidirectional	rule 1	Opinion dynamcis help the authors to discover the main features of an actual election results by using a `word-of-mouth' model
Gil et al. (2007) [2]	dynamic (e,g)	discrete - single attribute but links of network are cut due to update	rule 1	Opinion formation into clusters appear because of the tides of the confidence bounds of an agent.
Lorenz (2007) [1,*]	static (a)	continuous - single and multiple attribute	rule 2	Comparison between agent based models and density models (which can be interpreted as limit case for infinitely many agents) is developed. Connectivity remains as the biggest factor for consensus.
Nardini et al. (2007) [1,*]	dynamic (g)	continuous - single attribute	rule 2	Rewiring can lead to consensus or break the interaction between groups leading to non-consensus states.
Olfati-Saber et al. (2007) [1,*]	static and dynamic	continuous - single attribute	rule 2	High connectivity speeds consensus. A mathematical unified framework for the analysis of linear models and their dynamics towards consensus is presented, for continuous-time and discrete-time systems.
Olfati-Saber (2007) [1,*]	Static	discrete - single attribute	rule 3	Opinion diversity affects the existence of a unified consensus among society or even inside a small group of agents

Reference	Network type	Opinion definition	Update method	Important result/characteristics
Jackson et al. (2007) [1]	static (g)	discrete - single attribute	rule 3	Incentives –or payoffs- matter when a belief or idea needs to be propagated
Benczik et al. (2008) [2]	dynamic (g)	discrete - single attribute	rule 1	Before consensus is reached, two different meta-stable states can persist for exponentially long times
Suo et al. (2008) [1]	static (a,g,h,i)	continuous - single attribute	rule 3	The authors use a deterministic utility and payoff function for each agent to model opinions. The agent may or may not decide to give his opinion every time they interact. The main findings show that the public opinion varies from community to community due to the degree of impressionability (willingness to accept someone else's opinion) of the agents .In networks where agents accept opinions based on future rewards, it is misleading to predict results merely based on the characteristic path length of networks.
Apolloni et al. (2009) [2]	dynamic (l)	continuous - single attribute	rule 3	Teenagers play a significant role in spreading information through a community rapidly, mainly through interactions in schools and recreational activities.
Iñiguez et al. (2009) [2]	dynamic(g)	discrete - single attribute	rule 1	An important feature of opinion-network coevolution is the separation of the two basic time scales, the rapid dynamics of opinion, and the slow dynamics of the network rewiring.
Martins et al. (2009) [2]	Static (a)	discrete - single attribute	rule 1	A comparison of different set of initial adopters is performed. The initial state hardly matters for consensus if they are cluster or well connected.
Chen et al. (2010) [2,*]	static (a)	continuous - single attribute	rule 2	Using the spread of an opinion, one can find the influential nodes that initiate viral propagation.
Malarz et al. (2010) [2,*]	static (a)	continuous - single attribute	rule 2	Strong and continuous interaction leads to consensus
Acemoglu et al. (2010) [1,*]	static (a)	continuous - single attribute	rule 2	In a network where no agent is disconnected, opinions converge even in the existence of disagreement and fluctuations
Yildiz et al. (2011) [1,*]	static (a)	discrete - single attribute	rule 1	In social networks where opinions are binary and agents exchange their opinion based on adoption thresholds, stubborn agents (someone that influence the rest but cannot be influenced) prevent consensus from happening. In addition, a stubborn agent can be strategically place to change the opinion of a specific group towards his opinion.

Reference	Network type	Opinion definition	Update method	Important result/characteristics
Xie et al. (2011) [1,*]	static (a,g,i)	continuous - single attribute	rule 1	In a network where only two opinions (0 or 1) are possible, and the consensus value has been reached, this fact can only change if the network topology changes and a large number of agents become committed agents (agents that exert influence but cannot be influenced) with an opposite opinion. This result is consistent for random and free scale graphs. The opinion process towards the new consensus value shows exponentially asymptotic behavior.
Lanchier (2012) [1]	dynamic (e)	discrete - 3 choices	rule 1	Multiple attributes when updating an opinion can prevent a society from reaching consensus
Zollman (2012) [2]	Static	discrete	rule 1	Consensus concept is too restrictive
Jalili (2012) [2]	static (i,h)	continuous - single attribute	rule 1 and 2	In free scale and random network with agents that only can communicate to neighbors with a similar opinion, the time to reach consensus is greater than in a network with full communication among agents.
Li et al. (2012) [2,*]	Static (m)	continuous - single attribute	rule 2	Two different opinions can coexist in a stable relationship for a given threshold
Singh et al. (2012) [2,*]	dynamic (g)	continuous - single attribute	rule 2	Introduction of committed agents with the same opinion makes the opinion of other agents reach consensus in dynamic networks
Askari et al. (2013) [2,*]	static (a,g)	continuous - single attribute	rule 2	An algorithm for solving large scale optimization problems through consensus is developed
Lee (2013) [2,*]	small word (j)	discrete - single attribute	rule 1	Simulation results show that even though the network is sparse -Korean market for notebooks - diffusion happens very early
Jackson et al. (2013) [2,*]	static (i)	continuous - single attribute	rule 3	Homophility in a network is associated to connectivity with similar nodes. This process can speed up consensus , even when starting with a small portion of adopters
Li et al. (2013) [2,*]	static (a)	continuous - single attribute	rule 2	Strategic interaction can create opinion clusters
Molavi et al. (2013) [2,*]	static – complete	continuous - single attribute	rule 2	When the true state of a process is unknown to all the agents, consensus is reached but not at the true state

Subscript [1,2]: [1] Only theoretical results are presented , [2] Theoretical and simulation/ real life case results are presented

Subscript [*,] : No subscript the author uses only the strong concept of consensus, [*] The author uses the concepts of strong and weak consensus

Network types: (a) complete, (b) sparse, (c) hierarchical, (d) lattice, (e) grid, (f) Boolean, (g) random, (h) Watts strogatz, (i) Free Scale, (j) small world, (k) multilayer, (i) real data, (m) parallel

Rules: (rule 1) threshold method, (rule 2) linear update method, (rule 3) game theory

DS Theory and recent developments in Opinion Dynamics

In opinion dynamics, a different theoretical background and modelling perspective is given by Dempster-Shafer theory. In their seminal work (Dempster 1967) proposes the existence of upper and lower probability bounds for disjoint subsets S through multivalue mapping. In cases where it is not possible to actually know the true probability distribution of an event, a researcher can approximate it by using this lower an upper probability sets. The true probability will lie in between. In this context, this multiple mapping can be used to combine information from different sources. The authors acknowledge that the main assumption for information fusion is that each source of information needs to be independent from the other. Opinions of different people based on overlapping experiences does not comply with this requirement. In this sense, if the researcher wishes to aggregate the subjective probabilities from two experts about the functionality (failure rate, packing rate, etc.) of a machine, the experts must not be related (connected) and their criteria needs to come from observations of different equipment.

(Shafer 1976) extends this theory and formalizes the rules for aggregating information. Three important elements constitutes the pillars of this theory:

(i) Basic probability assignment (BP): it is denoted as m and defines the mapping of the power set to $[0,1]$. The mapping m(A) represents the proportion of relevant and available evidence that supports the claim that a particular element of the universal set X belongs to the set A. BP complies with $m: P(X) \rightarrow [0,1], m(\emptyset) = 0, \sum_{A \in P(X)} m(A) = 1$.

(ii) Belief: it is the lower probability bound defined as the sum of all BPs of the proper subset B of the set of interest A such that $B \subseteq A$. It is represented by $(A) = \sum_{B|B\subseteq A} m(B)$.

(iii) Plausibility: it is the sum of all the BAs of the set B that intersect the set of interest A. It is represented by $Pl(A) = \sum_{B|B\cap A\neq\emptyset} m(B)$

This formalization is known as Dempster-Shafer theory. The following relations holds for these elements:

$$m(A) = \sum_{B|B\subseteq A}(-1)^{|A-B|} Bel(B) \quad (5)$$

$$Pl(A) = 1 - Bel(\bar{A}) \quad (6)$$

The classical probability of an event (usual probability definition) lies within the lower and upper interval. Furthermore, it is uniquely identified if $Bel(A) = P(A) = Pl(A)$.

As noted by (Dempster 1967) independence of the information sources and of the phenomena under observation is required for combination of evidence. The aggregation of evidence m_1 and m_2 is given by the rule:

$$m_{1,2} = \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-\sum_{B\cap C=\emptyset} m_1(B)m_2(C)} \quad for\ A \neq \emptyset \quad (7)$$

The first applications of this aggregation theory are found in fusion of data from sensors to analyze failure time of machines or components of machines. This theory has been applied in in decision sciences, reliability analysis and data fusion. In addition, different mixing of the BAs using products, re-scaling or a different definitions have also been proposed (Sentz and Ferson 2002). The results in most cases are the same as DS theory.

DS theory has been large criticized by (Pearl 1988, 1990, Zadeh 1986). The main problems can be narrowed down to two key facts:

(i) It is unnatural to think that any probability space cannot be fully characterize, then when any subjective probability does not sum to one, the space can be easily re-arrange and made convex and complete.

(ii) The aggregation rule produces counterintuitive results in many contexts. As an example (Zadeh 1986) points out the two physician's expert problem. In this setting, two doctors are asked for their opinion regarding a patient's neurological symptoms. Doctor number one believes that the patient has either meningitis with a probability of 0.99 or a brain tumor with a probability of 0.01. Doctor two believes the patient suffered a concussion with a probability of 0.99 , and the possibility of a brain tumor has only a probability of 0.01. DS calculations of $m(brain\ tumor) = Bel(brain\ tumor) = 1$ yielding a result that implies complete support for a diagnosis that both experts considered to be very unlikely.

In the present research we opt for presenting the reader the DS theory background, but we choose not to use this framework. However, we can point out the following key contributions to the opinion dynamics literature based on DS Theory:

(i) The authors (Wickramarathne et al. 2014) study the convergence of belief update over time in a system where humans (soft sensors) and hard (physical-based sensors) communicate and exchange information. The idea of consensus comes from the notion that agents need to communicate between them to estimate some phenomenon of interest without global coordination. When

defining consensus, the authors introduce the term rational consensus to refer to a final protocol where the final fixed point value is closed to the ground truth. In a network, agents exchange opinions based on a conditional update equation rule (CUE) where they fused their beliefs as a convex combination of conditionals of the events (opinions) that are being modelled (Premaratne et al. 2009). The authors provide theoretical results for the existence of consensus under their update protocol using paracontracting theory. In addition a computational analysis is conducted using networks of size 100 for the topology structures complete, scale-free, random, and small word. The results show that all the agents converge to a single fixed point as their beliefs are updated.

(ii) DS theory, CUE and bounded confidence interval framework are used to model agents exchanging opinions over a social network and study the role of influential agents (Dabarera et al. 2016). The consensus concept requires all the elements of the state set to be equal as time goes to infinity (as the interaction progresses in time). In addition, the idea of opinion clusters is discussed as a subset of agents that reach agreement. A similar concept has been previously discussed and introduced by (Deffuant, Amblard, and Weisbuch 2004, Sobkowicz 2009, Li, Braunstein, et al. 2013). The results show that consensus is formed when the number of opinion leaders is no more than one. In addition, the existence of consensus is also determined by the size of the bound.

For more detailed explanation on opinion dynamics modelling approach using this framework as the core of the opinion process update we refer the reader to the works of

(Dabarera et al. 2016, Premaratne et al. 2009, Wickramarathne et al. 2010, Wickramarathne et al. 2014).

Social Judgment theory and Pool Opinions theory

A different theoretical support for modelling opinions can be found in the concepts of human judgement and how this judgements can be synthetized using meaningful math model. In this context, Social Judgement theory (SJT) started in 1960s as a way of understanding human judgement. (Hammond and Summers 1972) understands the decision making process as complex dynamic between the decision maker (DM) and her environment. The information that an individual has and perceives from the environment is probabilistic. In this context, there is uncertainty not only of the state of the nature or environment; there is also uncertainty within an individual. Then, the main question is how individuals use cue information to produce judgmental responses. In this line, the authors suggest that a DM can use his own information and the information from his environment using additive compensatory forms, multiplicative forms and other more complex forms. In addition, the authors show how multiple regression can be used to separate and infer the weights that a decision maker places in cues.

SJT studies human behavior through a 6 step design system:

(vi) Conceptualize the judgmental problem

(vii) Understand the conditions and circumstances of the environment

(viii) Identify the relevant cues

(ix) Sample a profile of cues from the individuals present in the environment

(x) Obtain the judgement

(xi) Capture the judgmental policy of each sampled judge

(xii) Compare policies and make a final judgment

The research of (Cohen et al. 1983, Cooksey 1996, Tabachnick and Fidell 1989) proposes that these steps are mainly analyzed and summarized by the multiple regression linear model, given by:

$$Y_s = a_0 + a_1 X_1 + \cdots + a_k X_k + e_s \quad (8)$$

Where Ys is the judgement value of the DM as a function of each cue of the environment (information from other DM and external information). In this case, a_i captures the amount by which the judgment value is affect by every single cue, holding the other information constant. In many cases, the authors suggest that a standardization of the weights can lead to a model where the weights reflect only the importance of the information available to the DM.

The context of SJT can be directly adapted and applied to the aggregation of opinions concept through a linear pool (DeGroot 1974, DeGroot and Mortera 1991, Stone 1961). Eq. (8) in the case of a DM can be explained as the process of aggregating information from contacts that are part of a social environment relevant to a specific problem (i.e. social network). In addition, the linear pool can be directed interpreted as an upper bound solution of the Savage minimax problem of choosing an action while minimizing the regret. In the case of an optimal aggregation, the weights a_i produce the minimax solution for the DM (Stone 1961). If this is not the case, the linear pool still provides a strong framework to express how individuals (that may not be acting optimality) aggregate information.

Based on this foundation, (DeGroot 1974) proposes that an opinion of a DM aggregating information can be viewed as a continuous probability function. The linear pool aggregates different pdfs from the available information of other experts. This pdf expresses the certainty of uncertainty that each expert has on a specific unknown parameter. This parameter governs the distributions of the experts, and ultimately is the focus of attention of the DM. (DeGroot and Mortera 1991) show that the linear opinion pool can also be seen as a weight of summarizing a set of posterior distributions from experts, such that the optimal aggregation weights are given by solving an expected value quadratic loss functions. Based on these results, the authors suggest that a valid mathematical assumption for modeling the aggregation of information is assuming that the experts' opinions are normally distributed. This makes the model trackable and straight forward to work with. It is worth noticing that the authors use the opinion pool to obtain a final value for decision making purposes.

In later theoretical works, (Nakata 2003) shows that the linear update pool is a consistent way of aggregating continuous opinions, so a DM can infer the state of public opinion on any specific topic. The DM changes the weights based on the rewards that she gets from updating and sticking to a specific opinion. (Chambers 2003) argues that the opinion pool and its associated weights represent a subjective ordinal scale of each DM. Finally, (Budescu and Chen 2014) show that a linear pool of opinions in the case of experts is using the "wisdom of crowds" to produce a better and improved opinion from the external information available to a DM. In this case, the opinion of experts directly connected can be deleted from the linear pool since they contribute nothing to the final aggregation value.

As a final comment, in all the linear pool literature, this aggregation rule assumes that an optimal set of weights need to be found so a final optimal opinion can be produced. Certainly, this fact is not necessarily a given truth when modelling opinions in a social network. In this context, each agent can be allowed to hold his own probability distribution reflecting his own opinion; the opinion of ach agent can be updated by an opinion pool given its theoretical and mathematical consistency; but the influence weights related to the opinion of each agent are not necessarily optimal. Ultimately, this weights will reflect the degree of influence that each agent perceives from direct peers and the environment. This individual aggregation may or may not be optimal.

Limitations of Extant Research

Some of the key limitations in the current work in opinion modeling are:

- *Scalability:* Researches generally have focused on small networks and stylized models to derive theoretical conditions and insights rather than the operationalization and analysis of opinion dynamics at a large scale level.
- *Extent of agreement:* consensus is generally modeled as a fixed point common to all or a group of agents. When no local or global agreement is reached, there is no possibility of finding a common region of agreement (or disagreement) among agents.
- Observability: Extent literature generally assumes that the underlying network and/or the influence structure is known. It also assumes that the behavior of all agents can be observed.

- *Identification:* A practical issue is the lack of insights on how the "unobserved" opinions could be estimated based on observed behavior. Further, the nuances of the estimation problem when the influence network is weighted or unobserved are mostly unknown.

Conclusions and Research Propositions for the Next Chapters

Based on the key limitations of the opinion dynamics' literature, this dissertation addresses the four outlined issues by making the following choices:

- We focus on opinion dynamics for mid-size and large scale networks.
- We model the opinion of each agent as a unique stochastic process, thus each agents has a probability density function that accounts for her opinion at each time. Agents are embedded in a social network and update their opinion through a functional linear opinion pool only using information from neighbor agents. In this process, agents use weights to account for the importance and influence that other agents have over them. This process is justified by Social Judgement Theory; and the aggregation process has its theoretical roots on the linear pool literature (DeGroot 1974, DeGroot and Mortera 1991, Stone 1961).
- We redefine consensus as the overlapping region between opinion distributions. In this sense, we can find common local or global regions of agreement. The mathematical support for this concept is built from the basis of functional hereditary systems (Paternoster and Shaikhet 2000, Shaikhet 1996).
- Since opinions and influence weights are unobserved, when modelling opinions as probability distributions, we face the problem of estimating these arguments. M-Estimation theory and online particle filters can be used to estimate the

parameters of a functional state space model representing the mean stochastic opinion dynamics of the agents (Anderson 1989, Carvalho, Johannes, et al. 2010, Lopes and Carvalho 2013, Muthén 2002, Skrondal and Rabe-Hesketh 2004, Vaswani 2008b, Wooldridge 2005).

Chapter 3: Efficient Simulation and Analysis of Mid-Sized Networks

Overview

There is growing interest in the emulation and analysis of large social and information networks such as the Internet, Facebook and Twitter. However, the sizes of the networks and the computational time required to estimate quantities such as the average path length is growing at a faster rate than the computational power of modern day PCs. In this chapter, we presents a computationally efficient network representation and analysis approach to generate and analyze mid-sized networks (networks of sizes ranging from about 50,000 to 5,000,000 nodes) on regular PCs with RAM as low as 2GB using the open source R platform. The proposed approach combines an efficient network representation with efficient programming constructs such as vectorization and multi-core processing to yield a scale-up of about 400 and a speed-up of about 20. According to these scale-up and speed-up figures, personal computers (PCs) that previously could barely handle the simulation and analysis of a network of 10,000 nodes could now support assessments of networks with about 5,000,000 nodes. The speed-up and scale-up enable the simulation and analysis of mid-sized networks on regular PCs and eliminates the need for researchers to compromise on the scope, depth, and scale of their studies.

We use R language[3] to develop our computational approach; R is popular with researchers and provides an easy and efficient way to integrate a wide variety of statistical packages (including many pre-built network/graph analysis libraries) with our algorithms. Although compiled languages such as C, C++, and Java could lead to higher computational efficiencies, we believe the research learning curve associated with the use of an R-based

[3] R provides an environment for statistical computing; see https://www.r-project.org/.

approach is less steep. Furthermore, though some R packages already exist for network simulation and analysis (see the background section), they do not necessarily enable the study of large networks with limited RAM capacity.

In the next section, we present a brief review of literature and highlight some current trends with regard to network theoretical analysis in management science and epidemiology. We also discuss why the use of subgraphs of sizes less than about 50,000 may not be advisable for studies related to diffusion over social and information networks, and we highlight the benefits of modeling and analyzing mid-sized networks. In Section 3, we focus on the performance analysis of the network representation and the R-language constructs we integrate to simulate and analyze mid-sized networks on PCs. Section 4 contains the three sampling-based algorithms used to estimate six key network metrics, along with comparisons of their performance against existing algorithms. The conclusions and suggestions for research and development paths are in Section 5. Although our discourse is influenced by observations of the simulation and analysis of social and information networks in management science and epidemiology, our approaches to gaining computational efficiencies and estimating network metrics are agnostic regarding the type of network under consideration. Further, though our method scales-up with the size of the core, the increase in the number of cores only leads to speed-up. We limit our analysis to mid-sized networks as larger networks would require graph partitioning on a 2GB core PC, and graph partitioning and its impact has not been studied in this chapter.

Background

Simulation-based studies use large networks for two main reasons: (1) to be able to emulate the behavior of a real system at faster rates so that various scenarios and/or evolutionary behaviors can be studied (Nicol et al. 2003, Yeom et al. 2014) and (2) to develop and test algorithms related to navigation (Chen, Wang, and Wang 2010, Liu et al. 2014), clustering and partitioning (Handcock, Raftery, and Tantrum 2007), community detection (Bickel and Chen 2009, Karrer and Newman 2011, Newman 2006, 2013), or distributed computing (Fujimoto et al. 2003) on networks. As some of the researchers listed in Table 1 point out:

- The topologies of large-scale networks may differ significantly from those of smaller systems (especially subgraphs) that serve as surrogates, and the true values of network-centric metrics, such as degree distribution and average path length, often are not preserved in subgraphs (Ebbes, Huang, and Rangaswamy 2013, Lee, Kim, and Jeong 2006a, Newman and Watts 1999).
- Dynamics (of diffusion or evolution of behavior) over large-scale networks may evolve in a different way than the dynamics over small-size networks, which in turn could influence the inferences made from the simulation analysis (Bhatt et al. 1998, Yeom et al. 2014).

Both factors can lead to an incorrect attribution of the impact of various topological features on network processes (Dong et al. 2015). Thus, there are important disincentives for simulating and analyzing small networks. (Section 2.3 explores issues related to the appropriate size of a subgraph.) However, as Table 2 demonstrates, the simulation and analysis of large networks is also not straightforward. The availability of large

computational resources (usually parallel processors) and the expertise to use it efficiently (using distributed computing and efficient manage memory[4]) are prerequisites.

Table 2 Key Large Studies Using Simulate Networks and Simulation on Networks

Author	Area of Study	Network Size	Processors/ Cores	Threads	Constructs Used for Scale-Up/Speed-Up	Key Findings
Bhatt et al. (1998)	Communication systems	1,000 nodes and 2,200 links	9	No		Modeling of 90,000 communication events in simulated network of size 1,000; computational gains increase up to a factor of N-processors
Fujimoto et al. (2003)	Parallel discrete event simulation communications network	4,000,000	1536	No	PDNS (Parallel Distributed Network Simulator) and GTNetS (scalable networks simulation in C++) synchronized	GTNetS found to be most scalable protocol for network simulation; actual high-performance computational capabilities and packages allow researchers to simulate millions of nodes systems in real time
Nicol et al. (2003)	Computer systems and simulation	4,000,000	20	No	Authors use C++ on scalable simulation framework	Abstraction reduces computational time by a factor of approximately 400 and parallelism by a factor of 20
Barrett et al. (2008)	Epidemiology and computer simulation	100,000,000 nodes	112	448	Grouping similar messages/task-work load balance	Development of computational tool EpiSimdemics; program provides fast simulation and information on network characteristics, subpopulations infected, and locations
Bisset et al. (2009)	Computer systems and simulation	500,000 nodes	224	No	Master–slave paradigm for calculations	Simulation cost decreased by reducing SEIR model to a sequence of graph operations; applicability of EpiFast ranges from health applications to social behavior studies
Chen et al. (2010)	Computer systems and marketing	650,000 nodes	1	4	-	Influence spread modeled 100%–260% faster than other algorithms because calculations restricted to local influential areas
Yeom et al. (2014)	Epidemiology and computer simulation	280,000,000	22,640	2	Detection mechanism for synchronization and load distribution	Development of EPISIMDEMICS allows researchers and policy makers to have a tool with high precision and speed to simulate large-scale systems and contagion within the system
Liu et al. (2014)	Computer science, parallel and distributed systems	11,300,000	4	8	Parallelization built to maximize split tasks while finding influence graph of problem	Solution to NP problem of influence maximization provided by: (1) bottom-up transversal algorithm sorted by level and degree and (2) adaptive k-level method to reorganize influence graph
Verma et al. (2015)	Algorithms and large scale graphs	18,500,000	2	8	Parallelization built to maximize split tasks while finding influence graph of problem	Authors propose scale reduction algorithm based on communities and cores to detect maximum cliques in graphs
SNAP (2008)	Simulation, analysis and storage of networks	240,000,000 nodes	Not available, depends on computer used		Not available	C++ and Python platform for analysis and simulation of large-scale systems

[4] The terms memory, primary memory, and random access memory (RAM) are used interchangeably.

Memory Management

It is imperative to manage available memory on a PC to process large graphs/networks. Efficient memory management for network analysis can be achieved through one or more of the following techniques: (1) choice of information representation, (2) use of efficient operational (primarily programming) constructs, and (3) efficient memory handling during intermediate processes/computations.

Information Representation: A network is a collection of nodes and links. The problem of information representation in a network involves its efficient storage and retrieval. In a network, nodes usually represent resources or economic agents. Of their various attributes, the state of its existence is key. This information can be stored as a vector or as a record/data frame. On the one hand, the memory requirements for representing the node information usually scale linearly with the number of nodes N. On the other hand, the number of links scale quadratically with N; there can be up to $(N-1)\times(N-1)$ links in a network of size N if the network is directed and $(N-1)\times(N-1)/2$ if it is undirected.

The existence of links, or the lack thereof, traditionally is represented by a $N\times N$ 0–1 "adjacency matrix" A, in which the element a_{ij} provides information about the existence of a link between nodes i and j. However, most large networks (especially social ones) are sparse, so allocating $N\times N$ memory units to store the link information can be inefficient. Significant efficiency gains in memory usage can be achieved by representing the adjacency matrix as a sparse matrix (Koenker and Ng 2011). This sparse matrix representation replaces the use of an $N\times N$ adjacency matrix with three arrays that store

information about nonzero matrix elements, column indices, and row indices[5]. Effective memory usage is significantly smaller than the number of memory units required to store information about the entire adjacency matrix. The author in (Butts 2008a) developed the *network* package in R for a sparse matrix–based representation of large and sparse networks; he reports a 99.8% reduction in memory requirements when an adjacency matrix of a network with 100,000 nodes and 100,000 links is represented as a sparse matrix.

An alternative to the adjacency matrix–based approach to storing network information is an egocentric representation of a network, wherein the focus is on individual nodes and their connections. Egocentric representation can be implemented in two ways: (1) using an array of size $N \times \overline{\overline{K}}$ (where $\overline{\overline{K}}$ is the maximum number of connections that any node can have in the network) or (2) by using a list with N rows, with each row having a variable size (i.e., variable number of columns) dictated by the exact number of connections for a specific node. If the average number of connections is $\overline{K}$, the typical storage requirement for the list scales as $N \times \overline{K}$. These two approaches to egocentric representation have significantly different memory requirements that depend on N and the ratio between $\overline{\overline{K}}$ and $\overline{K}$; the list is more efficient when the ratio between $\overline{\overline{K}}$ and $\overline{K}$ is larger than about 1.05 (an array of size $N \times \overline{\overline{K}}$ is usually more memory efficient than a list of size $N \times \overline{\overline{K}}$). We use the list-based egocentric representation of the network.

Operational Constructs: Most PCs today provide system-level features such as vector processors, multiple cores, and virtual memory to manage swapping and paging between primary and secondary memory to improve system performance. However, the onus is generally on the researcher to apply these performance enhancers. Two key operational

[5] https://cran.r-project.org/web/packages/SparseM/index.html

constructs that employ these system-level features and are relevant to network analysis are vectorization and multi-core processing.

- Vector Processors and Vectorization: Vectorization is the process of rewriting a repeated statement (usually a loop) on a vector of size N in such a way that the information is processed in batches of size M (where $M > 1$, and values of 2 or 4 are common) instead of being processed element-by-element. Thus, vectorization leads to processing the vector approximately $\lceil N/M \rceil$ times versus N times. This batch processing of operations can significantly speed up the network analysis, because repeated statements are often encountered during network generation and the study of diffusion over a network.

- Multi-core Processing: The author in (Reghbati and Corneil 1978) show that bounded parallelism—the concept of having a fixed number of available cores, each with one autonomous and individual task—is a natural way to process graph information, such as graph search and traversal. Modern PCs with multiple cores enable this approach. When used effectively, multi-core processors can run multiple instructions at the same time, leading to speed-ups that are similar to those achieved using multiple processors/servers.

In this chapter, we demonstrate that the simultaneous use of vectorization and multi-core processing substantially improves the computation time required for network simulation and analysis.

Handling Memory during Processing: The use of compiled languages, efficient uses of dynamic versus static memory, and freeing resources after use during intermediate

processing are some standard approaches that enable the efficient management of memory during processing.

- Compiled languages: Compiled languages such as C/C++/Java usually consume fewer resources than high-level interpreted computer languages such as Matlab and R, because the compiled languages usually do not require a runtime interpretation (whereas high-level languages do). Furthermore, the use of compiled languages often increases the efficiency of memory usage, because it allows direct memory addressing and control. These languages also provide the flexibility to handle the memory allocated to temporary storage (during intermediate steps) and "garbage collection" at the end of a simulation run (Ihaka and Gentleman 1996). However, some high-level languages, such as R, provide access to some previously compiled functionality and language constructs that can have significant impacts on processing speeds[6].

- Static versus dynamic memory allocation: Static allocation refers to cases in which compilers automatically allocate memory for the variables. Dynamic allocation refers to cases in which users control the exact size of the memory allocated to a variable. There are pros and cons for each method; for example, static memory allocation is more common because of its ease of use, but it can be less efficient. Most low-level languages allow dynamic memory allocation; high-level languages are limited to using static memory allocation. However, some high-level languages, such as R and Python, circumvent the potential inefficiencies associated with static allocation by providing dynamic structures (such as list objects).

[6] http://adv-r.had.co.nz/memory.html

We use R with (1) a mix of pre-compiled functions such as those for vectorization, such that R is more efficient despite being a high-level language; (2) dynamic data structures for more efficient memory allocation; and (3) efficient controls of workspace memory by implementing "garbage collection" during both the end of a simulation run and intermediate steps.

Graph Sampling and Efficient Algorithms: The concept of reducing the computational burden associated with the analysis of data from a large network by using only a sample of the available data and drawing an inference about the population from it dates back to the 1950s. Over the years, three broad classes of approaches have evolved for network sampling: node sampling, link sampling, and subgraph sampling. Researchers address the benefits and limitations of these approaches and their variations in different situations (see Table 2 (Barrett et al. 2008, Bisset et al. 2009, Crovella et al. 2002, Csardi and Nepusz 2006, Frank 1979b, Goldenberg et al. 2010, Goodman 1961, Hunter et al. 2008, Karrer and Newman 2011, Klovdahl et al. 1977a, Verma, Buchanan, and Butenko 2015)).

Table 3 Research Describing Key Approaches of Network Sampling and Their Implications

Sampling	Reference	Method	Estimation	Pros	Cons	Application	Results
Node	Frank (1979)	Uniform random sample of nodes from ego node	Degree distribution	Does not guarantee other metrics	Density of network on nodes can be estimated	Sampling unknown social networks	Unbiased estimators and variance estimators on edge and vertex occurrences; empirical approximation provided
	Chandra-sekhar & Lewis (2011)	Optimization problem to minimize MSE	Any network metric or regression coefficient	Statistical corrections for bias	Requires selecting all links from selected set of nodes	Econometric estimation of contact nodes in villages	Parameters estimated consistently using graphical reconstruction, allowing network heterogeneity
	Eppstein and Wang (2004)	Hoeffding´s inequality-based probabilistic framework for bounds on error	Estimate centrality of all vertices	Defines sample error and confidence level directly	May be too extensive in large networks	None listed	Consistent estimator derived, recovering centrality measures with desired precision
Link	Lakhina et al. (2003)	Edge sampling	Shortest path (SP)	Efficient for SP calculations	Good for path length estimation	IP Protocols and routes	Only sample nodes for SP, but produces biases in degree distribution
	Riondato and Kornaropoulos (2014)	Randomly select a shortest path, collect all edges on this direction	Betweeness and k-path	Highest computational performance	Estimator with highest mean square error when compared to Eppstein (2004)	Real and simulated networks up to 80,000 nodes	Computational time gains in the order of 200-300% depending on network topology; sampling approach also provided based on Hoeffding´s inequality (1963)
Subgraph	Goodman (1961)	Snowball sampling	Explore structure and estimate degree	Consistent and best uniform estimator	Requires big sample size; network statistics may change	None listed	Strong theoretical method for network exploration and link construction derived by authors
	Klovdahl et al. (1977)	Random walk	Explore an unknown network	Requires smaller sample size than snowball sampling	Cannot sample nodes not reachable from initial node	Epidemiology and sociology	Method captures full structure of contact network
	Leskovec and Faloutsos (2006)	Forest fire (FF)	Sampling to reduce size of graphs and to match evolution of a graph	Forest fire performs best for network exploration	Single source start – disconnected areas may not be reached	Real network data used, with graphs up to size of 76,000 nodes	FF requires about 15% of network to be sampled to recover degree, weakly connected components, clustering coefficients, and eigenvalues distribution
	Ebbes et al. (2015)	FF method	Impact of sampling on network metrics	Subgraph sampling outperforms other methods	Small networks with common spanning tree	Sampling from networks of size up to 20,000	Sampling to recover local properties should use low burnt forest fire method
	Kourtellis et al. (2013)	Navigate network from a highly connected source	Recover SP, k-measure and betweennes s	Exploration based on random walk	Single source start – disconnected areas may not be reached	Real-life networks up to 82,000 nodes	Sampling only 20% of nodes in sampling procedures recovers proposed metrics

A common use of graph sampling is for the estimation/recovery of network metrics (e.g., average path lengths, clustering coefficients) from sampled information, with a desired level of accuracy. Extant research provides some empirical guidelines on this topic:

- Different sampling techniques work differently for different types of networks and network metrics; there is no one-size-fits-all algorithm. However, snowball sampling and the forest fire approach seem to work better than other techniques (Ebbes, Huang, and Rangaswamy 2013).
- Sample sizes need to be quite large for network metrics to have the desired accuracy. When a study is empirical, and the objective is simply to derive an accurate estimate of metrics such as the average path length, betweenness, and other centrality measures, (Lee, Kim, and Jeong 2006a) show that almost 60%–80% of the network needs to be recovered in the sample.

More recent research focuses on adding error bounds to estimated metrics (Bader and Madduri 2006, Kourtellis et al. 2013, Riondato and Kornaropoulos 2014, Wang 2006) and improving the efficiency of the sampling approaches by developing hybrid approaches (Ebbes, Huang, and Rangaswamy 2013). Accurate estimates can be derived with sample sizes as small as 15% of the population (Leskovec and Faloutsos 2006a) and approximate results can be obtained with sample sizes as small as 2%–5% of the population. Results from graph sampling indicate it is possible to recreate most of the key properties of a population network by using appropriately sampled subgraphs.

Our focus is not on determining whether the subgraph is representative of the population but rather on estimating graph metrics, such as the degree distribution and average path length of the given graph by using graph sampling, vectorization, and multi-core processing. We assume the population network or a representative subgraph is already available.

Network Sizes

Some social networks are intrinsically small (e.g., influential families in Medieval Florence, the boards of directors of *Fortune* 500 companies)(Jackson 2010); analyzing such networks poses relatively few computational challenges. However, most other social and information networks, especially those studied in marketing, epidemiology, and sociology domains, are large; their sizes often exceed several million nodes. Most simulation-based (and agent-based) studies that discuss the impact of the structure of a network on evolutionary behavior therefore implicitly refer to such networks. We infer that some researchers conduct their analyses on representative subgraphs that are significantly smaller than the population. Table 4 provides a summary of such studies in key marketing journals in recent years (2013–August 2015) (Anderson et al. 2013, Aral and Walker 2014, Bapna and Umyarov 2015, Chen, Chen, and Xiao 2013, Gelper and Stremersch 2014, Goel and Goldstein 2013, Goodreau et al. 2008, Haenlein and Libai 2013, Hu and Van den Bulte 2014, Iyengar, Van den Bulte, and Lee 2015, Libai, Muller, and Peres 2013, Lu, Jerath, and Singh 2013, Ma, Krishnan, and Montgomery 2014, Ma, Yang, and Mourali 2014, Miller and Mobarak 2014, Risselada, Verhoef, and Bijmolt 2014, Shriver, Nair, and Hofstetter 2013, Stephen, Zubcsek, and Goldenberg 2016, Toubia, Goldenberg, and Garcia 2014, Trusov, Rand, and Joshi 2013).

Table 4 Papers on Social Network Analysis and Diffusion of Innovation that Appeared in Key Marketing Journals (2013–2015)

Reference	Journal	Study Type	Study Scale	Network Type	Key Hypothesis	Key Findings
Goel et al. (2013)	Mgmt Sci.	Empirical/Simulation	Sample of Twitter; 25,000,000 individuals on simulated networks	Twitter (empirical) and scale-free networks (theoretical)	Proposes concept of "structural virality" that characterizes two types of online diffusion contents: one grows through viral mechanisms; other obtains popularity through single broadcast	Structural diversity characterizes online diffusion; structural virality is typically low; size of largest broadcast drives popularity and scale-free network fails to replicate existing diversity of structural virality
Bapna and Umyarov (2015)	Mgmt Sci.	Empirical	3,800,000 users with over 23 million friendship pairs	Largest connected component of last.fm network	Causal peer influence exists in general population of a large-scale online social network	Peer influence contributes more than 60% increase in probability of adoption; individuals with small social circle more likely to increase their adoption decision due to peer influence.
Ma et al. (2014)	Mgmt Sci.	Empirical	3,70,000 customers with 300 million phone calls (Events)	Asian mobile network	Impacts of latent homophily, social influence and exogenous factors identifiable	Latent homophily and social influence have strong impact on purchase timing and product choice
Toubia et al. (2014)	Mgmt Sci.	Methodology	398 consumers	Observed network structure based on 398 consumers	Using disaggregate-level data on social interactions improves forecasts of aggregate penetration	Parameters of extant diffusion models (mixed or asymmetric influence) may be estimated by social interactions data sampled from as few as one group of consumers in one time period
Aral and Walker (2014)	Mgmt Sci.	Empirical	1,30,000 peers	Online network made by adopters of a Facebook app	Peer influence in networks characterized by structural conditions	Structural embeddedness and tie strength increase peer influence; amount of physical interaction does not experience an effect
Lu et al.(2013)	Mgmt Sci.	Empirical	6,705 reviewers with 2,314 ties and 27,634 reviews	Epinions (Jan. 2002–Dec 2008)	Formation and emergence of opinion leaders (nodes with high in-degree) driven by both networked-based property and intrinsic property of a node	"Preferential attachment" effect and number and quality of opinions of a node increasingly impact in-link behaviors to it; intrinsic property has strong but short-term effect on adding inlinks, preferential attachment effect has smaller but long-term effect
Shriver et al. (2013)	Mgmt Sci.	Empirical	703 self-identified windsurfers; panel data with 57,040 observations	Privately-held community website,Soulrider.com, including 10,677 users (June 2011)	Online content-generation activity partially determined by social ties	Online content-generation activity enhances effect of social ties in local networks; increased content-generation activity and tie density contribute more visitation and browsing on corresponding site
Gelper and Stremersch (2014)	IJRM	Methodology/ Simulation	55 countries	N/A	Sparseness of data and large number of potentially influential country characteristics contribute to difficulty of identifying country characteristics that drive diffusion patterns	Economic wealth, education have strongest effects on diffusion
Mukherjee (2014)	IJRM	Simulation	Varying from 81 nodes to 10,680 nodes	8 real social networks	"Chilling" effect of network externalities on new product diffusion partially caused by other network characteristics	Increasing network size and average degree of node mitigates chilling effect of network externalities; increasing clustering accelerates diffusion speed. Chilling effect not inherently nested in diffusion model of Goldenberg et al. (2010); network externalities likely to slow down diffusion of innovation most of time, but not always
Miller and Mobarak (2015)	Mkt Sci.	Empirical	2,280 households in 42 villages, 2 districts	Real social network of 2 districts in Bangladesh	Influences of opinion leaders and social networks have different impacts on diffusion of nontraditional technologies	At first stage of adoption, external information and marketing campaigns promote initial adoption and experiential learning of new products; late-period adoptions require new technologies to match local preferences
Iyengar et al. (2015)	Mkt Sci.	Empirical	193 physicians	Group of physicians in San Francisco, Los Angeles and New York City	Peer influence may affect repeat behavior	(1) Peer contagion occurs in both trial and repeat; (2) most influential nodes vary in time periods; (3) Nodes who is most susceptible also varies in time periods. Informational social influence moderates risk in trial and normative social influence promotes conformity in repeat
Hu and Van den Bulte (2014)	Mkt Sci.	Empirical	8,259 academic scientists	Population of life scientists	Middle-status anxiety and conformity play key roles in adopting products potential adopters expect to boost their status	Status affects (1) time adoption behaviors that occur regardless of social influence; (2) how much social influence one is able to have; (3) how influential one is in inducing another's adoption.
Goel and Goldstein (2013)	Mkt Sci.	Empirical	Over 100,000,000 people	Communications network	Large-scale social data promotes prediction accuracy of behaviors of individuals and their acquaintances	Social data improve identifying behaviors of individuals but role of social data in prediction may be mitigated when transactional data available
Wang et al. (2013)	Mkt Sci.	Empirical	215 students	Group of students in same university	Adoption mechanism may vary from fashion versus technology-related products	Social interaction results in different behaviors in different product adoption processes; experts exert significant influences on technology-related products, common individuals exert comparable impacts on fashion-related products. Early decisions likely to be more influential than later decisions for technology-related products
Stephen et al. (2015)	JMR	Simulation	6 to 16 nodes; 70 members	ER graph and WS graph; a group of a large U.S. online panel	Network structures may affect innovativeness of person's product idea in ideation contexts	(1) High clustering impedes innovativeness of a customer's idea; (2) inspirations tend to be redundant when their sources are clustered; (3) high redundancy in inspirational ideas causes lower innovativeness and (4) effect moderated when individual does not depend on other individual's idea for inspiration
Trusov et al. (2013)	JMR	Simulation	1,000	Regular lattice, random, small-world and BA graphs	Systematic diffusion conditions are stable and transferrable to new diffusion processes	Such systematic conditions improve prelaunch forecasts; incorporation of Bayesian inference models and stochastic relationships in complex systems
Libai et al. (2013)	JMR	Simulation/empirical	161 to 10,680	Several	Speed and range of word-of-mouth seeding program may generate various social values	Factors such as competition, program targeting, profit decline, and retention affect expansion and acceleration of WOM
Chen et al. (2013)	JMR	Simulation	3,000	BA, WS power-cluster and Flickr	Both sampling method, topology of social network can contribute to accuracy of estimating consumers' social inter-correlation	Magnitude of social inter-correlations likely to be underestimated in sampling data, especially for scale-free networks
Risselada et al. (2014)	JM	Empirical	15,700	Random sample of customers of mobile telecommunications operator in Deutschland	Dynamics of social influence and direct marketing simultaneously impact adoption of high-tech products	Over time, effect of social influence from cumulative adoption is positive and decreases, influence of recent adoptions remains constant; effect of direct marketing always decreases
Ma et al. (2014)	JM	Simulation	86 to 1,200	Sample of U.S. and Japanese consumers; undergraduate students from same university; adults recruited from online panel	Independent and interdependent mindset of consumer may affect adoption decision of new products	(1) Consumers in predominantly independent culture prefer to adopt revolutionary innovation, consumers in interdependent culture more likely to adopt incremental innovations; (2) newness level of product and distinctiveness level of consumer simultaneously affect adoptions; (3) distinctiveness-dampening and distinctiveness-enhancing cues can reverse effect of independent and interdependent self-perspectives
Haenlein and Libai (2013)	JM	Simulation	Around 1,000	Artificial social network generator (Jackson and Rogers, 2007)	Targeting potential adopters with high value, instead of high power of influence, may improve adoption by increasing network assortativity	Distribution of lifetime value (CLV) in population and size of initial adopter key factors in determining which seeding approach is preferable, i.e., targeting opinion leaders or revenue leaders

The size of a subgraph can be significantly smaller than the population; extant research helps approximate the bounds of this subgraph size.

- Lower bound: It is important to select subgraphs of 50,000 or more when conducting diffusion studies over social networks (networks with average connectivity of about 100 and population size of more than 5), because
    - The choice of the size of the subgraph influences both the rate and the extent of diffusion; this influence does not scale linearly with the size of the subgraph, often leading to biases in the inferences (Dong et al. 2015).
    - Average path length and diameter of a network are key metrics that influence the information flow between nodes. These metrics stabilize only when the number of nodes in the subgraph is around 50,000 (Castro and Shaikh, 2015) (see Fig. 1).

These findings, along with the bounds that researchers place on sample size requirements for subgraphs to be representative (Ebbes, Huang, and Rangaswamy 2013, Leskovec and Faloutsos 2006a), lead us to conclude that simulation-based analysis of diffusion dynamics on subgraphs with less than 50,000 nodes may not be representative of how evolution takes place in populations of actual social and information networks.

- Upper bound: Although there are no constraints on the upper limits of the sizes of subgraphs, the growing body of research on subgraph sampling indicates:
    - Studies conducted on sufficiently large and representative subgraphs that capture some of the key metrics of the population network can also capture the nuances of the interaction between the structure of the

networks and the dynamics of the diffusion over them (Dong et al. 2015; (Ebbes, Huang, and Rangaswamy 2013)).

- As the network size grows, sample size requirements (i.e., percentage of the population sampled) for obtaining accurate estimates of key network metrics decreases (Castro and Shaikh, 2015).

These findings, in conjunction with research on sample size requirements for obtaining representative subgraphs (Kourtellis et al. 2013, Riondato and Kornaropoulos 2014, Wang 2006) lead us to conclude that simulation-based analysis on subgraphs with more than about 5,000,000 nodes will rarely be required, even when we analyze large populations (e.g., the current population of the United States).

Subgraphs in the range of about 50,000–5,000,000 nodes can be used to capture the influence that a gamut of social and information networks have on evolutionary behavior, especially diffusion dynamics. We refer to networks with nodes in this range as mid-sized networks and focus on them (networks smaller than about 50,000 nodes are small-sized; those greater than 5,000,000 nodes are large-sized). Here, the boundaries are guidelines only; it is not possible to precisely demarcate systems. The density of the links and the type of process studied strongly influence where these boundaries actually exist. Furthermore, our classification also reflects computational needs. Mid-sized networks do not require multiple processors; they can be studied on most modern-day PCs. Larger networks are likely to require high-performance computing.

Key Results

As described in the last section, we make the following choices in our analysis:

- We use the R programming platform for simulation and analysis. Several tools required by network analysts are already available in R. Some R packages and their key features are presented in Table 5 (Admiraal and Handcock 2008, Butts 2008a, b, Goodreau et al. 2008, Hunter et al. 2008, Stadtfeld 2013, Visser et al. 2015).

Table 5 Key R Packages Useful for Simulation and Analysis of Networks

Contributors	Package	Key Features
Hunter et al. (2008)	Ergm	• Primary use of R package is to fit exponential random graphs models through maximum likelihood using Monte Carlo simulation • Once model is estimated, MCMC algorithms used to simulate synthetic networks from a model • Also allows for comparison of simulated networks to original network
Butts (2008a)	Network	• Focus of package is creation and storage of large and sparse networks; authors document 99.8% reduction in memory required to store network information • Package can interact with SNA and igraph to use specific social analysis capabilities • Package, however, does not allow distributed computing over multiple cores
Admiraal and Handcock (2008)	Networksis	• Package enables simulation of bipartite graphs through sequential importance sampling; technique, in contrast with MCMC methods, provides more efficient method that requires only a few samples of the graph • Package also allows parallelization commands through the use of library snow in R
Butts et al. (2014)	networkDynamic	• Facilitates handling on temporal network data • Package has built-in capability to open Sonia software in R; interactive movies of the evolution processes of the networks can be constructed and played
Butts (2008b)	SNA	• Social Network Analysis; package offers comprehensive set of tools for graph simulation and analysis • Package features over 125 functions for manipulation and analysis of data but does not allow distributed computing and uses adjacency matrix-based representation of networks
Goodreau et al. (2008)	statnet	• One of most comprehensive network analysis packages in R; allows calculation of usual network statistics (in and out degree, number of nodes and edges, clustering coefficients), parameter estimation of exponential random graph models • Package also groups most of R network packages • Potential downfall of package is networks can only be specified using adjacency matrix structure; may limit use of package for large-scale networks
Csardi and Tamas (2006)	igraph	• Uses object approach in which network can be stored as a list or matrix; different types of networks can be simulated • Network metric calculation algorithms improved for faster computing times (when compared with current C and Python packages) • Key issue with package is when simulating a network, code has been programmed in matrix form; although final element can be an adjacency list of neighbors, full adjacency matrix is used for network generation
Stadtfeld (2013)	NetSim	• Package uses ideas of micro-data models and agent-based simulation to construct various models to explain dynamics and evolution of a social network –nodes and links addition and deletion • Author proposes series of models based on random exponential graph models; models estimated by simulated likelihood • Potential downfall of package is networks can only be specified using adjacency matrix structure; may limit use of package for large-scale networks

- We use an egocentric representation of the networks. Egocentric representation is more conducive to multi-core processing, especially when the

objective is not linked to clustering, search, or navigation on a network. Our focus is primarily on enabling the study of diffusion/evolutionary processes on a network

- We use dynamic memory allocation by using list objects to represent networks. The ratio of $\overline{\overline{K}}$ and $\overline{K}$ is usually high (>2 in most social and information networks that have a long tail), making the use of lists (rather than arrays) more efficient.

- We use vectorization and multi-core processing on a PC. The R platform offers the use of certain pre-compiled functions for vectorization to speed up calculations and packages such as *doParallel, foreach,* and *parallel* to distribute the computation across multiple cores; we apply these functions and packages.

- We propose and use algorithms that combine node sampling, vectorization, and multi-core processing for estimating network metrics. Our objective is to derive accurate estimates of network metrics from a given graph; the input graph is the population or a subgraph, either real or synthetic.

All computations and results presented herein were implemented on an INTEL® CORE™ i7-4710MQ notebook with 16GB of RAM (8 cores).

Impact of Efficient Representation

Table 6 contains the results that reveal how the storage (memory) requirements change when we shift from an adjacency matrix–based representation of a network to a sparse matrix–based representation to a list-based approach in R.

Table 6 Comparison of Storage Space Required to Store Link Information for a Directed Random Network with Average Connectivity of 100 Links

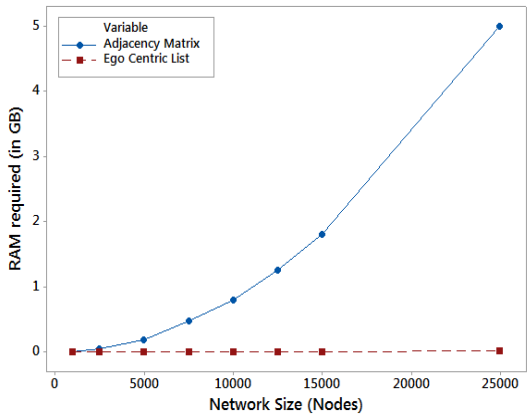
Network Size (Nodes)	Storage Required for Adjacency Approach (1)	Storage Required for Sparse Matrix Approach (2)	Storage Required for List-Based Approach		
			Storage	Reduction vs. (1)	Reduction vs. (2)
	GB	GB	GB	%	%
1.00*10E3	0.008	0.002	0.001	-88.96%	-63.08%
2.50*10E3	0.050	0.006	0.002	-95.56%	-63.07%
5.00*10E3	0.185	0.012	0.004	-97.78%	-63.06%
7.50*10E3	0.475	0.018	0.007	-98.52%	-63.06%
1.00*10E4	0.800	0.024	0.009	-98.89%	-63.06%
1.25*10E4	1.250	0.030	0.011	-99.11%	-63.06%
1.50*10E4	1.800	0.036	0.013	-99.26%	-63.06%
2.50*10E4	5.000	0.060	0.022	-99.56%	-63.06%
5.00*10E4	NA	0.120	0.044	-99.78%	-63.06%
1.00*10E5	NA	0.240	0.089	-99.89%	-63.06%
5.00*10E5	NA	1.202	0.444	-99.98%	-63.06%
1.00*10E6	NA	2.404	0.888	-99.99%	-63.06%

We were not successful in simulating and storing link information about networks of size 50,000*50,000 (and above) in the form of adjacency matrices on the 16GB RAM PC in our analysis. This result shows the limits on scalability of a study that relies on an adjacency matrix–based approach for network representation. Although the adjacency matrix–based representation is inefficient from a memory usage perspective, several algorithms that rely heavily on linear algebraic operations tend to be more efficient on adjacency matrices. Tasks such as clustering, search, and navigation are also more efficient on networks with adjacency matrix–based representation (assuming memory availability is not a constraint).

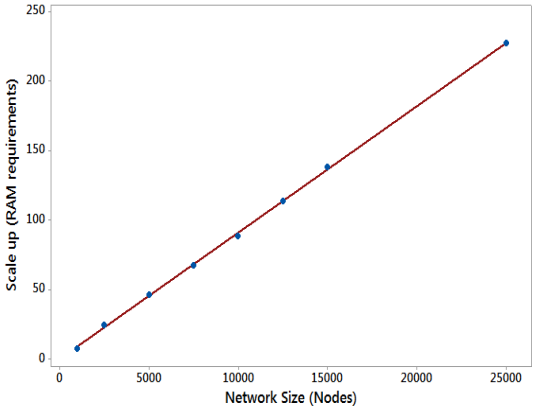
The gap between the list-based and sparse network–based representations is a constant, due to the use of three lists by the sparse matrix–based packages rather than one list in our egocentric list-based approach. Finally, as we expected, the storage required for the

adjacency-based approach scales quadratically with the size of the network (Fig. 1); however, the storage requirement for the sparse matrix–based and list-based approach scales linearly. The scale-up (ratio of RAM required to store the link information in an adjacency matrix format versus that in the proposed egocentric list format) is therefore a linear function of the number of nodes for a given average connectivity. A scale-up of about 400 is reached for networks with about 50,000 nodes. The projected scale-up for a network of with 1,000,000 nodes and average connectivity of 100 is about 9,000.

Figure 1 Changes in RAM requirements and scale-up (ratio of RAM required when network is represented using an adjacency matrix versus an egocentric list) with increasing network size



(a) RAM required to generate and store random networks of specified sizes



(b) Scale-up

In the case of weighted networks, the relative storage requirements for adjacency versus sparse matrix approaches remain fairly stable, because the existence of a connection (0 or 1) can be replaced by weights. However, in the case of the list approach, the generation of a weighted network requires the use of two lists: one to store the neighbors of each node and another to store the weights between neighbors. The storage gains are reduced approximately in half. Furthermore, all three methods are somewhat inefficient at handling dynamic networks when the links, or both the links and the nodes, change over time.

Impact of Efficient Computational Constructs

The speed-up achieved through the use of efficient computational constructs (vectorization and multi-core processing) is presented in Table 7. All these results are for the case in which the network is represented as a list and the computation is distributed to all eight cores of the PC.

Table 7 Comparison of Computational Time Required to Simulate a Directed Random Network with Average Connectivity of 100 Links (Using 8 Cores)

Number of Nodes	No VEC or MCP (1) (secs)	Only MCP (2) (secs)	Only VEC (3) (secs)	Both VEC and MCP (secs)	Speed-Up Comparisons		
					Speed-up (2) vs. (1)	Speed-up (3) vs. (1)	Speed-up Overall
1.00*10E3	3.02	0.71	0.08	0.024	4.25	37.75	125.8
2.50*10E3	17.51	3.88	0.58	0.090	4.51	30.19	194.6
5.00*10E3	70.01	16.09	1.87	0.385	4.35	37.44	181.8
7.50*10E3	171.02	36.85	3.72	0.778	4.64	45.97	219.8
1.00*10E4	290.60	66.28	6.15	1.480	4.38	47.25	196.4
1.25*10E4	440.89	101.65	9.28	2.118	4.34	47.51	208.2
1.50*10E4	646.96	147.16	13.05	3.013	4.40	49.58	214.7
2.50*10E4	1753.75	409.42	35.96	8.295	4.28	48.77	211.4
5.00*10E4	6167.91	1787.94	143.98	37.087	3.45	42.84	166.3
1.00*10E5	24044.9	6511.72	630.91	154.695	3.69	38.11	155.4
5.00*10E5	N.A	N.A	2557.57	665.111			
1.00*10E6	N.A	N.A	10324.13	2872.413			

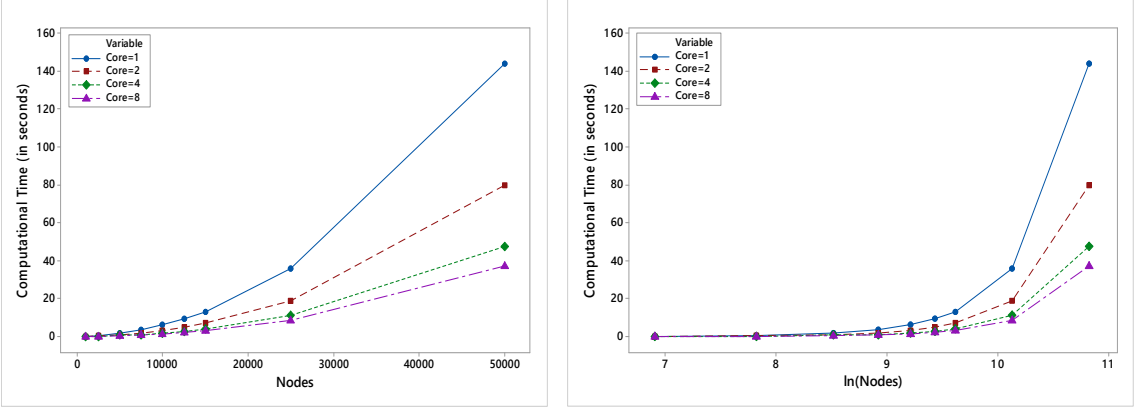
N.A not available

The speed-up achieved through the use of vectorization is about 40; it is about 4 with multi-core processing (using 8 cores), leading to an overall speed-up of about 160. Networks with 500,000 nodes (and average connectivity of 100) cannot be created within 24 hours when neither vectorization nor multi-core processing are used, so we do not present these results in Table 7.

Impact of Number of Cores

The number of cores available for computation plays an important role in how effective distributed computing is; Fig. 2 depicts the speed-up achieved as the number of cores increases.

Figure 2 Comparison of computational time required to simulate a directed random network with an average connectivity of 100 links, when different numbers of cores are used. When the number of nodes equals 1,000,000, ln(Nodes) = 13.8.



(a) Computational time as a function of the number of nodes (up to 50,000 nodes)

(b) Computational time as a function of the log of the number of nodes (up to 1M nodes)

In Fig. 2, we see that the speed-up is a nonlinear function of the number of nodes. The marginal improvement declines with increases in the number of cores. The speed-ups and scale-ups from using multiple cores depend on the algorithm, and the type of scale-ups in Table 6 may not be possible for algorithms in which some steps involve communication across cores or aggregation of information across cores. For example, some network generation algorithms (e.g., Watts-Strogatz, Barabasi-Albert, Erdos-Reyni (Newman 2006)) likely show a lower scale-up when the number of cores increases because some intermediate steps in these algorithms require communication across cores. Table 7 provides a comparison of the computational times required for generating networks using

four different algorithms. Because all steps in the construction of a directed random network and the caveman model can be parallelized, their generation times are lower than the generation of networks using the Watts-Strogatz algorithm (for which only some steps are parallelized) or the Barabasi-Albert algorithm (for which no steps are parallelized).

Although the networks are generally sparse, and theoretically it should be easy to partition the networks and distribute computational tasks over multiple cores, the reality is different. The authors in (Lumsdaine et al. 2007, Marino and Stawinoga 2011) provide detailed descriptions of some of the problems inherent to this task.

Graph Sampling-Based Network Metrics

The problem of estimating network metrics, such as average path length and average degree, is commonly encountered in both simulations and empirical studies and is among the most time-consuming activities in network analysis. Computationally efficient algorithms thus are of significant value to researchers. In this section, we focus on algorithms to estimate six common network metrics (degree distribution, average degree, shortest path length distribution, diameter, average diameter, and average path length) by combining the concepts of vectorization and distributed computing with graph sampling. The algorithms are agnostic about whether the data are real or synthetic.

Degree Distribution

When we use an egocentric list-based representation of a network, the degree of a node i is simply a count of the number of links that correspond to a node at location/index i in the list. Sampling involves randomly selecting a set of nodes, counting the number of links for each node, and generating their degree distribution. We use the following algorithm to

implement the sampling and calculate the degree distribution using multi-core processing and vectorization:

Table 8 Degree Distribution Algorithm

Algorithm 1: Degree Distribution

1. Determine C, the number of cores available for computing the degree distribution, and obtain s, the sample size selected by the user (default $C = 1, s = 0.1$).
2. Sample $\lceil s * N \rceil$ nodes from the network.
3. Assign C of the sampled nodes to each of the C cores (one node to each core); obtain the degree of the node on the cluster.
4. When the operation in a core is complete, assign a new node to that core to obtain its degree. Store the degree in a list object in the core.
5. Repeat Step 4 until the degree of all the $\lceil s * N \rceil$ nodes has been obtained.
6. Collect the C list objects on one core and merge them.
7. Obtain the degree distribution of the nodes as a list object.

The input to the algorithm is the network in in an egocentric list-based representation. The algorithm can be further speeded up by assigning nodes in batches of size > 1 (ideally equal to $\lceil s * N/C \rceil$); however, this method assumes there are no other ongoing processes on the cores.

Average Degree

The average degree is derived from the degree distribution and can be represented as:

$$average\ degree = \frac{\sum_{i \in Nodes} degree(v_i)}{\sum_{i \in Nodes} i}$$

(1) where i is an index of the sampled node.

Shortest Path Length Distribution

We use the following algorithm to estimate the shortest path length distribution:

Table 9 Shortest Path length Algorithm

Algorithm 2: Shortest Path Length Distribution

1. Determine C, the number of cores available for computing the shortest path length distribution, and obtain s, the sample size selected by the user (default $C = 1, s = 0.1$).
2. Create C images of the network, one on each available core.
3. Sample $\lceil s * N \rceil$ nodes.
4. Assign one of the sampled nodes to each of the C cores; obtain the shortest path length of the sampled node with the remaining $N - 1$ nodes in the network using the algorithm in (Dijkstra 1959a).
5. When the operation in a core is complete, assign a new node to that core to obtain its shortest path-length distribution. Store the degree in a list object in the core.
6. Repeat Step 5 until the degree of all $\lceil s * N \rceil$ nodes has been obtained.
7. Collect the C list objects on one core and merge them; there will be $(N - 1) * \lceil s * N \rceil$ elements in this list.
8. Obtain the distribution of the shortest path lengths.

We also use the egocentric-list-based representation of the network in this algorithm. Our algorithm can be speeded up further by using the Dijkstra algorithm on only a subset of the network on each core, such as a select sample of $\lceil s' * N \rceil$ nodes (where s and s' may or may not be equal).

Diameter

An estimate of the diameter is simply the largest of the $(N-1) * \lceil s * N \rceil$ estimated path lengths.

Average Diameter

When the algorithm for the shortest path length distribution has been executed, we obtain a vector with $\lceil s * N \rceil * (N-1)$ elements. The average diameter can be estimated as follows:

Table 10 Average Diameter Algorithm

Algorithm 3: Average Diameter

1. Determine C, the number of cores available for computing the shortest path length distribution, and obtain s, the sample size selected by the user (default $C = 1, s = 0.1$).
2. Estimate the $(N-1) * \lceil s * N \rceil$ shortest path lengths.
3. For each of the sampled $\lceil s * N \rceil$ elements, find the maximum of the $(N-1)$ shortest path lengths (i.e., diameter of the network from the sample nodes perspective).
4. Find the average of these diameters by $average\ diameter = \frac{\sum_i^{\lceil s*N \rceil} max\ D(v_i, v_j)}{\lceil s*N \rceil}$, where $D(v_i, v_j)$ is the shortest path (or geodesic) between node i and j, i is the sampled node, and j is the index for all remaining $(N-1)$ nodes in the network.

Average Path Length

The average path length is also calculated using the shortest path length distribution; it is the average of the shortest path lengths between any two sets of nodes. The average path length can be represented as:

$$average\ path\ lenght = \frac{\sum_{j}^{N}\sum_{i}^{\lceil s*N \rceil} max\ D(v_i,v_j)}{\lceil s*N \rceil*(N-1)} \quad (2)$$

Tables 11-13 summarize the speed-up and accuracy achieved using various sampling-based algorithms. The speed-up in the computational times is about 40, and the accuracy of the results is robust to changes in sample size. However, they depend on how accurately the metrics are preserved in the subgraph.

Table 11 Performance of Sampling-Based Algorithm for Estimating Average Degree of the Network

Network Size	Actual Metric Value	Estimated Metric Value with Sample Sizes from 10–30%			% Error for 10%	Computational Time		Speed-Up Ratio
		10%	20%	30%		Full Network	10%	
1.00*10E3	98.5110	98.5099	98.5101	98.5102	0.0011%	0.0301	0.0116	2.6
2.50*10E3	98.7730	98.7712	98.7716	98.7717	0.0018%	0.0532	0.0225	2.4
5.00*10E3	99.1530	99.1498	99.1505	99.1508	0.0032%	0.2011	0.0797	2.5
7.50*10E3	99.5060	99.5018	99.5026	99.5031	0.0042%	0.3108	0.1224	2.5
1.00*10E4	99.7790	99.7725	99.7738	99.7744	0.0065%	0.4074	0.1584	2.6
1.25*10E4	100.1120	100.1045	100.1060	100.1067	0.0075%	0.5438	0.2116	2.6
1.50*10E4	100.3320	100.3234	100.3251	100.3260	0.0086%	0.7096	0.2879	2.5
2.50*10E4	100.5930	100.5873	100.5885	100.5890	0.0056%	1.0805	0.4188	2.6
5.00*10E4	100.5390	100.5355	100.5362	100.5366	0.0035%	2.5285	1.0401	2.4
1.00*10E5	100.4480	100.4459	100.4463	100.4466	0.0021%	5.4485	2.2030	2.5

Table 12 Performance of Sampling-Based Algorithm for Estimating Average Diameter of the Network

Nodes	Actual Metric value	Estimated Metric Value with Sample Sizes from 10–30%			% Error for 10%	Computational Time		Speed-Up Ratio
		10%	20%	30%		Full Network	10%	
1.00*10E3	3.0318	3.0313	3.0314	3.0315	0.0139%	1.588	0.199	7.9
2.50*10E3	3.5510	3.5503	3.5505	3.5505	0.0200%	17.016	0.778	21.9
5.00*10E3	3.9861	3.9849	3.9852	3.9853	0.0281%	54.424	1.827	29.8
7.50*10E3	4.2221	4.2204	4.2208	4.2209	0.0402%	110.530	3.834	28.8
1.00*10E4	4.5914	4.5891	4.5896	4.5898	0.0495%	186.858	5.446	34.3
1.25*10E4	4.9135	4.9105	4.9111	4.9114	0.0606%	303.058	7.615	39.8
1.50*10E4	5.1135	5.1101	5.1108	5.1111	0.0668%	459.608	12.556	36.6
2.50*10E4	5.2200	5.2177	5.2181	5.2183	0.0443%	1377.267	37.541	36.7
5.00*10E4	5.5135	5.5122	5.5125	5.5126	0.0241%	8166.649	197.584	41.3
1.00*10E5	5.7958	5.7950	5.7951	5.7952	0.0150%	64737.193	1549.112	41.8

Table 13 Performance of Sampling Based Algorithm for Estimating Average Path Length of the Network

Nodes	Actual Metric value	Estimated Metric Value with Sample Sizes from 10–30%			% Error for 10%	Computational Time		Speed-Up Ratio
		10%	20%	30%		Full Network	10%	
1.00*10E3	1.8991	1.8987	1.8988	1.8988	0.0211%	1.590	0.200	8.0
2.50*10E3	2.0174	2.0167	2.0168	2.0169	0.0347%	17.038	0.780	21.8
5.00*10E3	2.1114	2.1103	2.1105	2.1106	0.0521%	54.500	1.830	29.8
7.50*10E3	2.2199	2.2183	2.2186	2.2188	0.0721%	110.752	3.843	28.8
1.00*10E4	2.3543	2.3521	2.3525	2.3528	0.0934%	187.270	5.450	34.4
1.25*10E4	2.4211	2.4183	2.4189	2.4191	0.1156%	303.756	7.630	39.8
1.50*10E4	2.5034	2.5002	2.5008	2.5012	0.1278%	460.160	12.570	36.6
2.50*10E4	2.6638	2.6616	2.6620	2.6623	0.0826%	1379.060	37.620	36.7
5.00*10E4	2.8152	2.8139	2.8142	2.8143	0.0462%	8178.098	198.020	41.3
1.00*10E5	2.9029	2.9021	2.9023	2.9023	0.0276%	64853.930	1551.750	41.8

Conclusions and Further Work

We present an efficient network simulation and analysis approach that can be used to generate and analyze mid-sized networks on regular PCs with limited RAM, using the open source R platform. This approach allows us to simulate and analyze networks that are more

than 400 times larger than is possible using traditional network simulation/analysis tools, at a rate that is more than 20 times faster. We gain computational efficiency through:

- The use of efficient network representation, with an egocentric representation of the network with a list-based (as opposed to a matrix-based or sparse matrix–based) approach for storing link information. This representation reduces the memory required for the storage of link information by a factor of more than 400 for networks of size 50,000 or above.
- The use of efficient computational constructs, with computing constructs such as vectorization and multi-core processing (if available), to gain a speed-up factor of about 20 or more for networks of size 50,000 or above.
- Efficient algorithms, which combine the concepts of graph sampling with those of efficient computing to reduce computational times for some key network metrics, such as degree distribution, shortest path length, and diameter, by a factor of more than 40.

Although some R packages and Python and C++ libraries exist for simulation and network analysis, they do not necessarily enable the modeling and simulation of large-scale systems with limited RAM capacity.

We envision extending the capabilities of the computational approach and developing computationally efficient algorithms as follows:

- Extending the number of algorithms for measuring network metrics. Global metrics such as clustering coefficients can be estimated quickly and accurately by graph sampling–based or distributed computing–based algorithms; we plan to

develop algorithms for some metrics, such as those related to centrality, authority, and transitivity.

- Incorporating the ability to scale the algorithm to multiple processors. Future extensions of our methods are possible, to allow communication between computers and high-level parsing and parallelization in networks with similar or different approaches to job assignation by using a thread-type programming environment within R.

We are also developing algorithms for efficient partitioning of the nodes on cores and processors (workload balancing) and the use of asynchronous simulations to capture diffusion dynamics.

Chapter 4: Random Sampling Based Approaches for the Estimation of Average Path Lengths of Networks

Overview

The average of the shortest paths between any two nodes of a network is a global metric of great relevance. Popularly called as the average path length (APL), it provides useful insights on the level of interconnectivity in a network and the time it would take for information/goods to flow between any two randomly selected points on the network. APL has been shown to be an important metric for tasks such as the designing of real life transportation networks (Balmer, Nagel, and Raney 2004, Klunder and Post 2006, Ziliaskopoulos, Kotzinos, and Mahmassani 1997), design of routing networks (Costa et al. 2007, Dabek et al. 2004), design of web-based networks (Backstrom et al. 2012, Fu, Liu, and Wang 2008, Kleinberg 2000, Newman 2000), studying propagation of diseases (Dekker 2013), diffusion of information (Cha, Mislove, and Gummadi 2009) and opinion dynamics (Yildiz et al. 2011b). Researchers have also shown that search and navigation is easier when APL is small (Zhang et al. 2008). However, estimating APL takes a lot of time and is sometimes infeasible on account of lack of computational resources (Wang 2006).

Researchers have shown that he computational time required to estimate APL scale as $V(V + E')$ where V is the number of vertices and E' a function of the number of edges in the network (Madduri et al. 2007). E' itself scales as $O(V^{\gamma})$ where $1 \leq \gamma \leq 2$. This quadratic to cubic scaling of the computational time with network size makes the estimation of APL impractical as the network sizes increase (Wang 2006). As an example, it takes approximately 9.6 hours to estimate the APL of a synthetic BA network with about 100,000 nodes on a 16GB RAM PC and the time requirement increases to more than 5 days for a network with about 1 Million nodes. Such a long wait time is generally

impractical, especially in simulation and emulation studies that require generation of 100s of synthetic networks and the estimation of APL for each scenario.

In this chapter, we develop a random node pair sampling based strategy to estimate the APL for mid-sized networks when both computational time and capacity are limited. Though sampling introduces some uncertainty in the reliability of the parameter estimates, the precision and confidence can be controlled using a combination of the right sampling strategy and sample size. We therefore propose and demonstrate the efficacy (in terms of computational time, confidence level, and precision) of the proposed node pair sampling algorithm. We compare the proposed algorithm with random node sampling algorithm and algorithms wherein the node sampling is non-uniform. The random node pair sampling algorithm yields a speed up factor of more than 411 when compared to the algorithm that uses random node sampling and a speed up of 750 when compared to the algorithm that measure APL using the population information. The proposed algorithm uses the central limit theorem approximation to determine the sample size for a given precision and confidence level.

This chapter is organized into 6 sections. We present a brief literature review in Section 2 where we focus on the algorithms used for estimating the shortest path lengths (SPL) and network sampling. APL estimates generally use SPL. The proposed sampling based APL estimation algorithms are presented in Section 3, and their performance on simulated networks is discussed in Section 4. The algorithms are also applied to real networks and the performance of the algorithms on real networks in presented in Section 5. Finally, the conclusions are presented in Section 6. We limit our focus on mid-sized network-i.e., networks of size up to 5 million nodes as these can be processed on a single core on most

modern day PCs. We do this so that the focus of the chapter stays on speed-up and scale-up due to sampling without the need for discussing graph partitioning.

Background

Estimation of APL is intricately linked to the distribution of the shortest paths between any two randomly selected nodes.

The SPL Problem

The SPL (also called the geodesic distance) between a pair of nodes, is defined as the minimum number of nodes that need to be traversed to reach a desired destination node from a given source. SPL is used in several areas such as transportation systems and route planning (Balmer, Nagel, and Raney 2004, Klunder and Post 2006, Ziliaskopoulos, Kotzinos, and Mahmassani 1997), server selection and data queries (Costa et al. 2007, Dabek et al. 2004, Rétvári, Bíró, and Cinkler 2007), and path finding in social networks (Boyles and Rambha 2016, Kleinberg 2000, Leskovec and Faloutsos 2006b, Leskovec, Kleinberg, and Faloutsos 2005).

Based on the domain and the purpose of estimating SPL, the research on SPL can be classified into four. The first class of research on SPL attempt to find the shortest path between a specified source i and destination j. The second class of research focuses on finding the SPL from source i to all other nodes while the third class focuses on finding the SPL between all combinations of sources i and destinations j. The fourth class relies on the creation of component hierarchies for each node based on spanning trees and subsequently estimate the SPL.

Researchers usually rely on the Floyd-Warshall algorithm (Floyd 1962) and its derivatives for finding the shortest path and SPL between a specified source i and destination j. This algorithm compares all the paths in a network from a node i to j, starting with a comparison based only on their neighbors, and continuously increasing until the optimal value of the shortest path is reached. This recurrence may be seen as a dynamic programming sequence. The algorithm scales as $\mathrm{O}(\mathrm{V}^3)$ when it is used to find the SPL between all pairs of nodes and requires that the weights of the network are positive real numbers.

The second class of algorithms build upon the research in (Dijkstra 1959b). Dijkstra's algorithm solves the problem of finding the SPL from source (i) to all other nodes by visiting vertices in a non-decreasing order. For this purpose, three types of sets are kept in memory: unreached, queued, and visited nodes. Starting from a source node, the graph is explored, visiting first the neighbor nodes of the source, then moving to the neighbors of their neighbors, and continuing this iteration until all the nodes have been visited. In the process, nodes are removed from visiting if they were already visited. If a node is unreachable from a source, the infinity value is assigned to that specific path. The running time from a source to all the destinations is $O(E + V \log V)$. This formulation works for networks with real weights; nonetheless, in the case of negative weights the search could take exponential time. In the case of unweight networks, directed or undirected, this algorithm turns into a depth breath search that performs a depth breath search in $O(E + V \log V)$ time.

The third set of algorithms build upon the research in (Johnson 1977). In this algorithm, a reweighted process is done first, so every edge has non-negative values. After this

process, the algorithm follows the same principles of Dijkstra's, but iterating over all the nodes. In this case, the computations take $O(V^2 \log V + VE)$ for the all-pairs problem.

The fourth class of algorithms create component hierarchies for each node based on spanning trees and then find the shortest paths as the distances represented from the nodes contained in the different components. This procedure works in a linear fashion only for networks with integer and non-zero weights. The processing time in the best case scenario is $O(E + V \log r)$ where $r \ll V$. However, as noted by (Crobak et al. 2007) this construction needs a careful tailoring of the component set since it is a complex database tasks that vary depending on the topology of the graph.

Scalability of SPL Algorithms

Several researchers have focused on the scale-up and speed-up of the estimation of the SPL using distributed/parallel computing. The general problem with parallelization is maintaining the balance between the overhead of communication between cores and the amount of information to load on each core. As the amount of information on a core increases, the communication overhead decreases but the memory requirements increase. The scale-up as well as the speed-up are dependent on the graph partitioning algorithms.

In (Madduri et al. 2007) the authors parallelize the Dijkstra's algorithm and propose a strategy that reduces the overhead communication in multithreaded computing. They use 40 processors and tested their graph partitioning and SPL estimation approach on both real and synthetic networks with up to 100 million nodes and 1 billion edges. The authors report that one full exploration from a source to the rest of the nodes takes on average 9.73 seconds, with a speed up in the order of 31 times. The authors in (Mao and Zhang 2013, 2014) calculate all the shortest paths by dividing the jobs equally among cores based on

the out degree of the pair of nodes. They use a cluster with 72 cores and report finding the value for the APL of a 2.5 million node network in 6 days and 5.5 hours.

Thorup's algorithm (Thorup 1999) with component hierarchies can be parallelized as well. A parallel implementation of their algorithm for undirected weighted graphs on a MT2 computer (40 processors) has been reported and results suggest that the scale up increases from 2 to 40 as the network size increases from 1 million to 1 billion nodes.

Calculating APL

Analytical approaches to estimating the APL exist for some synthetic networks (see Table 14).

Table 14 Closed form of APL for 5 type of networks

Network	Complete Network	Random graph	Regular Lattice	Watts-Strogatz Graph	Barabasi-Albert Graph
Closed form	1	$l \approx \frac{\ln(V)}{\ln(K)}$	$l = \frac{V}{2K} \gg 1$	$\left(\frac{V}{2K}, \frac{\ln(V)}{\ln(K)}\right)$	$l \sim \frac{\ln V}{\ln \ln V}$

However, in general, the APL can be estimated from SPL as:

$$\text{APL} = \frac{1}{N(N-1)} \sum_{i \neq j} d(i,j) \quad (1)$$

Where N is network size or the number of nodes, , $j \in V$, V is the set of nodes, $d(i,j)$ denotes the SPL between i and j. If there is no path between i and j this value is not considered in the calculations.

Other than the use of pre-computed SPL's the APLs can be computed using the goal post algorithm (also called the shortest path queries (Sommer 2014)). In this case the emphasis lies on simplifying the SPL algorithm, using intermediate nodes to calculate

approximated distances instead of a full shortest path search (Dijkstra 1959b). In the large network case, node sampling have been used to determine the goal post nodes out of all the possible nodes of the network (Potamias et al. 2009). However, the authors show that selecting the optimal set of landmarks is a NP-Hard problem.

One approach to scale-up/sped-up the goal post algorithm is through the use of random sampling. In (Sommer 2014) the authors propose this strategy and show that mixed sampling methods as random sample based on degree distribution performance well.

Interestingly, even though sampling based approaches have been introduced in the goal post algorithms and are common in estimating other network metrics such as betweeness centrality, they are unavailable for estimating APL using SPL. Sampling based approaches have been used to estimate several network metrics, especially those related to node centrality. For example, (Wang 2006) proposed a sampling algorithm to estimate the centrality of all vertices within a probabilistic framework, selecting a desired level of error based on Hoeffding´s inequality (Hoeffding 1963). Similarly, (Brandes and Pich 2007) proposed a dependency score based approach to estimating the centrality were the scores are estimated for a small sample of nodes.

Network Sampling

There are three general approaches have been developed in the literature for network sampling:

- Random node sampling: In the uniform random node sampling method, a subset of nodes is selected independently with equal probability from the set of all nodes (Frank 1979a). There are several variations of random node sampling

wherein the version with probability of a node's selection being proportional to their degree is common (Ebbes, Huang, and Rangaswamy 2013).

- Random link sampling: This method is also called incident subgraph sampling. In this procedure edges are selected independently with the same probability. This method is common in IP protocols and internet literature (Crovella et al. 2002).
- Snowball sampling: The methods under this category sample a part of the graph based on the exploration of links and nodes. The first method can be attributed to the works of (Coleman, Katz, and Menzel 1957)and formally to (Goodman 1961). There are two main variations of snowball sampling.
    - The random walk method: In the random walk method, (Klovdahl et al. 1977b) proposes a selection of exactly one neighbor uniformly at random from all the unselected neighbors after the first step of the snowball method.
    - The forest fire method: In this case the selection lies between one and all the nodes or contacts of a preceding node based on an ad-hoc called burning probability, or percentage of nodes to be followed after the first step of sampling in a snowball procedure.

Out of the three methodologies, snowball sampling is recognized as the best one for sampling dynamic graphs and recovering structures that might have change in time and are unknown (Leskovec and Faloutsos 2006b).

The authors in (Lee, Kim, and Jeong 2006b) test the three different type of sampling methods – node sampling, link sampling, snowball sampling- to calculate centrality

measures, including APL. In this case, the sampling technique is used to reconstruct a sub-graph, and then use it to construct the centrality measures. Through simulation and real network data –size up to 50,000 nodes, the authors showed that snowball sampling gives the best results. Nonetheless, to get an accurate estimate of the APL, betweenness and other centrality measures, almost 60% to 80% of the network needs to be recovered in the sampled subgraph. This fact causes computational issues for the calculations.

Our Approach

In this chapter we propose a node pair based sampling approach to calculate the APL. We sample a fixed number of pairs of nodes from the $\binom{V}{2}$ possible combinations and estimate the SPL between each pair. Further, we build upon the work of (Mao and Zhang 2014) and split the sampling and SPL estimation problem on multiple cores by loading copies of the network on all cores. The proposed algorithm, and three variations of the algorithm are presented next.

Sampling Based Algorithm for Estimating APL

We propose four algorithms to calculate the APL. Two algorithms (Algorithm 1 and 2) are based on random node sampling while the other two (Algorithm 3 and 4) use the proposed random node pair sampling approach. Further, Algorithms 1 and 3 use the uniform random distribution for sampling while Algorithms 2 and 4 use a degree weighted sampling algorithm. All of them share the Dijkstra's algorithm foundations for SPL calculations. The details are as follows:

Algorithm 1

In the first case, we propose a node selection method based on uniform random sampling without replacement. Once the source nodes have been selected, the shortest path calculations are performed from all the sources against all the nodes in the network. Following (Brandes and Pich 2007) the required sample size r for an error ε with probability at least $1-\delta$ is given by

$$r \leq \frac{1}{2\varepsilon^2}(\ln N + \ln 2 + \ln \frac{1}{\delta}) \qquad (2)$$

The algorithm is defined in table 15.

Table 15 APL node sampling algorithm

Steps
1. Determine C, the number of cores available for computing the shortest path length distribution.
2. Define s, the sample size selected by the user (default $\varepsilon = 0.03, \delta = 0.05$)
3. Create C images of the network, one on each available core
4. Sample s nodes and split the sample size equally in batches of size $s_C = \frac{s}{C}$ among the C cores
5. On each core, use the first node of the selected node's set s_C as a source to obtain the shortest path length of the sampled node with the remaining $N-1$ nodes in the network using the Dijkstra algorithm (Dijkstra 1959b).
6. Store C lists on each one of the cores only with the number of steps from the source to each of the $N-1$ nodes explored
7. Repeat step 5 and 6 on each core for the rest of the sample nodes

8. Collect the C list objects on one core and merge them; there will be $s(N-1)$ elements in this list.
9. Obtain the average from the extracted list of shortest path lengths. This result is the final APL output.

Algorithm 2

The second case follows the same statistical foundations of case one, with a slightly modification on the selection process of the source nodes. The sampling selection is made based on the out degree distribution of the nodes. The main idea is that highly connected nodes should have a higher probability of being selected since shortest path calculations depend on the connectivity of the nodes. Our conjecture is that uniform sampling is given the same importance to all types of nodes, therefore missing relevant nodes and not accurately capturing the topology of the network.

The algorithm is defined in table 16.

Table 16 APL weighted node sampling algorithm

Steps
1. Determine C, the number of cores available for computing the shortest path length distribution.
2. Define s, the sample size selected by the user (default $\varepsilon = 0.03, \delta = 0.05$)
3. Calculate the degree distribution of the network for all the nodes

4. Calculate the probability $p_v = \frac{Deg(v)}{\sum_v Deg(v)}$ where Deg is the degree of node v

5. Create C images of the network, one on each available core

6. Sample s nodes and split the sample size equally in batches of size $s_C = \frac{s}{C}$ among the C cores. Use p as a vector of sampling probabilities for the selection process

7. On each core, use the first node of the selected node's set s_C as a source to obtain the shortest path length of the sampled node with the remaining $N - 1$ nodes in the network using the Dijkstra algorithm (Dijkstra 1959b).

8. Store C lists on each one of the cores only with the number of steps from the source to each of the $N - 1$ nodes explored

9. Repeat step 7 and 8 on each core for the rest of the sample nodes

10. Collect the C list objects on one core and merge them; there will be $s(N - 1)$ elements in this list.

Obtain the average from the extracted list of shortest path lengths. This result is the final APL output.

Algorithm 3

The third approach is based on the actual specifications of the APL. Since we are interested in calculating an average using only a sample, we can rely on the asymptotic distribution of samples. Therefore, we need to focus our sample size and selection process not on nodes, but on $\sigma_{i,j}$. This means that we need to select pairs of nodes and perform the

shortest path calculations. For a sample size r with an error ε with probability at least $1 - \delta$ the sample size is given by

$$r = \left(\frac{Z_{\delta} s}{\varepsilon}\right)^2 \qquad (3)$$

where s is the standard deviation and Z the standardized value of a normal distribution. In this case, the selection process of the pair of nodes is done using uniform random sampling without replacement, selecting the nodes simultaneously. To estimate s we sample 1,000 $\sigma_{i,j}$, then calculate the sample size and proceed with the final sample calculations.

The algorithm is defined in table 17.

Table 17 APL node pair sampling algorithm

Steps
1. Determine C, the number of cores available for computing the shortest path length distribution.
2. Sample $1{,}000$ pairs of nodes to determine the variance s of their shortest path distribution. The shortest path search uses a modified Dijkstra (1956) algorithm that stops the search as soon as the destination node is reached.
3. Define r, the sample size selected by the user (default $\varepsilon = 0.03, \delta = 0.05$)
4. Create C images of the network, one on each available core
5. Sample r pair of nodes and split the sample size equally in batches of size $r_C = \frac{r}{C}$ among the C cores

6. On each core, use the pair of the selected nodes from the set r_C to obtain the shortest path length between them using the modified Dijkstra algorithm.
7. Store C lists on each one of the cores only with the number of steps from the source to destination for each element of r_C
8. Repeat step 6 and 7 on each core for the rest of the sample pairs
9. Collect the C list objects on one core and merge them; there will be r elements in this list.
10. Obtain the average from the extracted list of SPL. This result is the final APL output.

Algorithm 4

Our four method relies on the same statistical ideas of the third sampling approach, with the only difference that the selection process is done using a random sampling based on the out degree distribution of the nodes. Once again, the fundamental idea is to get the majority of the $\sigma_{i,j}$ based on a more accurate distribution of the connection topology of the network.

Table 18 APL weighted node pair sampling algorithm

Steps
1. Determine C, the number of cores available for computing the shortest path length distribution.
2. Calculate the degree distribution of the network for all the nodes

3. Calculate the probability $p_v = \frac{Deg(v)}{\sum_v Deg(v)}$ where Deg is the degree of node v

4. Sample $1{,}000$ pairs of nodes to determine the variance s of their shortest path distribution. Use p as a vector of sampling probabilities for the selection process. The shortest path search uses a modified Dijkstra (1956) algorithm that stops the search as soon as the destination node is reached.

5. Define r, the sample size selected by the user (default $\varepsilon = 0.03, \delta = 0.05$)

6. Create C images of the network, one on each available core

7. Sample r pair of nodes and split the sample size equally in batches of size $r_C = \frac{r}{C}$ among the C cores. Use p as a vector of sampling probabilities for the selection process

8. On each core, use the pair of the selected nodes from the set r_C to obtain the SPL between them using the modified Dijkstra algorithm.

9. Store C lists on each one of the cores only with the number of steps from the source to destination for each element of r_C

10. Repeat step 8 and 9 on each core for the rest of the sample pairs

11. Collect the C list objects on one core and merge them; there will be r elements in this list.

12. Obtain the average from the extracted list of shortest path lengths. This is the final APL output.

Computational Analysis

We test our four methods on three different types of simulated networks, using ten different network sizes, and nine different combinations of confidence level and error. Table 19 summarizes our setting.

Table 19 Computational experiment setting

Network Type	Random (RN) , Barabasi-Albert (BA) and Wattz-Strogatz (WS) with average connectivity of 100 nodes
Size[7]	1000, 2500, 5000, 7500, 10000, 15000, 25000, 50000, 100000
Error	0.03, 0.05, 0.07
C.I	90%, 95%, 99%

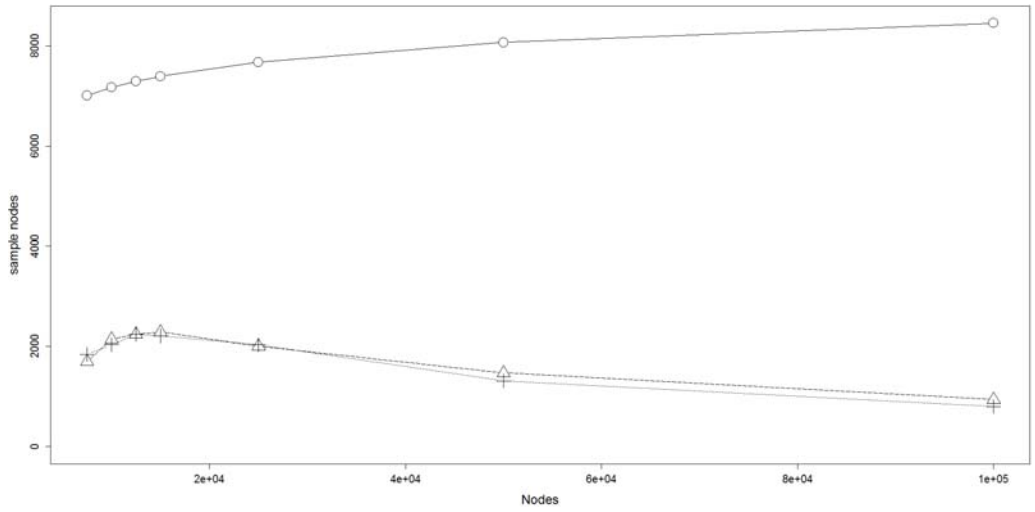
We use an INTEL® CORE™ i7-4710MQ notebook with 16GB RAM for our experiment. In addition, we have used the R programming platform for the simulation and analysis.

Key Results

Impact of Increase in Network Size: The impact of increase in size of a RN on three measures (a) the number of nodes sampled, (b) the number of links sampled, and (c) the computational time is presented in Figure 3. The confidence interval was fixed at 95% and precision at +/-3% for all the simulations.

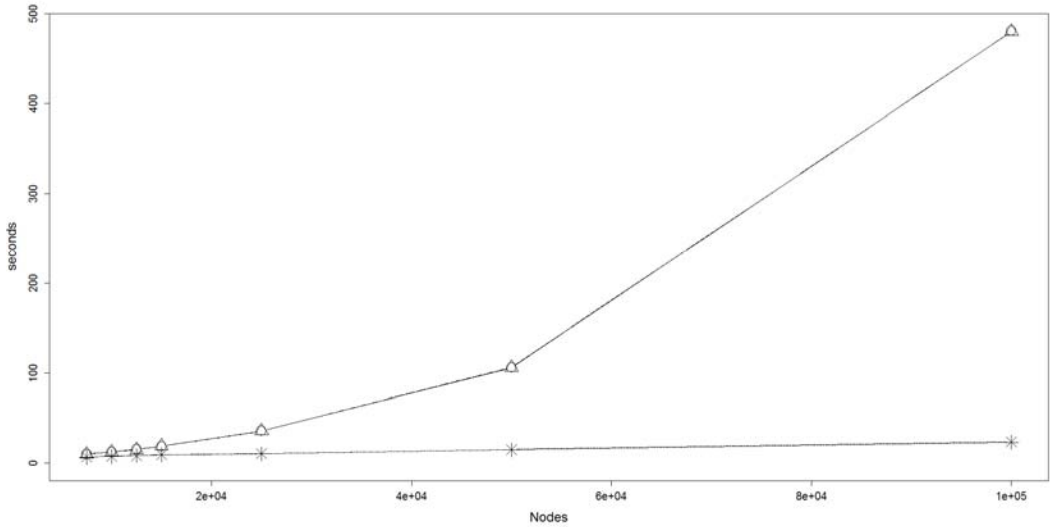
[7] Only in the case of the RN we include estimations for 500,000 and 1 million node networks.

Figure 3 Impact of increase in size of the underlying RG on (a) the number of nodes sampled, (b) the number of links sampled, and (c) the computational time.



Circle Algorithm 1 and 2, Triangle Algorithm 3, Cross Algorithm 4

(a)



Circle Algorithm 1 and 2, Triangle Algorithm 3, Cross Algorithm 4

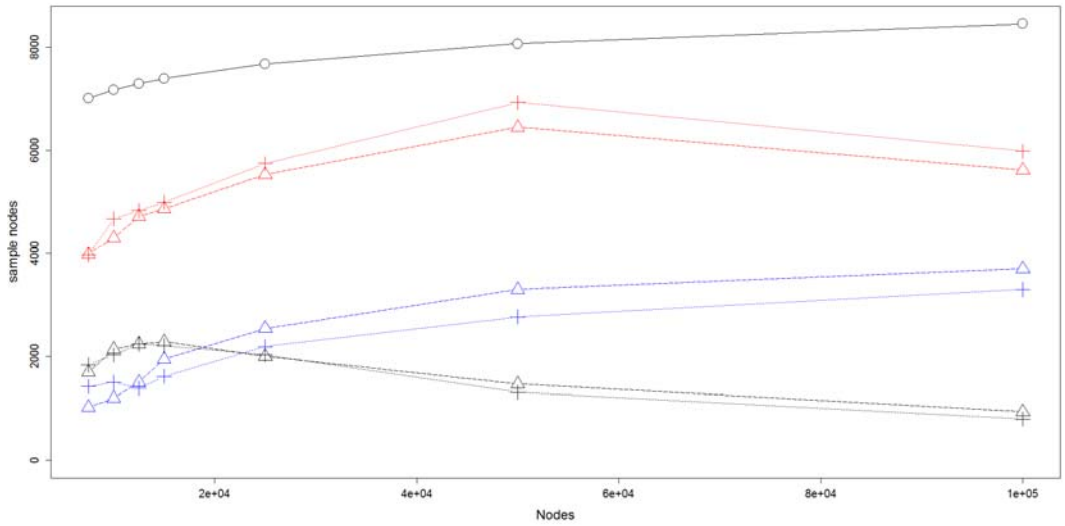
(c)

The results indicate that all the three metrics are significantly lower for algorithms 3 and 4 as compared to algorithms 1 and 2. Further, we find that the sample size requirements start decreasing as the network size increases beyond about 100,000 nodes.

Joint Impact of Network Structure and Size

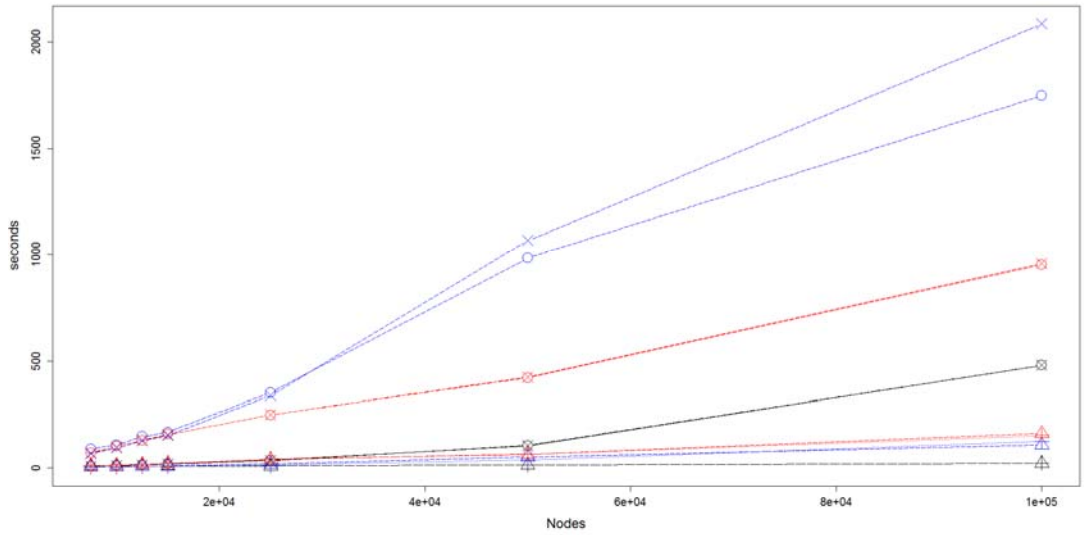
We extended Study 1 to include 3 network types-RG, WS, and BA. The confidence interval was fixed at 95% and precision at +/-3% for all the simulations. The results are presented in Figure 4.

Figure 4 Joint impact of network structure and size on (a) the number of nodes sampled, and (b) the computational time.



Circle Algorithm 1 and 2, Triangle Algorithm 3, Cross Algorithm 4; Black RN, Blue BA, Red WS

(a)

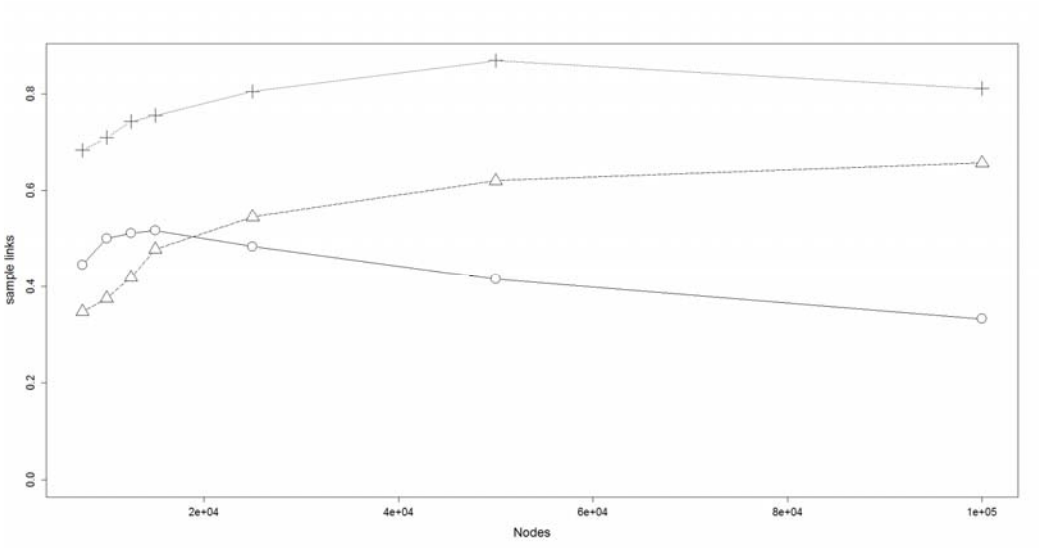


Circle Algorithm 1 and 2, Triangle Algorithm 3, Cross Algorithm 4; Black RN, Blue BA, Red WS

(b)

We observe that (a) the computational times are also dependent on the network structure, and (b) the required number of nodes that need to be sampled decreases as the network size increases for the RN and WS network. Further, as we saw in Study 1, the sampling as well as the time requirements for Algorithm 3 and 4 are significantly lower that the time and sample size requirements for Algorithms 1 and 2.

Figure 5 Standard Deviation in SPL as a function of Network Type and Size

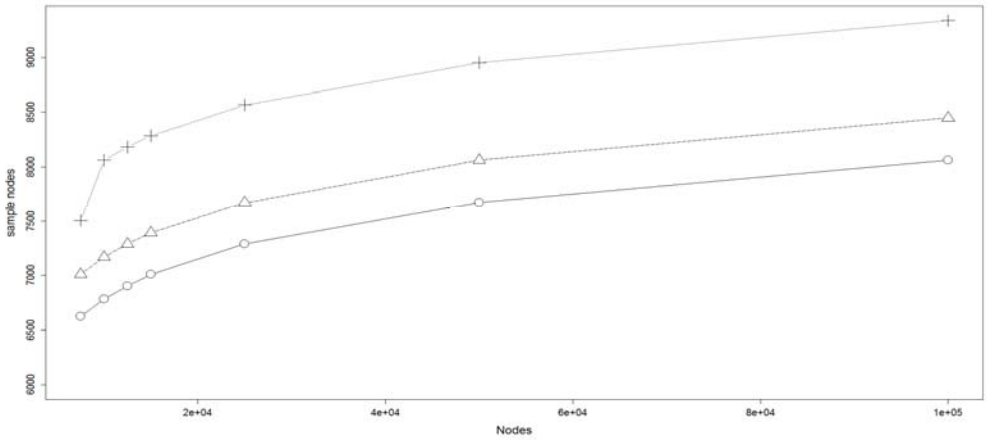


Circle RN, Triangle BA, Cross WS

Impact of Changes in CI and Precision on Sample Size

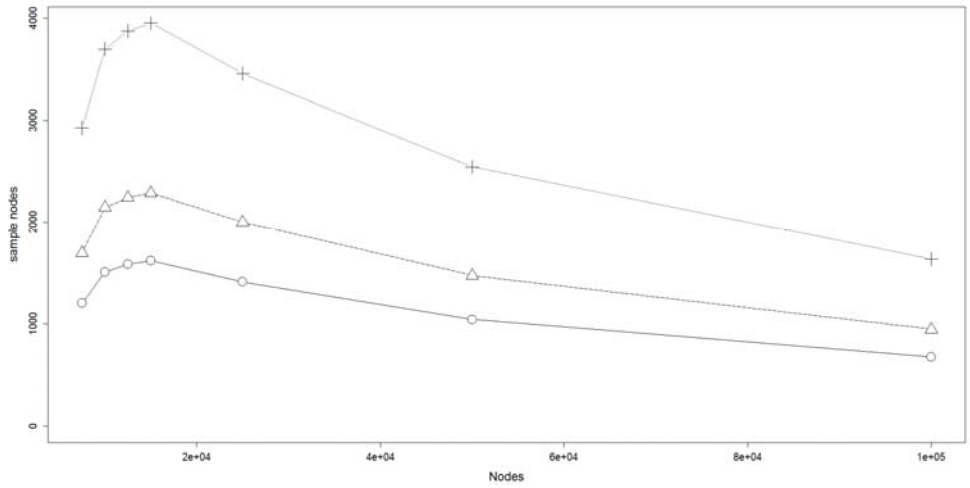
We studied the impact of changes in the confidence interval and precision on the sample size from a RN and the results are presented in Figure 6 and 7 respectively.

Figure 6 Impact of Change in Confidence Intervals on the Sample Size when we use (a) Algorithm 1 or 2, (b) Algorithm 3, and (c) Algorithm 4.



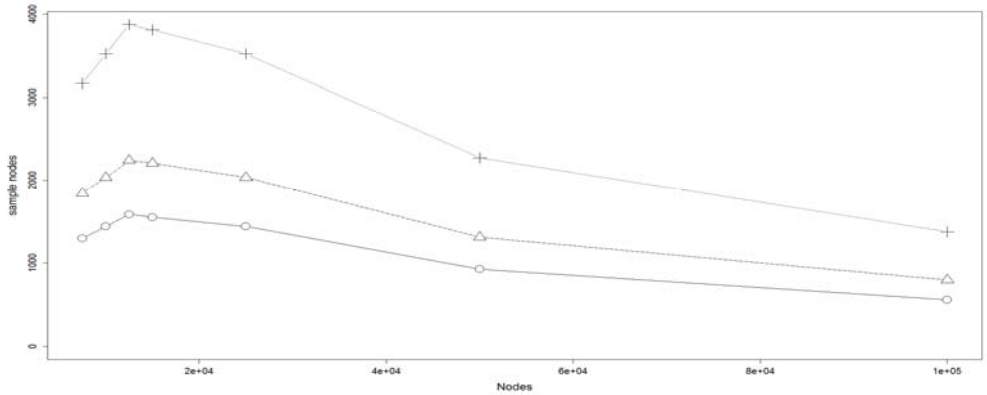
Circle C.I 90% , Triangle C.I 95%, Cross C.I 99%

(a) Algorithm 1 and 2



Circle C.I 90% , Triangle C.I 95%, Cross C.I 99%

(b) Algorithm 3

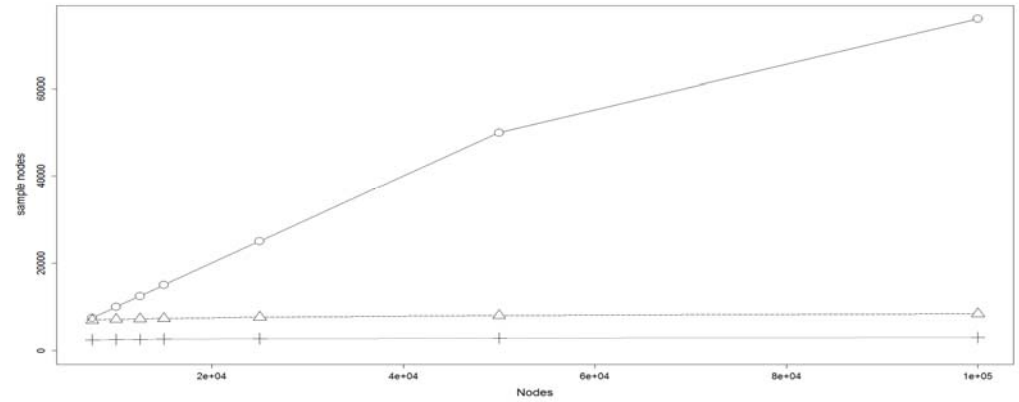


Circle C.I 90% , Triangle C.I 95%, Cross C.I 99%

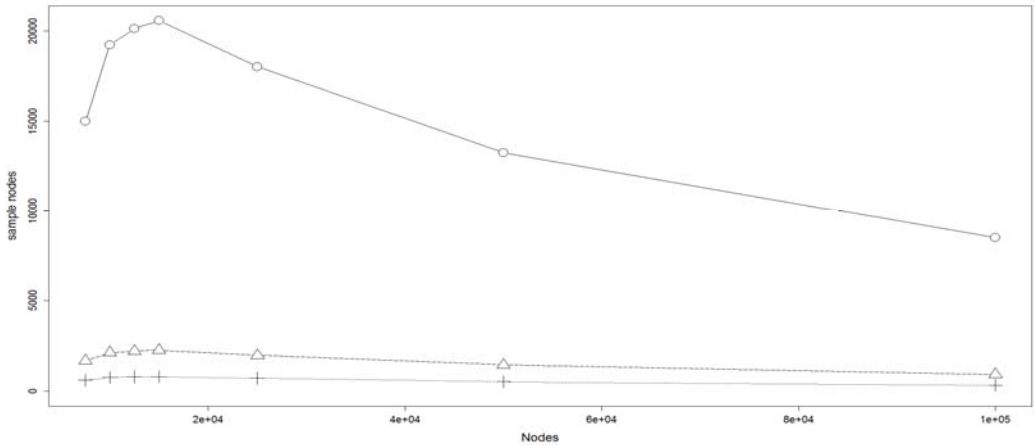
(c) Algorithm 4

Though the increase in confidence interval increases the sample size, the increase in the sample size is nonlinear and dependent on the Algorithm. The sample size requirements decline for Algorithm 3 and 4 due to the decrease in the variation in the SPL for a RN (Figure 7).

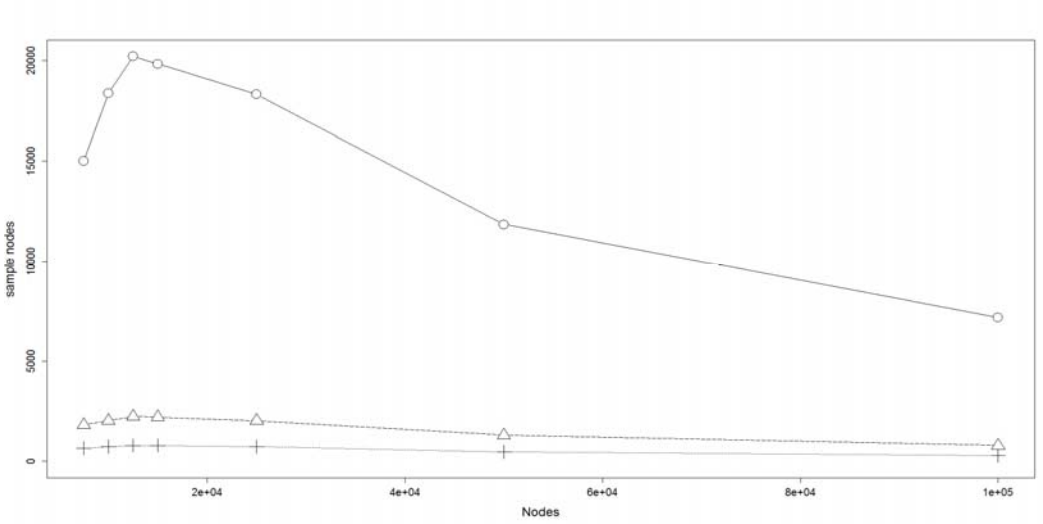
Figure 7 Impact of Change in Precision on the Sample Size when we use (a) Algorithm 1 or 2, (b) Algorithm 3, and (c) Algorithm 4.



Circle 0.01 error , Triangle 0.03 error, Cross 0.05 error
(a) Algorithm 1 and 2



Circle 0.01 error , Triangle 0.03 error, Cross 0.05 error
(b) Algorithm 3



Circle 0.01 error , Triangle 0.03 error, Cross 0.05 error

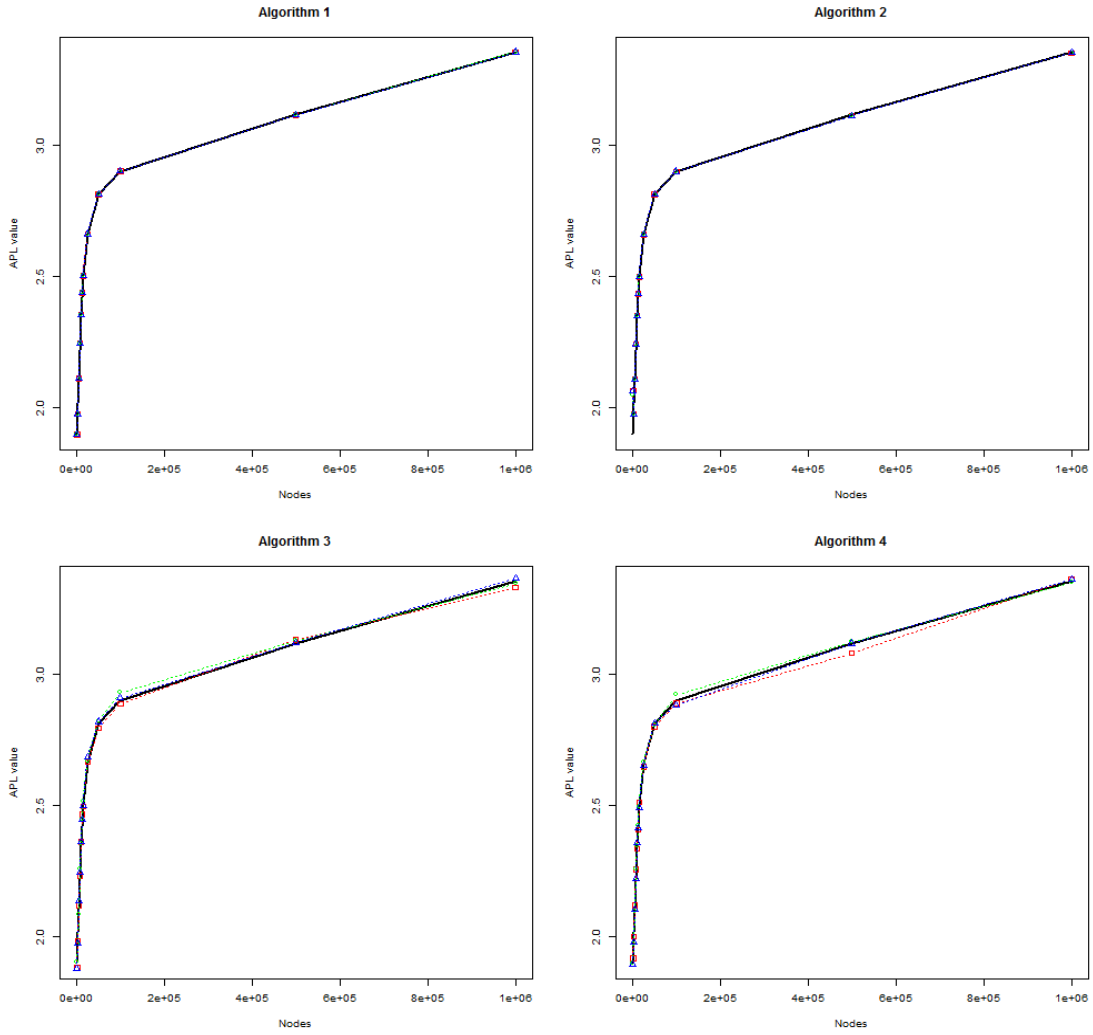
(c) Algorithm 4

Similar to confidence interval, increase in precision leads to an increase in the sample size; however, this increase is nonlinear and dependent on the algorithm. The sample size requirements decline for algorithm 3 and 4.

Impact of Network Structure on Precision

The proposed algorithms were tested on RN, WS, and BA networks with precision set at 3%. The results are presented in Figures 8-10. The estimation of the four algorithms show that the precision of the point estimates vary depending on the network topology.

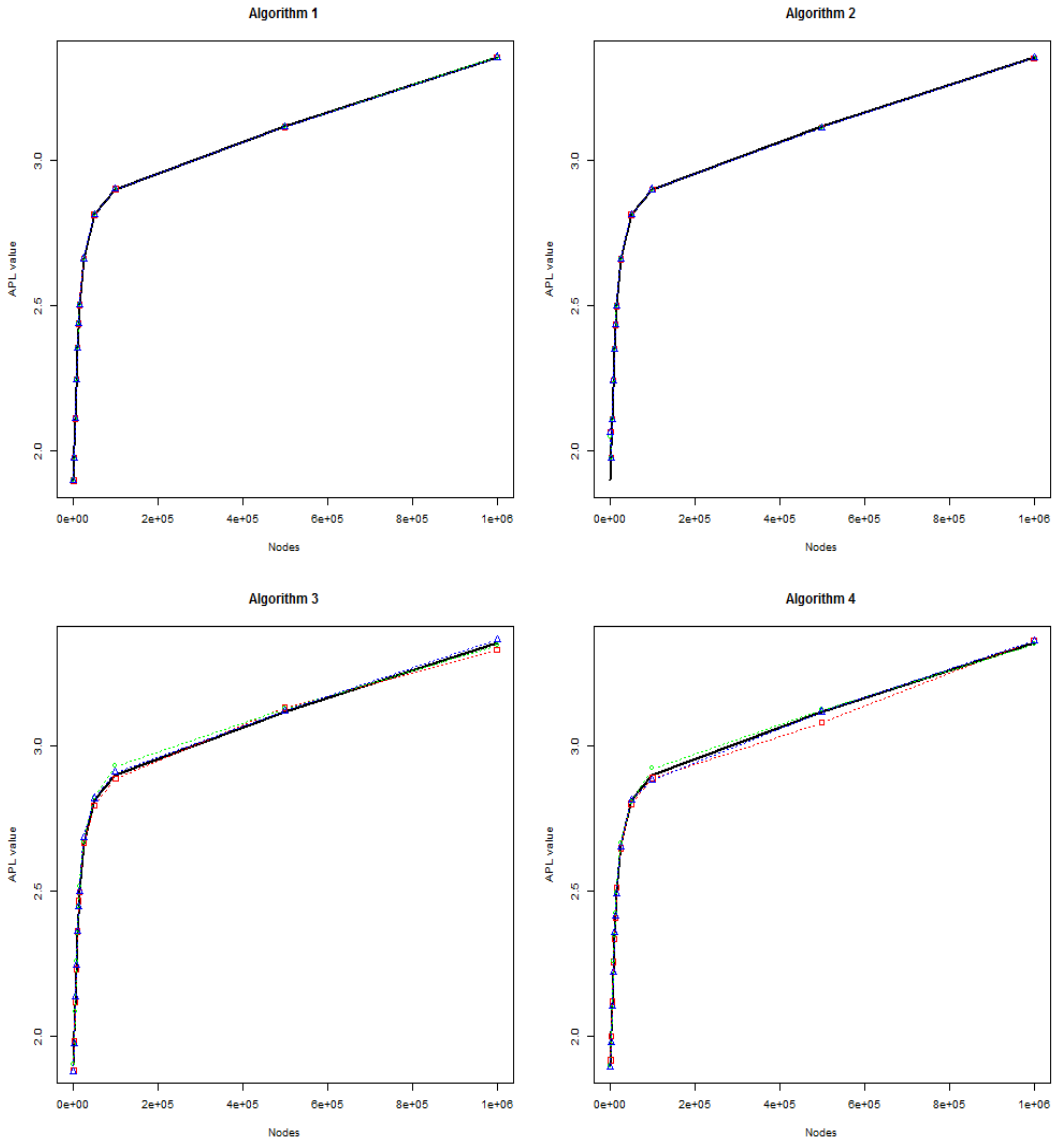
Figure 8 Precision of the proposed algorithms in recovering true APL of a RN.



Black - population value, Red -90% C.I, Green -95% C.I, Blue -99%

All four algorithms yield precise results when the underlying network is RN. This results is consistent for all the different network sizes.

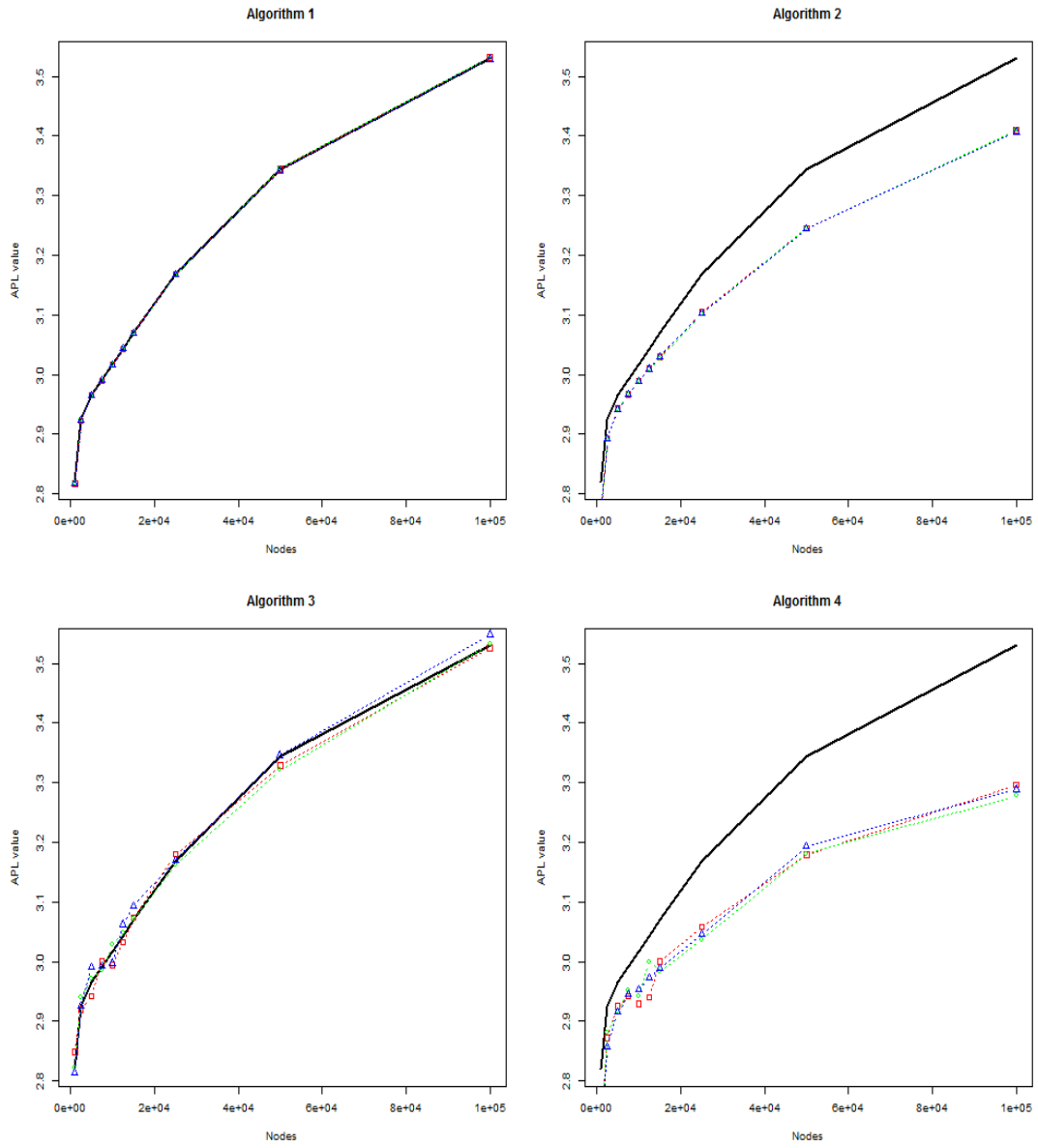
Figure 9 Precision of the proposed algorithms in recovering true APL of a RN.



Black - population value, Red -90% C.I, Green -95% C.I, Blue -99%

As in the case of RN, all four algorithms produce valid estimates in case of WS network as well.

Figure 10 Precision of the proposed algorithms in recovering true APL of a BA Network.



Black - population value, Red -90% C.I, Green -95% C.I, Blue -99%

In the case of the BA networks, the APL estimates' precision shows two different results for the four estimation methods. Algorithms 1 and 3 present a trajectory that is consistent with the population value of the APL as the network size increases while Algorithms 2 and 4, that use a selection process based on the degree of the nodes, perform

poorly. These results suggest that selecting highly connected nodes causes bias in APL estimation. Figure 10 presents these trajectories for the four methods.

Parallelization of Sampling

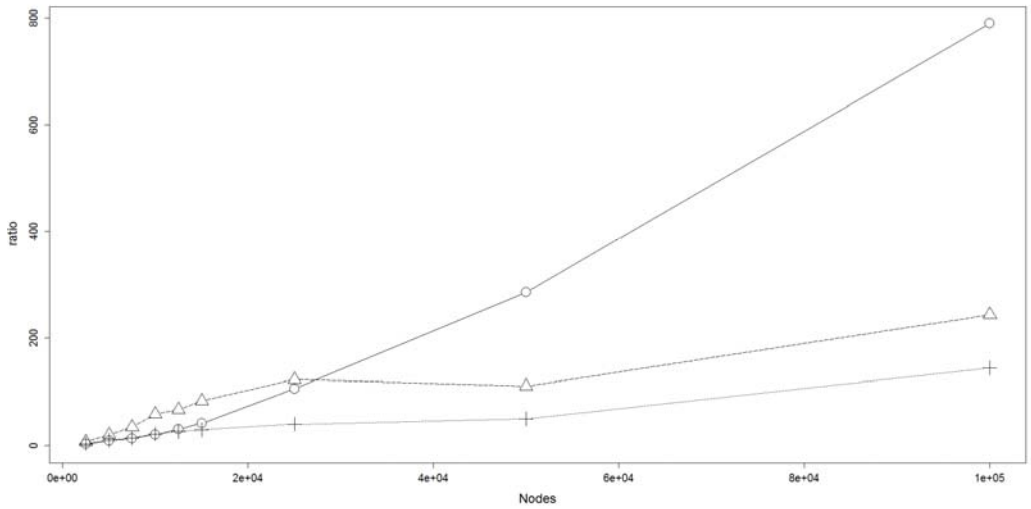
We explore the scalability of Algorithm 3 across the 3 network types, namely RN, WS, and BA. The speed up performance of the APL algorithm increases as the number of cores goes from 1 to 8 for all three network types. Table 20 depicts these results. Only in the case of small networks (size 2,500 nodes) the communication among cores hinders the gains. The larger gains are observed for the larger network sizes. In the case of the million size network, going from 1 core to 8 cores speed up the calculations in the case of RN by 7 times, for WS by 3 times and BA by 10 times.

Table 20 Scalability of APL Algorithm 3 on Multi-Core PCs (Time in seconds)

Network	**RN**				**WS**				**BA**			
Cores	*1C*	*2C*	*4C*	*8C*	*1C*	*2C*	*4C*	*8C*	*1C*	*2C*	*4C*	*8C*
2500	2.4	2.3	3.2	4.8	7.7	5.3	5.3	7.0	6.2	4.3	4.6	4.2
5000	9.7	6.3	6.6	5.9	15.6	8.6	8.8	9.2	7.5	5.2	5.6	5.2
7500	29.5	17.2	11.3	7.8	29.5	17.5	14.3	14.3	10.4	7.4	7.1	6.7
10000	53.9	30.2	21.0	9.0	49.4	27.3	21.0	17.2	16.0	10.9	9.4	7.1
12500	78.3	43.9	29.1	9.8	65.5	40.0	26.7	22.3	29.1	16.4	12.7	10.0
15000	96.3	55.4	35.0	11.0	87.1	55.1	34.2	28.8	44.5	27.4	18.4	11.5
25000	145.3	80.8	51.2	13.0	168.4	90.2	61.7	55.7	164.8	87.9	58.8	20.2
50000	178.7	98.7	61.1	15.7	426.2	219.8	152.7	144.4	489.7	271.1	180.9	72.4
100000	175.2	96.9	67.6	24.4	742.8	435.3	256.4	214.9	1520.0	927.6	673.2	142.3

The speed-up achieved using Algorithm 3 versus using the population measurement based approach for measuring APL is presented in Figure 11.

Figure 11 Speed up achieved by using Algorithm 3



Circle RN, Triangle BA, Cross WS

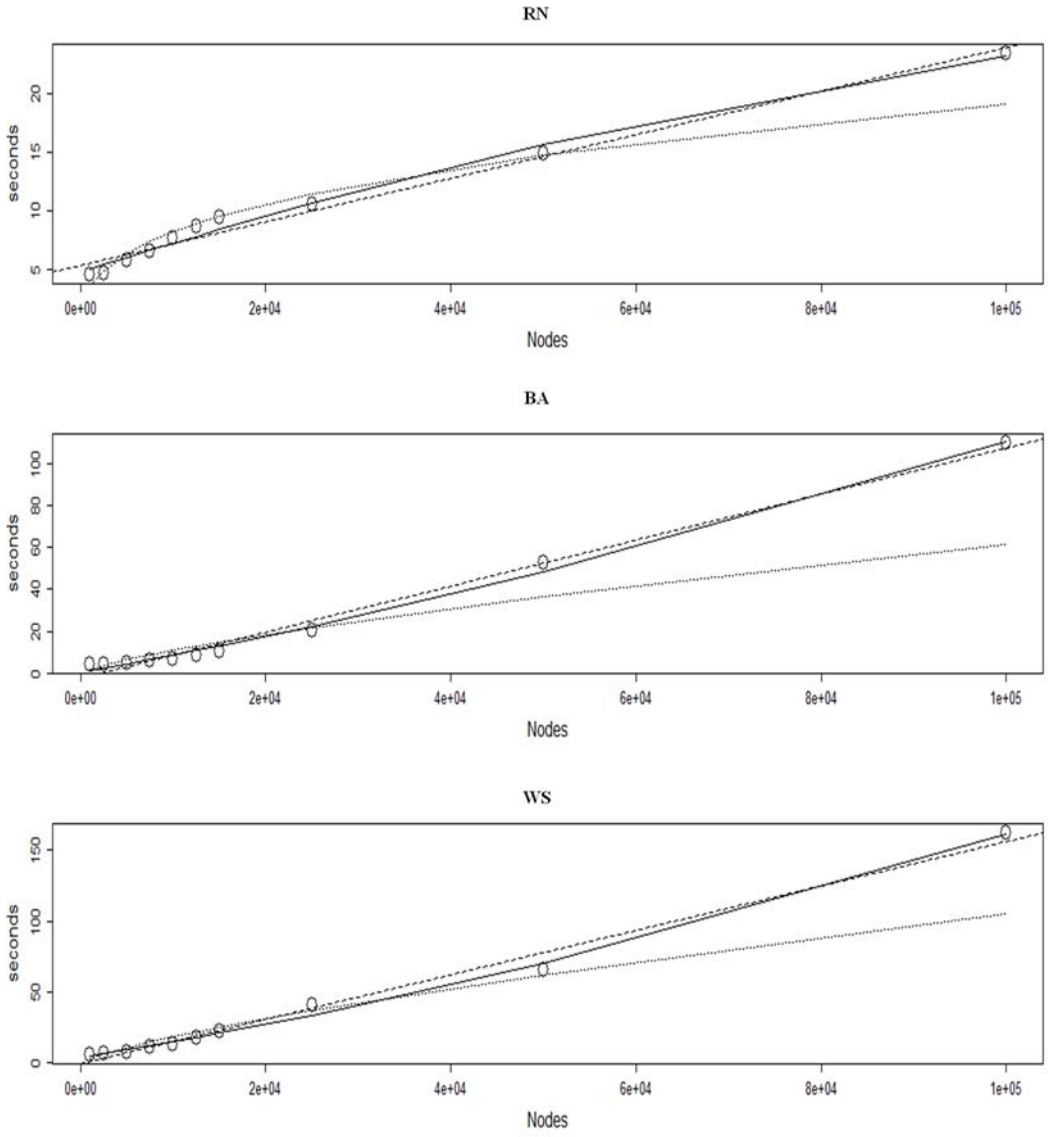
The speed up is dependent on the network topology with speed up in the RN being 750, WS being 145, and BA is at 245 for a 100,000 node network.

The authors in (Madduri et al. 2007) have shown that the computational time required to estimate APL scale as $V(V + E')$ where V is the number of vertices and E' a function of the number of edges in the network. E' itself scales as $O(V^{\gamma})$ where $1 \leq \gamma \leq 2$. In the case of our best performing algorithm, given that the modified Dijkstra (1956) is no more than a breadth-first search, a complete shortest path calculation takes $O(E + V)$. Thus, the complexity of calculating the APL via algorithm 3 is given by $O(r(E + V))$ where $r = \left(\frac{Z_{\delta S}}{\varepsilon}\right)^2$ is no more than the number of pair of nodes required to explore. In this case, the worst performance scenario for algorithm 3 is to explore all pair of vertices with complexity $O(V(E + V))$ yielding a quadratic scaling.

Using our experiment set up, we fit a linear, quadratic and logarithm regression model to the computational time obtained under algorithm 3. In all the cases, the quadratic fit had

the lowest mean squared error and highest fit. These results confirm a quadratic scaling across all network types. Figure 12 shows this pattern.

Figure 12 Computational Complexity of Algorithm 3



Lines: solid quadratic, dash linear, dot logarithmic

Summary of Computational Study

The sampling based approaches yield precise estimates even when the sample sizes used for the estimation scale linearly with increase in the network size. Algorithms 1 and 3 show the best performance in terms of precision while Algorithm 3 is significantly better than Algorithm 1 from a computational time and effort requirements perspective. Further,

- The sample size required for achieving a required precision and confidence depends on the algorithm used to generate the graph. It therefore depends on factors such as the degree distribution, cluster size distribution.
- Even though the sample size may be dependent on the degree distribution, incorporating degree dependent sampling strategies perform worse than strategies that are purely random.
- The sample size requirements don't necessarily increase as the network size increase. In fact, for networks such as the ER graphs and WS graphs, the sample size decreases when the network size increases.

Empirical Analysis of Real Networks

Real networks can often behave significantly differently from synthetic networks. We therefore test the proposed algorithms on four real life social networks that were used in (McAuley and Leskovec 2012). The descriptive statistics and the APL estimates for the entire population are presented in Table 21.

Table 21 Real life social networks characteristics and actual APL

Network	Nodes	Edges	APL Metrics	
			APL	Computational Time (sec)
Facebook	4039	88234	3.6078	53.46
Citation – Astrophysics	18772	198110	4.1784	737.22
Enron	36692	183831	3.9000	1451.96
Twitter	81306	1768149	3.5813	22924.56

We estimate the APL using the four proposed algorithms with a 0.03 error margin and a confidence level of 95%.

Figure 13 APL Estimates Using the Proposed Algorithms

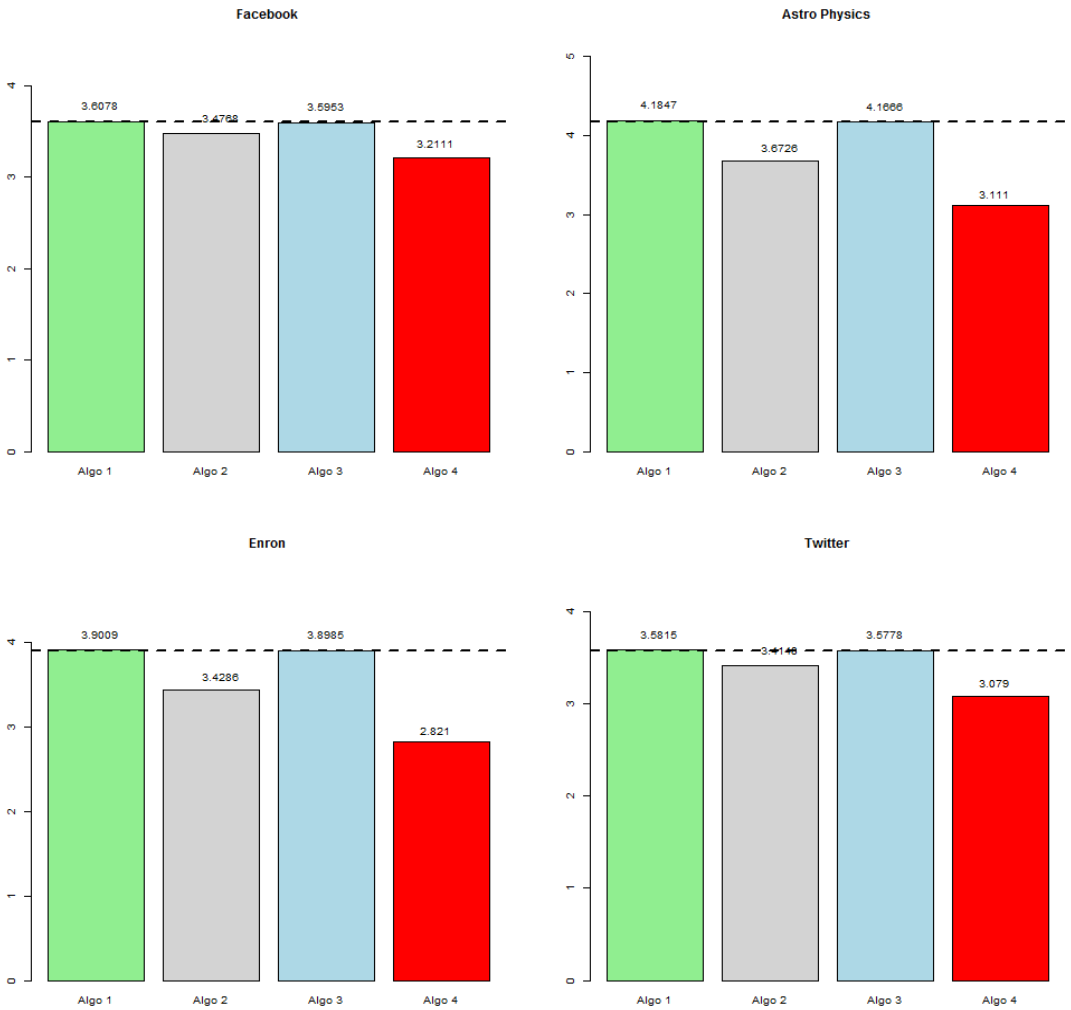
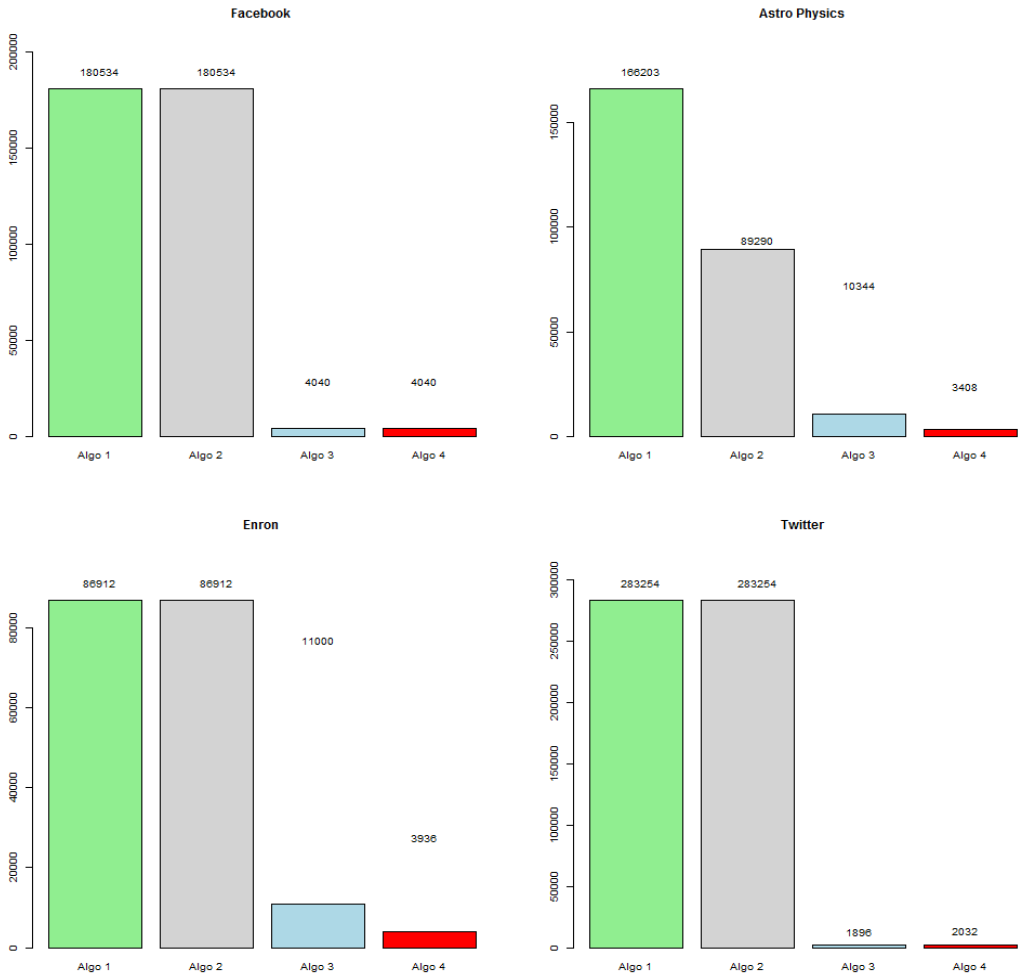


Figure 12 is consistent with what we found in the computational study; Algorithm 1 and 3 have the best results for all four networks. For these two methods, the potential error is below the 3% target marked. This result confirms that when dealing with social networks, where a mixture of randomness and preferential attachment is always present, algorithm 1 and 3 allow for a precise estimation of the APL.

The sample sizes used for estimating the APL metrics for the four networks are presented in Figure 14.

Figure 14 Sample Size Estimates for the Proposed Algorithms



Note that the sample size required by Algorithm 1 is about 44 to 153 times larger than the sample size required by Algorithm. Further, this difference is bigger as the network size increases. The estimates of the standard deviation of the SPLs are presented in Figure 12 while the estimates of the computational time are presented in Figure 14.

Figure 15 Estimates of the Standard Deviation in SPL for the Four Real Life Networks

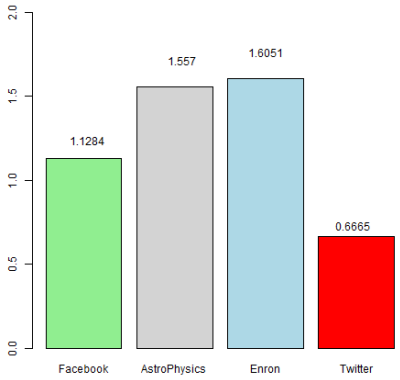


Figure 16 Speed up achieved using Algorithm 3 for estimating APL

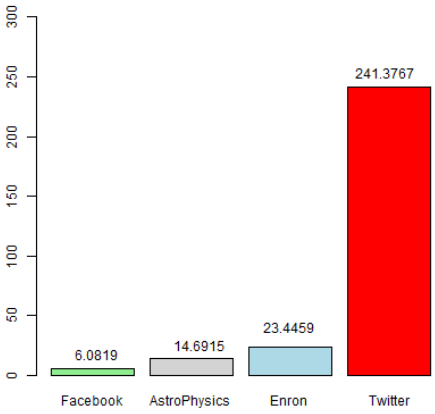
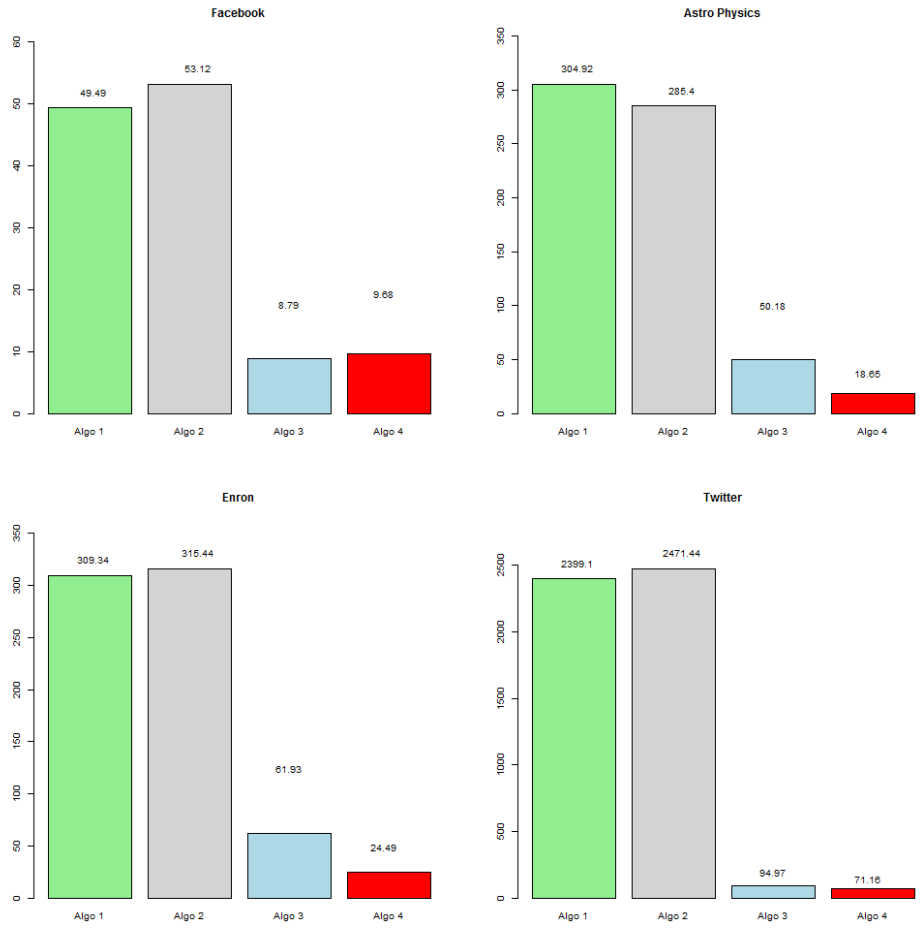


Figure 17 Computational Time (in Seconds) for Four Real Life Networks



The speed up increases as the sample size increases.

Conclusions

On precision of APL estimates: Algorithm 1 has the highest precision across all the synthetic and real life networks. Nonetheless, Algorithm 3 always achieves a precision that is always within +/-1% of the population value and less than the desired level of error. Furthermore, the simulation results show that Algorithm 3 is unbiased. Algorithm 2 and 4 yield biased results.

On the sample size: Algorithm 1 uses a much larger sample size as compared to Algorithm 3 yet results are comparable. This makes Algorithm 3 more efficient. In addition

to the sampling strategy, the topological characteristics of the network play an important role in the determination of the sample size. This means that the SPL distribution is more homogenous, therefore a smaller sample size and fewer explorations are needed for the estimation. This fact is true not only for synthetic networks, but also for real life networks. This may be a sign of the phenomena reported by (Albert and Barabási 2002) and later confirmed related to the SPL distribution and diameter.

On computational time: Our experiment shows that the sample size and computing times scale linear and logarithmically in the case of Algorithm 3 as the network size increases. Since Algorithm 1 oversamples, the computational time increases exponentially, confirming its inefficiency. In the case of synthetic and real networks, our procedure produces large gains. For a real life mid-size network (around 80,000 nodes) we report speed up ratios of 250 times. Algorithm 3 reduces the APL calculations under 3 minutes for all cases.

The proposed sampling based APL estimation can benefit significantly from integration with graph partitioning algorithms.

Chapter 5: A Functional Stochastic Opinions Dynamic Model over a Network

Overview

The ability to track how opinions about people, place, issues or things are evolving over time is important in a multitude of domains such as engineering, social sciences, and management science. This ability enables firms to identify emerging trends and dominant opinions, opinion leaders, influencers and stubborn agents, and take proactive/predictive actions. This work focuses on expanding the concept of the extant opinion dynamics literature by developing a new theoretical framework based on functional hereditary systems. We build a Stochastic Opinions Dynamics Model (SODM) where the opinion of each agent in the network is modeled as a probability function for every point in time. The concept of opinion updates is reformulated using functional stochastic difference equations so the evolution of opinions can be uniquely identified stochastic processes for each agent.

Further, we extend the concept of consensus into the probabilistic framework based on the stability concepts of (Kovalev, Kolmanovskii, and Shaikhet 1998, Paternoster and Shaikhet 2000). We propose the existence of global and local consensus. In the global concept, all the opinion distributions need to converge in probability to a single opinion. In the former, this is required only for a subset. Agents arrive to consensus (i) when their individual opinions are bounded, (ii) converge individually to a unique probability distribution, (iii) and the influence matrix admits a Jordan Normal form.

Agent based simulation are used to illustrate our main theoretical results, and to analyze the role of influential agents and stubborn agents. Our main findings suggest that (i) when a spanning tree is present in the network, a global asymptotic consensus distribution exist; however, the probability that all agents reach a fixed point (a deterministic consensus

value) is zero; (ii) when no global consensus exists, our model allows us to find the number of local asymptotic consensus distributions and the common regions among these groups. This facilitates the comparison, characterization and quantification of opinions between groups; (iii) only a global influencer is capable of imposing his opinion; (iv) local influencers have no effect on the final distribution and time for consensus; (vii) if an agent is almost stubborn and the contact network exhibits preferential attachment properties, only local consensus is possible.

The document is organized into four sections. Section III develops the construction of our probabilistic opinion model, formalizing its definitions, interaction, and advances with respect to the current model. Section IV details our simulation framework; Section V outlines the conclusions and future research path.

Stochastic Opinion Dynamics Model (SODM)

Notation

General notation. We denote the set of real numbers by $\mathbb{R}$ and the set of integers by $\mathbb{Z}$. The triad $(\Omega, \Im, \mathrm{P})$ denote a probability space. We denote the cardinality of a set A as $card(A)$. Probability density functions are denoted by $f_{\cdot}$ and cumulative functions by $F_{\cdot}$. The set of time is given by $t \in \mathbb{Z}$.

Agent. Each agent is represented by a node of the network. Agents can only interact with their neighbors and some general environmental conditions. The set of agents throughout the document is represented by $i \in I,\ i = 1, \dots, j, \dots J \in \mathbb{Z}$.

Social Network. A social network expresses the underlining correlations, interactions and structures of a set of agents (nodes). Inside this setting, agents interact with each other sharing opinions. This interaction whether it may seem limited to a set of neighbors or local

vicinity; in fact, it depends on the interactions that take place along the whole network structure – Newman (2003). The notation $\mathbb{S}(N, E)$ represents a social networks constituted of set of nodes $n \in N$ and arcs $e \in E$. Each individual arc connects uniquely a pair of agents $e_{i,j}$. The topology of $\mathbb{S}$ is fixed in time. The adjacency matrix is denoted as D. In our setting, this structure represents a contact network for the agents. This network is a static directed graph[8].

The model

Definition 1 (Probabilistic opinion space). Let the triad $(\Omega_i, \Im_i, \mathrm{P}_i)$ be the set of opinions Ω_i with a □-algebra $\Im_i$ and a probability measure P_i. These three elements define formally the probabilistic opinion space of agent i. This concept expresses that each agent i is has his own collection of opinions with a particular probability measure.

Definition 2 (Opinion Profile). Let P_i be the probability measure defined on $(X_{i,t}, \mathfrak{H}_{i,t})$.An opinion for agent i at time t is a random variable $x_{i,t}: \mathrm{X}_{i,t} \to \mathbb{R}$ such that $\{v: x_{i,t}(v) \leq y\} \in \mathfrak{H}_{it}, \forall\, y \in \mathbb{R}$.

Definition 3 (Influence contact network). Let $\mathbb{S}(N, E)$ be the contact network structure for a set of agents (nodes) N and a set of arcs E such that $e_{i,j} \in E$. Let $\mathcal{I}(\mathrm{V}, L)$ be influence network $\mathcal{I}$ of the agents where the values of the edges for agents i and j are given by:

$$l_{i,j} = \begin{cases} 0 & if\ e_{i,j} = 0 \\ a_{i,j} & if\ e_{i,j} = 1 \end{cases} \quad for\ 0 \leq a_{i,j} \leq 1$$

[8] The terms network and graph are used interchangeably

Under this framework, let A be the adjacency matrix of this network such that $l_{i,j} \in A$. The influence network is a weighted directed network fixed in time.

Definition 4 (Opinion update rule). For each agent i at time t let $r_{i,t}(.)$ be a Borel function such that $x_{i,t} = r_i(x_{1,t-1}, \dots, x_{i,t-1}, \dots x_{M,t-1}, y_{k,t})$, for agent i considering all his $M \leq card(N)$ neighbors. A valid functional form for the update rule (LUR) of agent i is given by:

$x_{i,t} = \sum_{j=1}^{M} a_{i,j} x_{i,t-1} + \sum_{k=0}^{K} b_k y_{k,t} \ \ for\ a_{i,j} \geq 0$

where $a_{i,j} \in A$

The set of opinions of agent i at time t is evolving due to two facts: (i) the opinion of agent i and his neighbors at time $t-1$ and (ii) an external source of information $y_{k,t}$. All these facts are expressed formally in the opinion update rule. The Borel function condition is needed to take as inputs random variables, and have as an output an updated random variable.

Definitions 1 to 4 constitute our general functional stochastic opinion dynamics model (SODM). Our interest lies in the study of the dynamics of this system. For this reason, we define a new concept for consensus.

Definition 5 (Consensus). Global consensus is reached if $lim_{t\to\infty} Fx_{i,t} = lim_{t\to\infty} Fx_{j,t} = Fx_c$ for all agents $i \in I$. Local consensus is reached if this condition only holds for a subset $M \subseteq I$.

The new concept of consensus implies that as time passes and agents update their opinions, a common opinion distribution may arise for all the agents in the network. Our next step is to construct the conditions for global and local consensus.

Lemma 1. Let V be a nonnegative functional. The zero solution $(x_{i,0})$ of the SODM for agent i is mean square stable if:

$$\mathbb{E}[\boldsymbol{V_{i,t}}(\boldsymbol{x_{i,t}}, \dots, \boldsymbol{x_{i,0}}) \leq \boldsymbol{c_1} \parallel \boldsymbol{f} \parallel^{\boldsymbol{2}} \quad (4a)$$

$$\mathbb{E}[\Delta \boldsymbol{V_{i,t}}] \leq \boldsymbol{c_2 E}[\boldsymbol{x_{i,t}}]^{\boldsymbol{2}} \quad \boldsymbol{t} \in \mathbb{Z} \quad (4b)$$

$$\boldsymbol{c_1} > \boldsymbol{0}\,, \boldsymbol{c_2} > 0 \text{ and } || \text{ a suitable function norm} \quad (4c)$$

Proof. This construction comes from theorem 1 and 2 of (Paternoster and Shaikhet 2000).

The result from this lemma says that any construct using this setting have opinions as individual probably functions evolving in time, but bounded (trackable).

Lemma 2. Let $\gamma < \infty$. The $(x_{i,0})$ for agent i is stable in probability if $\sum_{j=1}^{J} a_i + \sum_{k=0}^{K} b_k \leq 1$.

Proof. This construction comes from theorem 3 of (Paternoster and Shaikhet 2000).

This result implies that the opinion of agent i evolves in time, guided by his past opinions and the opinion of his peers, to a fixed probabilistic distribution.

Corollary 1. Let $\{x_{i,n}\}c_{n=1}^{\infty}$ be the sequence of random variables for agent i obtained under SODM. Then, an asymptotic opinion distribution for agent i has been reached.

Proof. Given that the zero solution $(x_{i,0})$ is stable in probability, it is bounded by the limiting function x_i. In addition, since LUR is a valid Borel function, x_i is a random variable with a valid pdf denoted by f_x. Given this fact, the limiting opinion distribution of agent i exists, so we have $lim_{t\to\infty} F x_{it} = F x_i$.

Lemma 3. If the sequence of row stochastic matrix $\{A^t\}_{t=1}^{\infty}$ converges, then $lim_{t\to\infty}A^t = lim_{t\to\infty}PL^tP^{-1} = A^*$ where P is matrix composed of the eigenvectors of A , L is a diagonal matrix where each diagonal element has an eigenvalue of A , and $a^*_{1,j} = a^*_{2,j} = \cdots a^*_{j,j}$ for all $j \in I$.

Proof. This result comes directly from (Condon and Saks 2004).

Lemma 4. Let X_1 and X_2 be two random variables with characteristic functions $\varphi(x_1)$ and $\varphi(x_2)$. If $(x_1) = \varphi(x_2)$, then $Fx_1 = Fx_2$.

Proof. It is a well-known result of probability theory; it follows directly from (Resnick 2013)

Theorem 1. In a SODM, global consensus exists if:

(a) Opinion distributions $x_{i,t}$ are stable in probability for each agent

(b) The functions $y_{k,t}, \ldots, y_{K,t}$ are a valid probability functions that comply with mean square stability and stability in probability

(c) For $\tilde{A} = \begin{bmatrix} A & B \\ 0 & B \end{bmatrix}$ a square row stochastic matrix $lim_{t\to\infty}\tilde{A}^t = \tilde{A}^*$

Proof. The global definition of consensus require all the opinion distributions of all the agents to be equal; our strategy is based on the construction of a sequence of characteristic functions for each agent, and show that the sequence converges to the same characteristic function for all the agents. At time t, let us have a SODM for all the network given by:

$$X_t = AX_{t-1} + BY_t = A^tX_0 + BY_t \tag{5}$$

Let us re-write Eq. (5) as:

$$\tilde{X}_t = \tilde{A}^t\tilde{X}_0 \tag{6}$$

where $\tilde{X}_0 = \begin{bmatrix} X_0 \\ Y_t \end{bmatrix}$. Taking limit in (6) we have:

$$lim_{t\to\infty}\tilde{A}^t\tilde{X}_0 = lim_{t\to\infty}\tilde{A}^t lim_{t\to\infty}\tilde{X}_0 \quad (7)$$

By Lemma 1 and Lemma 2 (condition (a)) we know that individual opinion distributions are bounded and each one converges to a final distribution. In addition, by construction of SODM $y_{k,t}$ are valid probability functions, bounded and stable in probability by condition (b). So, we have

$$lim_{t\to\infty}\tilde{X}_0 = \tilde{X}_c \quad (8)$$

where $\tilde{X}_0 = \begin{bmatrix} X_c \\ Y_c \end{bmatrix}$.

By Lemma 3 (condition (c)), we notice that $\tilde{A}$ is a row stochastic matrix, then it can be written in Jordan canonical form. Its limit is given by:

$$lim_{t\to\infty}\tilde{A}^t = \tilde{A}^* \quad (9)$$

Putting Eq.(8) and Eq.(9) together we have

$$lim_{t\to\infty}\tilde{A}^t\tilde{X}_0 = \tilde{A}^*\tilde{X}_c \quad (10)$$

Then, our stochastic system as $t \to \infty$ converges to $\tilde{A}^*\tilde{X}_c$. Now we show that all the functions in the system $\tilde{A}^*\tilde{X}_c$ are the same.

The characteristic function of the opinion distribution of agent 1 when $t \to \infty$ is given by:

$$E\left[e^{i\tau\tilde{x}_{1,c}}\right] = \varphi(\tilde{x}_{1,c}) \quad (11)$$

where $\tilde{x}_{1,c} = a^*_{1,1}x_{1,c} + a^*_{1,2}x_{2,c} + \cdots + a^*_{1,n}x_{J,c} + b_{1,1}y_{1,c} + \cdots + b_{1,1}y_{1,K}$

Repeating this construction for the remaining $J-1$ agents we have the sequence of characteristic functions

$$\{\varphi(\tilde{x}_{i,c})\}_{i=1}^{J} \quad (12)$$

Let us notice that this sequence is constant and equal to the characteristic function $\varphi(x_c)$ since

$$\tilde{x}_{1,c} = a^*_{1,1}x_{1,c} + a^*_{1,2}x_{2,c} + \cdots + a^*_{1,J}x_{n,c} + b_{1,1}y_{1,c} + \cdots + b_{1,1}y_{1,K} = \tilde{x}_{2,c} =$$

$$a^*_{1,1}x_{1,c} + a^*_{1,2}x_{2,c} + \cdots + a^*_{1,J}x_{J,c} + b_{1,1}y_{1,c} + \cdots + b_{1,1}y_{1,K} \ldots = \tilde{x}_{J,c} = x_c \qquad (13)$$

Therefore, all the characteristic functions for all the agents are equal. By Lemma 4 this implies that $lim_{t\to\infty}Fx_i = lim_{t\to\infty}Fx_{j,t} = Fx_{c,t}$ for all agents $i \in I$. Consensus has been reached.

Corollary 2. Under a SOMD $X_t = AX_{t-1} + BY_t$, let $\mathcal{E}$ be the extended influence network of the system with adjacency matrix $\tilde{A} = \begin{bmatrix} A & B \\ 0 & B \end{bmatrix}$. Let A and B comply with

Lemma 3. If $\tilde{A}$ has a spanning tree, and both the opinion distributions $x_{i,t}$ and external functions $y_{k,t}, \ldots, y_{K,t}$ are stable in probability; as $t \to \infty$ the agents reach consensus.

Proof. Since the network has a spanning tree each row of $\tilde{A}$ will have at least two element different than zero by rows. Given that A is row stochastic by construction, by Lemma 3 $lim_{t\to\infty}\tilde{A}^t = \tilde{A}^*$. Then, conditions (a) to (c) for consensus are met and the result comes from theorem 1

Corollary 3. If global consensus is reached in SOMD, the mean opinion of the agents converges to $\mu_c \in \mathbb{R}$ if $\mathbb{E}[x_c]$ exist.

Proof. For agent i, there exist pdf given by $fx_{i,c} = \frac{dFx_{i,c}}{dx_{i,c}}$, so $\mathbb{E}[x_{i,c}] = \int (fx_{i,c})(x_{i,c})dx_{i,c} = \mu_{i,c}$. Since the agents have reached consensus, $fx_{i,c} = \frac{dFx_{i,c}}{dx_{i,c}} =$

$\frac{dFx_{j,c}}{dx_{j,c}} = fx_{j,c}$ for all $j \in J$. Then $\mu_{1,c} = \cdots = \mu_{J,c} = \mu_c$.

Corollary 4. If a SOMD complies with theorem 1, an expression for global consensus is given by $x_c = A^* x_0$.

Proof. It follows directly from Eq. (10) of theorem 1.

Theorem 2. In a SODM, local consensus exists if for a subset of agents $M \subseteq I$:

(a) Opinion distributions are stable in probability for $j \in M$

(b) The functions $y_{k,t}, \ldots, y_{K,t}$ are a valid probability functions that comply with mean square stability and stability in probability

(b) $A_J = \Pi' A \Pi$ is a row stochastic matrix with a spanning tree where Π is a selector matrix with column vector elements $\pi_j' = [0\ 0\ \ldots 1 \ldots 0]$ with 1 only in the j position for $j \in M$ and $\tilde{A} = \begin{bmatrix} A & B \\ 0 & B \end{bmatrix}$.

Proof. We have two cases. When $M = 1$, the SODM depends of only agent 1. In this case conditions (a)-(c) are met trivially by lemma 1 and 2, so consensus is reached. In fact, this is the only case when global consensus and local consensus are equivalent.

When $1 < M < J$, we use a similar strategy to theorem 1. Without losing generality, let us assume that $B = 0$. At time t, the opinion of the M agents under consideration is given by:

$$X_{j\in M,t} = A_J X_{j\in M,t-1} = {A_J}^t X_{j\in M,0} \tag{14}$$

As $t \to \infty$ Eq.(14) becomes $lim_{t\to\infty} {A_J}^t x_{j\in M,t}$. By Lemma 1 and Lemma 2 we know that individual opinion distributions are bounded and converge to a final distribution, so this still holds for every individual opinion distribution in the subset M. Moreover, by Corollary 2 we know that a row stochastic matrix with a spanning tree can be written in

Jordan canonical form and by Lemma 3 its limit exists. These arguments imply that $lim_{t\to\infty} {A_J}^t X_{j\in M,t} = {A_J}^* X_{j\in M,c}$.

Let us construct the sequence of characteristic functions for the M agents:

$$\{\varphi_J(x_{j,c})\}_{j\in M} \tag{15}$$

This sequence is constant and bounded by $\varphi_J(x_c)$ since

$$x_{j,c} = \sum_{j\in M} a^*_{j,n}\, x_{j,c} = x_{k,c} \;\; \forall j,k \in M \tag{16}$$

Therefore, $lim_{t\to\infty} F x_{j\in M,t} = lim_{t\to\infty} F x_{k\in M,t} = F x_c$ for all agents $j,k \in M$. Local consensus has been reached for subset M.

Examples of SODM under LUR

In this section we analyze our model at the light of small practical examples by comparing its results to the current opinion model when opinions are real numbers and updated using a liner update rule.

Example 1. (Consensus) *Setting.* Let us have a 3 agents living in complete contact network. The influence matrix is given by.

$$A = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \tag{17}$$

Actual approach. Currently, the literature treats opinion dynamics as a deterministic system. At time 0, the opinion value for all the agents x_0, are drawn randomly from a common distribution D. In our case, let us have $x_0 \sim N(10,1)$. The initial opinion vector for the network is given by:

$$x_{,0} = \begin{bmatrix} 8.9 \\ 10.3 \\ 11 \end{bmatrix} \quad (18)$$

where each value is the opinion of agents 1,2 and 3 at time 0, respectively. After time 0, opinions are updated using the linear update. The literature is interested in the evolution of opinions and whether a common final consensus will be reached. For this purpose, we follow Olfati-Saber et. al (2007) and obtain the consensus value by calculating:

$$lim_{t\to\infty} A x_{t-1} = lim_{t\to\infty} [P L^t P^{-1}] x_0 = A^* x_0 \quad (19)$$

Since the limit exists, consensus (in this case a common fixed real value) is given by:

$$x^c = \begin{bmatrix} 9.1 \\ 9.1 \\ 9.1 \end{bmatrix} \quad (20)$$

SODM. As argued during the definitions section, in social networks –especially when opinions are generated by humans- a final immutable consensus may not be the case. Furthermore, each agent can have its own opinion space. These claims translate into the following characteristics for our model:

$$x_{i,0} \sim D_{i,0}() \quad (21)$$

where x_i is the opinion distribution at time 0 of agent i. In this case, we assume that the opinion distributions per each agent at time 0 are:

$$x_{1,0} \sim N(9,1), x_{2,0} \sim N(10,1), x_{3,0} \sim N(11,1) \quad (22)$$

The influence contact network A complies with Lemma 2, so our model is stable in probability. Furthermore, A has a spanning tree so global consensus can arise. We can construct characteristic functions for the three agents, and see that the sum of normal random variables is a normal random variable. Then, consensus is given by the asymptotic opinion distribution:

$$x_c \sim N(\mu = (9.8,9.8,9.8), \sigma = (1,1,1)) \quad (23)$$

This result has four implications: (i) all agents have reached consensus in the sense of convergence in probability that translates into convergence in distribution, (ii) consensus is not a fixed value, it is a whole well-defined pdf which is equal for all agents; (iii) furthermore, the probability of observing an specific value c from x_c is zero, (iv) if we observe only one draw for each agent from the global consensus distribution, these values may not actually be equal.

Example 2. (Consensus by groups and common region). *Setting.* The objective of this problem is to find out whether or not two different groups of friends can reach agreement between them without any coordination channel. The problem explores whether a common referential time for dinner for the two groups may exist. Let us have a network of 6 agents where two different groups of friends exist: group 1 – agents 1 to 3; and group 2 – agents 4 to 6. Matrix A summarizes the influence structure of the network at $t = 0$.

$$A = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \quad (24)$$

Actual approach. At time 0, the opinion value for all the agents x_0 are drawn randomly from a common $N(10,1)$. The initial opinion vector for the network is given by:

$$x'_{.,0} = [8.9 \quad 10.3 \quad 11 \quad 8.1 \quad 8.3 \quad 9.5] \quad (25)$$

In this case the limit $lim_{t\to\infty} A x_{t-1} = [PL^t P^{-1}]x_0$ does not exist since A^t does not convergence Ren (2005). Nonetheless, given the form of A, we can split it into 2 matrixes,

A^1 a matrix without zeros for agents 1 to 3, and A^2 a matrix for agents 4 to 6. The limit for each group can be found, so we have:

$$x_c^1 = \begin{bmatrix} 9.1 \\ 9.1 \\ 9.1 \end{bmatrix} \; and \; x_c^2 = \begin{bmatrix} 8.6 \\ 8.6 \\ 8.6 \end{bmatrix} \tag{26}$$

In this case, no common meeting time is possible.

SODM. Each agent has his own probability distribution of time available to meet at dinner, at time 0 they are given by:

$$x_{1,0} \sim N(9,1), x_{2,0} \sim N(10,1), x_{3,0} \sim N(11,1),$$

$$x_{4,0} \sim N(9.5,1), x_{5,0} \sim N(11,1), x_{6,0} \sim N(8,1) \tag{27}$$

For instance, the opinion of agent 1 at time 0 is given by a Normal distribution, with an average meeting time of 9 pm and variance of 1 hour.

In this case A complies with theorem 2, so all the agents interact with their neighbors up to the point when an agreement has been reached among all the members of each group. As our first conclusion, a global consensus does not emerge. On the other hand, two local consensus emerges and they are given by: $x_c^1 \sim N\big(\mu = (9.8,9.8,9.8), \sigma = (1,1,1)\big)$

$$x_c^2 \sim N\big(\mu = (8.8,8.8,8.8), \sigma = (1,1,1)\big) \tag{28}$$

Our results allow us not only to see an $L2$ difference between the consensus values as in other meet at the dinner problems where no common spanning tree exists among agents; but we can plot both final distributions and see the common area that is shared by both consensus distributions.

In practical terms, if the final goal is to come up with a coordination time between the two groups, the only feasible time to maximize the number of group member present on time for the dinner should not be a real number, but a time interval with an occurrence

probability for each of the groups. For instance, if only two hours are given as slack time before starting the dinner, these hours may be between 8:30 pm to 10:30 pm.

Example 3. (Only consensus by groups) *Setting.* Consider now the problem of finding whether two different political parties can reach consensus. Let us have a network of 6 agents where two different groups exist: political party A – agents 1 to 3; and political party B – agents 4 to 6. The discussion is centered on topic Z (i.e immigration reform). In real applications, an opinion index for topic Z can be available by construction following the guidelines of sentiment, economic or political indexes by Gaski et. al (1986), Gallup et. al (1999), Connor et. al (2010), Hebster et. al(2010), Tumitan et. al(2014), or Anderson et. al (2014). In the following application, our index is continuous and ranks from -1 to +1, where -1 represent extreme right wing policies on immigration, 0 represents lack of importance on the issue, and +1 represents extreme left wing policies[9].

Matrix A summarizes the influence structure of the network at $t = 0$.

$$A = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \tag{29}$$

Finally, it is worth noticing that no member of political party A places a positive weight on any opinion of any member of political party B. In a similar fashion, party members of B also proceed.

[9] Notice that policy blocks and ranges can be constructed, so numerical values can be clearly translated into well defined opinions.

Actual approach. At time 0, the opinion value for all the agents x_0 are drawn randomly from a common $N(0,0.05)$. The initial opinion vector for the network is given by:

$$x'_{.,0} = [-0.5 \quad -0.8 \quad -0.4 \quad 0.5 \quad 0.8 \quad 0.4] \tag{30}$$

In this case the limit $lim_{t\to\infty} Ax_{t-1} = [PL^tP^{-1}]x_0$ does not exist since A^t does not convergence Ren (2005). Nonetheless, given the form of A, we split it into 2 matrixes, A^1 a matrix without zeros for agents 1 to 3, and A^2 a matrix for agents 4 to 6. The limit for each group can be found, so we have:

$$x_c^1 = \begin{bmatrix} 9.1 \\ 9.1 \\ 9.1 \end{bmatrix} \; and \; x_c^2 = \begin{bmatrix} 8.6 \\ 8.6 \\ 8.6 \end{bmatrix} \tag{31}$$

SODM. All the agents express their opinions period after period. In period t, the distributions of the opinions per each agent at time 0 are given by $x_{1,0} \sim N(-0.5,0.05), x_{2,0} \sim N(-0.8,0.05), x_{3,0} \sim N(-0.4,0.05)$ and $x_{4,0} \sim N(0.5,0.05), x_{5,0} \sim N(0.8,0.05), x_{6,0} \sim N(0.4,0.05)$.

Based on Theorem 1, we conclude that global consensus does not emerge. Nonetheless, by Theorem 2 we can identify 2 groups and only local consensus emerge:

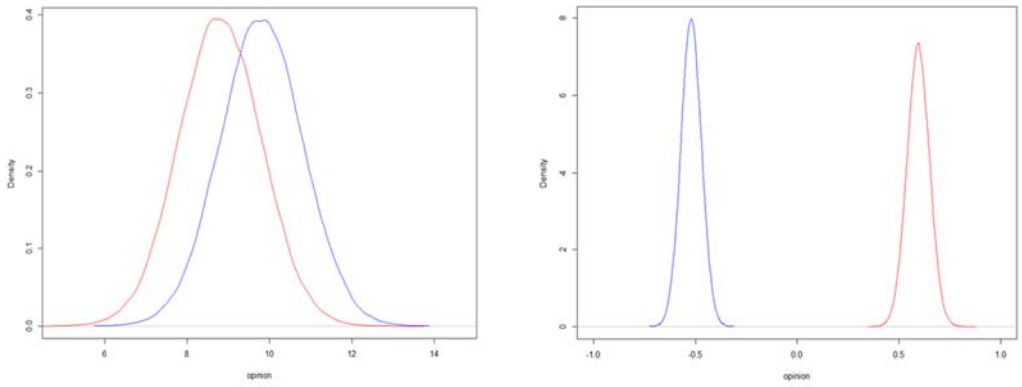
$$x_c^1 \sim N\big(\mu = (-0.52,-0.52,-0.52), \sigma = (0.05,0.05,0.05)\big)$$

$$x_c^2 \sim N\big(\mu = (0.59,0.59,0.59), \sigma = (0.05,0.05,0.05)\big) \tag{32}$$

In practical terms, if the final goal is to come up with a coordination policy between the two groups, the intersection region between the two final opinion (consensus) distributions would suggest the potential zone of agreement between these two groups. In this extreme case, not even a potential agreement zone exists with a small probability – close to zero. So no coordination is possible. Finally, notice that the low variability of opinions within each group causes the non-existence or a common region between groups

(contrasting this result with Example 2). In this case, from the construction of our SODM, if we want to enforce at least a common region between the agents, we will need to bring a 7th agent with a big influence on the opinion of one of the groups, but with a higher variance (flexibility) in his opinion. More solutions can be thought in this line, but are beyond the scope of this work.

Figure 18 Asymptotic distributions for groups 1 and 2



(a) Example 2 (a) Example 3

Agent Types

As noted in the examples from last section, it is relevant to the analysis of consensus the type of influence contact network that each particular SODM has. The adjacency matrix of this network is directly linked to the concept of agent types.

Definition 6. Let N be the set of n neighbors of agent i. An adopter type agent i of the influence contact network $\mathcal{I}$ has edges $l_{i,j} \in L: l_{i,j} = \frac{1}{n+1} \quad \forall j \in J \; and \; i \in V$.

This configuration represent the type of agent that is willing to listen and learn from his peers.

Corollary 5. In a complete network with only adopter type agents and no external influence, consensus arises after the first interaction among the agents.

Proof. Without loss of generality, let us assume that $B = 0$. Notice that in a complete network a spanning tree can always be found (i.e. construct the spanning tree by taking the elements $e_{i,i+1} \in \mathbb{S}, i \neq J$). By construction, the influence network has a spanning tree, therefore global consensus arises by Corollary 2. Now, notice that after the first interaction all the rows of the influence matrix are $\left[\frac{1}{J} \quad \dots \quad \frac{1}{J}\right]$which is equal to the $i - row$ of the consensus influence matrix. Therefore, we can construct J characteristic functions after period 2 such that $\{\varphi(x_{i,1})\}_{i=1}^{J}$ is the sequence. Finally, notice that this sequence is constant and equal to the characteristic function $\varphi(x_c)$ since

$$x_{1,2} = a_{1,1}^{*} x_{1,c} + a_{1,2}^{*} x_{2,c} + \cdots + a_{1,J}^{*} x_{n,c} = x_{2,2} = a_{1,1}^{*} x_{1,c} + a_{1,2}^{*} x_{2,c} + \cdots + a_{1,J}^{*} x_{J,c} = x_{J,2} = x_c \quad (33)$$

Therefore, all the characteristic functions for all the agents are equal, so after period 1 global consensus is reached.

Definition 7. Let $J' = \{J, i\}$ be the set of neighbors of agent i plus the node of agent i. A stubborn type agent i of the influence contact network $\mathcal{I}$ has edges $l_{i,j} \in L: l_{i,j} = \begin{cases} 1 & i = j \\ 0 & otherwise \end{cases} \forall j \in J \; and \; i \in V$.

Definition 8. An influential type agent k for an agent i in an influence contact network $\mathcal{I}$ exists if the edges $v_{j,i} \in V: v_{j,i} = \begin{cases} \omega_{i,j} & for \; j = k \\ c_{i,j} & otherwise \end{cases} such \; that \; \sum_{i \neq j} c_{i,j} < \omega_{i,j} \; with \; c, \omega \in [0,1]$. This agent is globally influential if this definition holds for all $v_{j,i} \in V$.

From these constructions, we have the following conjectures:

Conjecture 1. A unique influential agent drive the asymptotic consensus distribution near his asymptotic distribution.

Conjecture 2. The existence of more than 1 influential agent has no effect on the final value of consensus.

Conjecture 3. An agent close to be a stubborn agent delays the time at which consensus is reached.

Notice that given the complex dynamic of the system, it is not possible to find an analytical answer for these conjectures.

Simulation Results and Discussion

We design a simulation setting to test numerically our theoretical conclusions, and also to extend our analysis and draw conclusions for the conjectures of the previous section.

Simulation Setting

We have a total of 240 configurations for the analysis.

We use three contact network topologies: complete graph, random network with connectivity 0.05 and free scale graph. For each network, 4 different numbers of agents are simulated: 100, 1,000, 5,000 and 10,000.

We use four initial opinion configuration settings: type (1) where each agent has its own opinion probability distribution which is normal. The mean parameters at time zero

are sample from $Unif\ (0{,}100)$ and the variance is assumed to be equal to 1 for all agents. In this case, the initial mean opinion values are located all along the interval [0,100]. Type (2) where opinions are still unique and normally distributed, but the mean opinion comes from one of the 2 opposite mean initial opinion groups. The mean parameter for agents in the first group is $Unif\ (10{,}30)$ and for the second group $Unif\ (70{,}90)$. The variance is assumed equal to 1. Type (3) has the same characteristics of type (1), except opinions are volatile so the variance is given by $Unif\ (1{,}5)$. Type (4) has the same characteristics of type (2), but with variance $Unif\ (1{,}5)$.

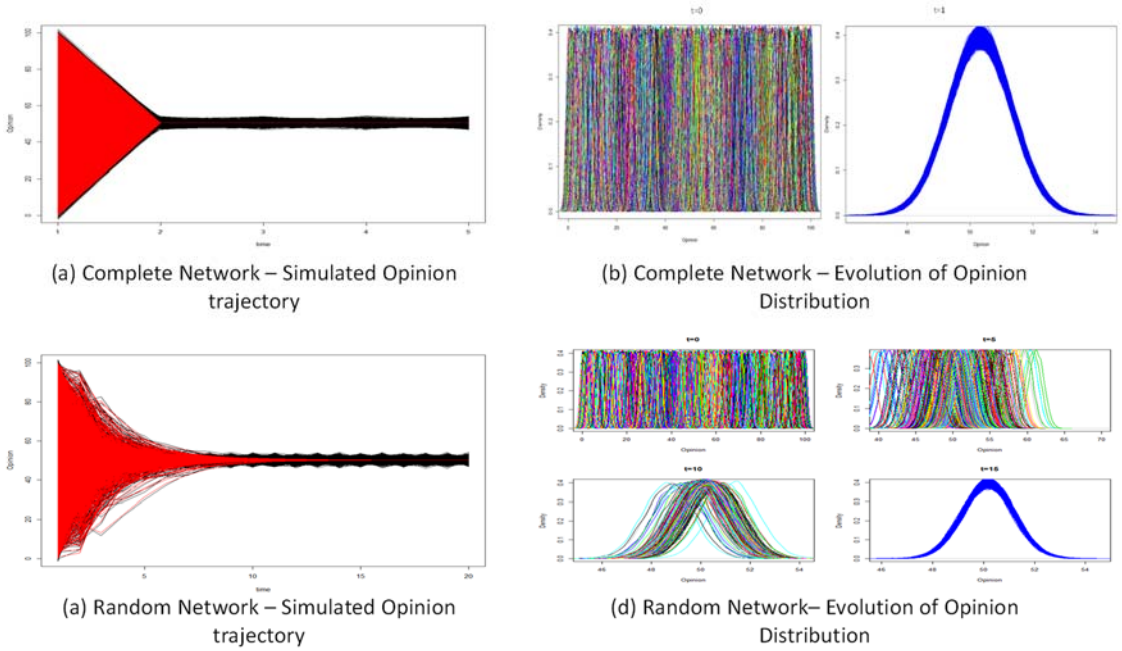
We use five different types of influence matrix configurations: type (a) where all the agents are adopters; type (b) where only 1 agent is globally influential and the remaining agents are adopters; type (c) where each agent has up to 20 influential neighbors; type (d), where the own opinion of the agent accounts for 90% of the influence (almost-stubborn agent), and the remaining weight is equally distributed randomly but only among 25% of his neighbors; and type (e) where the agents behave similar to type (d) but the remaining weight is equally distributed only among 25% of his neighbors with similar opinion.

Results

In our SODM, the existence of a spanning tree in the contact network does not necessarily guarantee global consensus. As noted, the spanning tree condition is required on the influence network. For 224 out of our 240 configurations, a spanning tree exists. For these configurations, the following set of claims are valid:

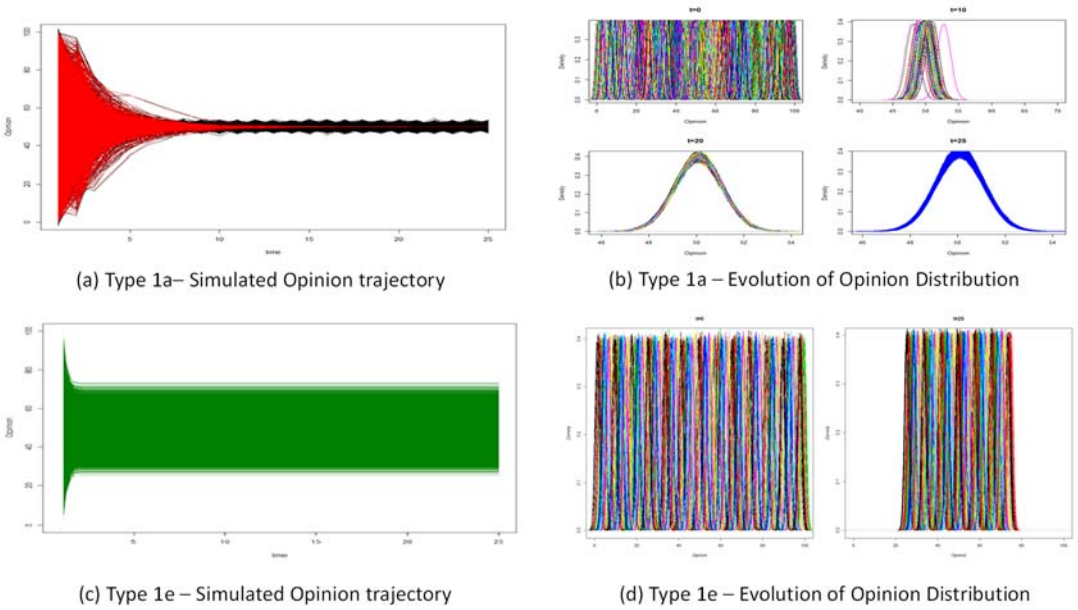
Remark 1. Initial conditions per se are not relevant for time of consensus. Connectivity and structure of the influence network is what matters. If the influence network is highly connected, global consensus is reached faster. (Fig. 18, 19a)

Figure 19 Evolution of Opinion for a network of 10, 000 agents and type (1a). Each color in (b) and (d) represents a different opinion distribution



Remark 2. The time for consensus does not change as the size of the network increases. In a complete network to reach global consensus takes 1 period (Fig. 19a-b); in a random network it takes less than 15 periods (Fig. 20c-d); and in a free scale network it takes between 20 to 25 period of interaction (Fig. 21a-b).

Figure 20 Evolution of Opinion for a Free Scale network of 10, 000 (Consensus and No-Consensus). Each color in (b) and (d) represents a different opinion distribution



(a) Type 1a– Simulated Opinion trajectory

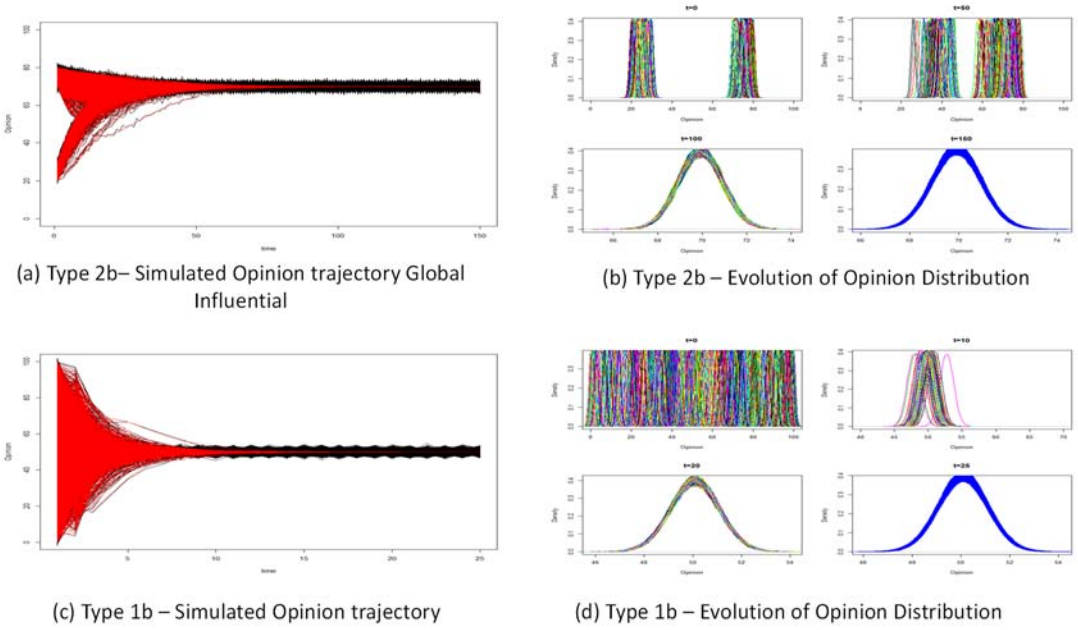
(b) Type 1a – Evolution of Opinion Distribution

(c) Type 1e – Simulated Opinion trajectory

(d) Type 1e – Evolution of Opinion Distribution

Remark 3. When no global influential agent is present, the mean value of consensus and the global consensus distribution are the same for the three network topology structures. In addition, the mean consensus value is always centered at the mean value of the initial opinions. Even though this is true, no common real value opinion can be claimed to hold among the agents (Fig. 19-20a). This finding holds as the number of agents increase.

Figure 21 Evolution of Opinion for a network of 10, 000 agents for a Free Scale Network ((a)&(b) Global Influential vs Local Influential (c)&(d)). Each color in (b) and (d) represents a different opinion distribution.



(a) Type 2b– Simulated Opinion trajectory Global Influential

(b) Type 2b – Evolution of Opinion Distribution

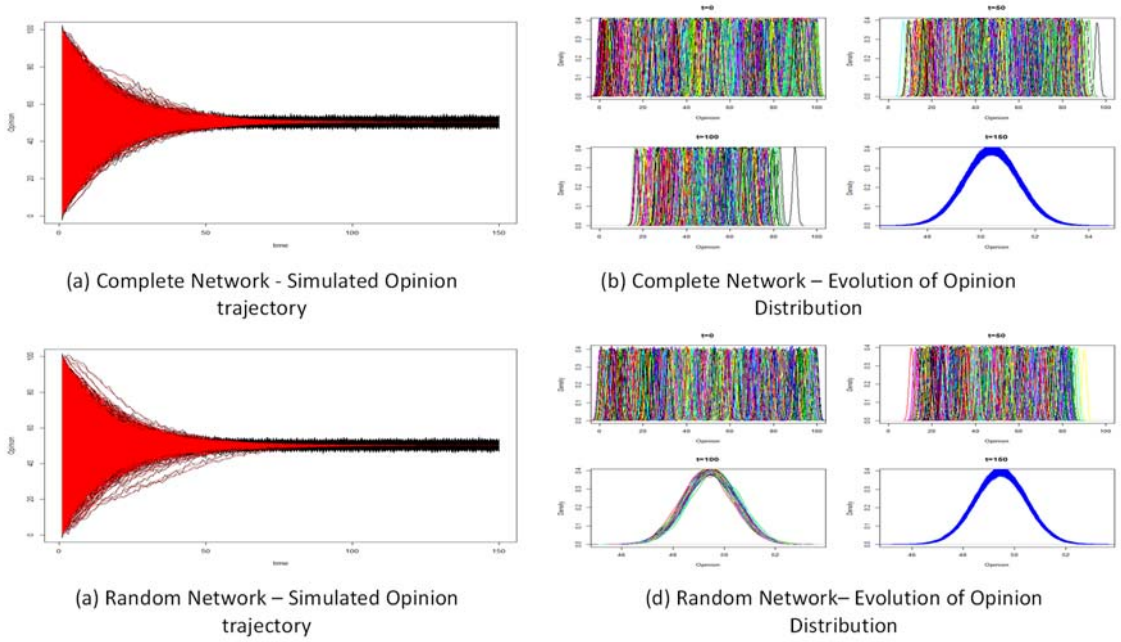
(c) Type 1b – Simulated Opinion trajectory

(d) Type 1b – Evolution of Opinion Distribution

Remark 4. Under the existence of a global influential agent, global consensus is reached but driven towards the opinion of the influential agent (Fig. 21a-b). Notice that an influential agent under our definition hardly changes his opinion distribution, but he manages to convince the rest of the agents towards his own views. Nonetheless, to reach global consensus takes more time. On the other hand, local influential agents have no impact on the consensus value nor time (Fig. 21c-d).

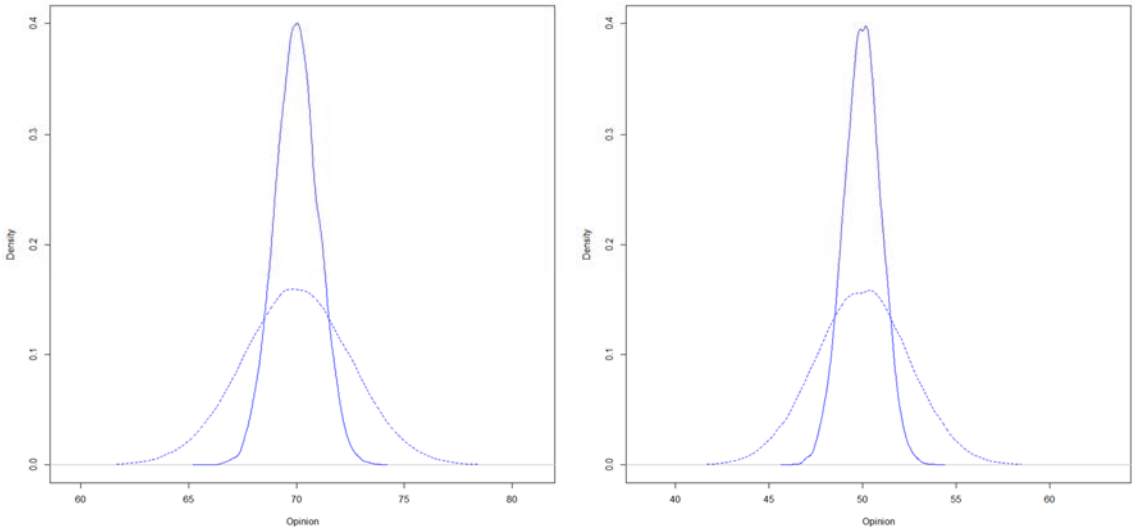
Remark 5. Global consensus can only be delayed if the agents always place a high weight on his own opinion. Under an influence matrix type (e) and (d), the time to reach consensus increases by a factor of 150 in the case of a complete contact network (Fig. 22a-b), by 10 in a random contact network (Fig. 22c-d), and by 6 in a scale free contact network (Fig. 21c-d).

Figure 22 Evolution of the opinion of a network of 10, 000 almost stubborn agents. Each color in (b) and (d) represents a different opinion distribution.



Remark 6. Given that an opinion system has reached global consensus, when the individual opinion distributions of the agents are volatile, the global consensus distribution inherits this structure. This fact implies that the variance accounts for the level of uncertainty of the opinion of the agents. For practical implications, a high variance implies uncertainty about a specific topic –potentially, an opinion about a new or controversial issue. On the other hand, a small variance reflects high level of certainty of an agent about his opinion (Fig.23).

Figure 23 Global Consensus Distribution.



(a) Global Influential agent (a) Other type of agents

(--- N(70,2.5), ____ N(50,1))

In the case of the other 16 configurations, we have the following findings:

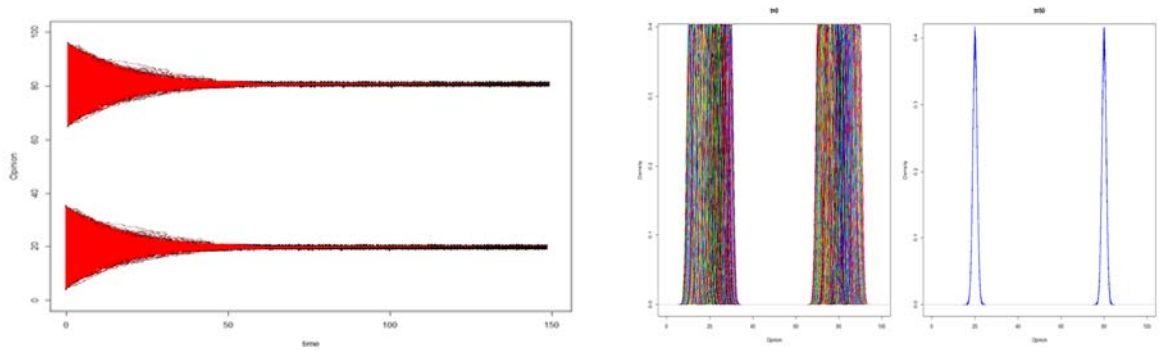
Remark 7. Global consensus is not reached if in a free scale contact network all agents are almost stubborn agents that from the beginning only communicate with peers that have a similar opinion. In this case, the influence network has no spanning tree (Fig.20c-d).

Remark 8. As opinions are updated, each agents reaches his own stable opinion distribution after interacting only 10 periods. This finding holds regardless of the network size. In addition, as the number of agents increases, the number of opinion groups increases (Fig.20c-d).

Remark 9. Under configuration type 1, all the opinions are bounded between the region (30,70). Depending on the variance of the opinion of each agent at time 10, a potential agreement region can be constructed. However, it is not guarantee that all the agents will share it (Fig. 20c-21d).

Remark 10. Under configuration type 2, two common regions of agreement arises. It takes, at least, 150 periods for the agents to reach local consensus. In this case, the initial existence of two opinion groups remained over time (Fig.24).

Figure 24 Evolution of Opinion for a Free Scale network of 10, 000 agents with two initial opinion groups. Each color in (b) and (d) represents a different opinion distribution.



(a) Type 2e- Simulated Opinion trajectory

(b) Type 2e – Evolution of Opinion Distribution

Conclusion

We have proposed a new opinions dynamics model based on the stochastic nature of opinions. We have shown that the SODM can explain the existence of a persistent group of agents with different opinions. Under SODM, consensus needs to be define as a global and local condition based on the opinion distribution of each agent. In these cases, the structure of the influence network determines the existence and the number of groups of opinions (consensus). Our main findings are:

• Agents arrive to consensus when their individual opinions are bounded, converge individually to a unique probability distribution, and the influence matrix admits a Jordan Normal form.

• When a spanning tree is present in the network, a global asymptotic consensus distribution exist; however, the probability that all agents reach a fixed point (a deterministic consensus value) is zero;

• When no global consensus exists, our model allows us to find the number of local asymptotic consensus distributions and the common regions among these groups. This facilitates the comparison, characterization and quantification of opinions between groups.

• Only a global influencer is capable of imposing his opinion

• Local influencers have no effect on the final distribution and time for consensus

• If an agent is almost stubborn and the contact network exhibits preferential attachment properties, only local consensus is possible.

Chapter 6: Influence Estimation and Opinion Tracking over Social Networks under Complete Information

Overview

Studying how a person's opinion on a specific topic is formed, how these opinions evolve, and how a consensus in a network that the person belongs to is reached is important in domains such as organizational behavior, politics, marketing, sociology, psychology, engineering, and economics. Researcher have been focusing on estimating hidden opinions, identifying opinion trends, opinion leaders, consensus points, and the impact of the structure of the network on consensus. In this paper we assume that opinions can be measured and focus on identifying who is influencing whose opinions. This is important for identifying the influential agents in the network and also for predicting when and where a consensus would be formed. On this chapter we operationalize the SODM from the previous chapter, to track in a first instance average opinion of agents over a social network.

This chapter is organized into five sections. Section 2 presents the SODM into a functional state space model and proposes a restricted maximum likelihood (Enders 2001, Schoenberg 1997, Skrondal and Rabe-Hesketh 2004, Wooldridge 2010) based approach to estimate who influences whose opinion and to what degree. In this setting, the mean opinion of each agent and the contact network of the agents are assumed to be observable while the probabilistic opinion functions of the agents and the structure of the influence network are unobserved. The results are presented in Section 3. We use different SODM scenarios to simulate the opinion process and establish the asymptotic properties of the estimator and the robustness of the estimates. Section 4 outlines the conclusions, limitations and topics for future research.

Estimation of the SODM

Model

When modeling SODM for the whole network, Eq. (2) can be written as a system of functional stochastic difference equations of first order given by:

$$X_t = AX_{t-1} + BY_t \tag{1}$$

where X_t is a functional with $i = 1, \dots J$ opinion distributions of the J agents, A collects all $a_{i,j}$ coefficients of the influence that agent j exerts in agent i, and B gathers all the $b_k y_{k,t}$ exogenous functions of external information.

Note that each agent at time $\boldsymbol{t}$ has its own probability distribution, and this function is evolving over time. The functional form of each agent's probability distribution is unknown and potentially different across agents. In this sense, it is not clear how a likelihood function for the opinion process over the whole network can be written in a classical estimation context. Since the SODM is in a functional setting, classical regression methods[10] cannot be applied directly. In this context, we develop an estimation procedure based on the ideas of tracking the mean opinion of the agents individually.

For agent, i we apply the expectation operator $E[\,.\,]$ with respect to Eq. (1) and obtain Eq. (2a) and (2b).

[10] There are two more alternative options to this problem: (i) Use the functional regression framework of (Ferraty and Vieu 2006) We do not opt to do so since the results of this method are subject to the election of the semi-metric that is chosen to estimate the functional model. There is still no common agreement between researchers which functional norm should be used as the optimal one. (ii) We can try to fit individual kernels using the k measurements for every agent. Once the kernels are fitted, we can use nonparametric regression to recover mean value estimates of A. We do not choose to do so since the kernel estimation would consume most of the degrees of freedom of our observations, and the computational time would be exponential given the large size of the parameters space – (Racine, Parmeter, and Du 2009)

$$E\left[x_{i,t}\right] = E\left[\sum_{j=1}^{M} a_{i,j} x_{i,t-1} + \sum_{k=0}^{K} b_k y_{k,t}\right] \quad (2a)$$

$$\bar{x}_{i,t} = \sum_{j=1}^{M} a_{i,j} \bar{x}_{i,t-1} + \sum_{k=0}^{K} b_k \bar{y}_{k,t} \quad (2b)$$

Further, from (Rosenblatt 1956), it is known that the distribution of the mean can be approximated by a normal distribution. In this context, it is natural to approximate the mean opinion distribution of each agent $\bar{x}_{i,t}$ by $N_{i,t}(\mu_{i,t}, {\sigma^2}_{i,t})$ where $\mu_{i,t}$ is the true mean opinion at time t for agent i and ${\sigma^2}_{i,t}$ is the variance of the process. In the same line, the external source of information is $\bar{y}_{k,t} \sim N(\gamma_{i,t}, {\varrho^2}_{i,t})$.Therefore, Eq. (2a) and Eq. (2b) model the functional stochastic mean opinion process.

In a data context, opinions are not directly observed (Ren, Beard, and Kingston 2005). This same proposition applies to our model, we do not observe the mean opinions distribution of each agent. For every agent i at time t only n measurements $o_{i,t,n}$are observed. In addition, all the coefficients of influence $a_{i,j}$ are not observed; only the adjacency matrix C of the contact network $\mathbb{S}$ is observed. Finally, it is known r values $w_{k,t-1,r}$ from the exogenous function inputs $\bar{y}$, but not their coefficients b_k.

Let $\overline{o}_{i,t} = \frac{\sum_k o_{i,t,n}}{n}$ be the sample average of the k measurements of agent i at time t. In the case of the exogenous functions, let $\overline{\gamma}_{k,t} = \frac{\sum_r y_{r,t,k}}{r}$ be the average of the r observations for function. Under this setting, Eq. (2b) can be written as a latent variable model for the mean parameter value (Anderson 1989, Muthén 2002, Skrondal and Rabe-Hesketh 2004, Wooldridge 2005) with additive noise; $\varepsilon_{i,t} \sim N(0, \xi_i)$ and $\epsilon_{i,t} \sim N(0, \varphi_i)$.:

$$\mu_{i,t} = \sum_{j=1}^{M} a_{i,j}\mu_{i,t-1} + \sum_{k=0}^{K} b_k \overline{\gamma}_{k,t} + \varepsilon_{i,t} \quad (3a)$$

$$\overline{o}_{i,t} = \mu_{i,t} + \epsilon_{i,t} \quad (3b)$$

Using the relationship between $\overline{o}_{i,t}$ and $\mu_{i,t}$ at t and $t-1$, Eq.(3a) and Eq. (3b) can be rewritten as:

$$\overline{o}_{i,t} = \sum_{j=1}^{M} a_{i,j}\mu_{i,t-1} + \sum_{k=0}^{K} b_k \overline{\gamma}_{k,t} + w_{i,t} \quad (4)$$

where $w_{i,t} = \varepsilon_{i,t} + \epsilon_{i,t}$ with $N(0, \varpi_i = \varphi_i + \xi_i)$. The formulation of Eq. (4) is a structural equations model of the linear structural type (LISREL) (Skrondal and Rabe-Hesketh 2004). The log-likelihood function for agent i is given by:

$$\ell_i \cong -{}^1\!/_2 \sum_t \left[\left(\overline{o}_{i,t} - \mu_{i,t}\right)' {\Omega_i}^{-1} \left(\overline{o}_{i,t} - \mu_{i,t}\right) + \ln|\Omega_i| + \ln 2\pi \right] \quad (5)$$

where $\Omega_i = A_{i_{1\times J}} \Psi_{J\times J} A_{i_{J\times 1}} + \varpi$ is the product of the matrix with all the influence coefficients $a_{i,j}$ for agent i and the covariance matrix of the latent variables $\bar{x}_{i,t-1}$ plus the variance of the error $w_{i,t}$. As shown by (Jöreskog 1967), a sufficient statistic for Ω_i in the complete data case is given by the empirical covariance matrix S_i calculated from the measurements. In the case of $\mu_{i,t}$, it is approximated by the sample value $\mu_{i,t} = \sum_{j=1}^{M} a_{i,j}\bar{o}_{i,t-1} + \sum_{k=0}^{K} b_k \bar{y}_{k,t}$ which by construction assumes that $E\left[\overline{o}_{i,t-1}\right] = E\left[\mu_{i,t-1} + \epsilon_{i,t}\right] = E[\overline{x}_{i,t-1}]$. Depending on the structure of the social network $\mathbb{S}$, many elements of Ω_i can be set to zero given the construction (Eq. (1)) of the elements of A_i.

Notice that the previous construction can be done for all the J agents, then J individual log-likelihood functions can be maximized. Nonetheless, the SODM imposes a restrictions on the values of $a_{i,j}$, then Eq. (3a) and Eq.(3b) are estimated by restricted maximum likelihood (Enders 2001, Schoenberg 1997, Skrondal and Rabe-Hesketh 2004, Wooldridge 2010). The identification and efficiency of the parameters has been discussed using M-Theory (Skrondal and Rabe-Hesketh 2004, Wooldridge 2010). In this sense, the restricted likelihood function of Eq. (6) is Quasi-Concave, and a unique maximize exists for the problem. Therefore, the parameters can be uniquely identified. In addition, when the true process follows the proposed functional form the parameters are efficient. As more information is available, the parameters reach their true value (Muthén 2002).

Notice that no distributional assumptions on the individual opinion probability functions of the SODM have been made. In addition, given that only the neighbors of an agent i may influence an agent, we can estimate J individual problems.

Estimation and tracking algorithm

In many applications, we are required to track opinions over mid-size (networks with less than 5 million nodes) or large size social networks. We tackle this issue, by proposition a two-step algorithm. The first step is the estimation of the influence parameters. At this stage, we use the latent variable model of Section 3.1 for each agent since only local information is required in the estimation. This formulation allow us to recover the parameters of a group of agents $N < J$ in parallel. The second step is opinion tracking over

time; for this we use the SODM structure to perform a Monte Carlo simulation given the influence parameters and variance of the error.

Our algorithm is given by:

Table 22 SODM estimation and mean opinions' tracking algorithm

Estimation Step
1. Determine C, the number of cores available for computing.
2. Split the J agents of the contact network in batches of size $s_C = \frac{J}{C}$ among the C cores
3. For each group s_C collect only the $e_{i,j}$ information of their immediate neighbours, and their respective measurements.
4. On each core, for each i node in the set s_C solve the optimization problem: $\max_{a_{i,j},b_k} \ell_i$ $a_{i,j} = 0 \; if \; e_{i,j} \notin \text{Neighbors(i)}$ $0 \leq a_{i,j}, b_k \leq 1$ where ℓ_i is given by Eq. (7).

5. Store C lists on each one of the cores with the estimated parameters and the error variance $\widehat{\Pi}$.

Tracking Step

1. Collect all the parameters from the C lists in the first core and arrange it in a matrix form $\hat{A}$ and $\hat{B}$.

2. Set $\widehat{\mathrm{M}}_0 = \begin{bmatrix} \overline{o}_{0,t} \\ \cdots \\ \overline{o}_{J,t} \\ \overline{s}_{1,t} \\ \cdots \\ \overline{s}_{J,t} \end{bmatrix}$, $\widehat{\mathrm{A}} = \begin{bmatrix} \hat{A} & 0 \\ 0 & \hat{\mathcal{A}} \end{bmatrix}$, $\Gamma_t = \begin{bmatrix} \overline{\Lambda}_t \\ \overline{\Lambda}_t \end{bmatrix}$, and $\widehat{\mathrm{B}} = \begin{bmatrix} \hat{B} & 0 \\ 0 & \hat{\mathcal{B}} \end{bmatrix}$

3. Simulate τ periods of the mean opinion process by using the recursion

$$\widehat{\mathrm{M}}_t = \widehat{\mathrm{A}}\widehat{\mathrm{M}}_{t-1} + \widehat{\mathrm{B}}\overline{\Lambda}_t + v_t$$

Where v_t is a vector of simulated errors sampled from $N(\vec{0}, \widehat{\Pi})$

We use the statistical software R to construct the estimation algorithm. Since the log-likelihood can be solved by maximizing only the mean square part of Eq. (7), we solve the optimization as a quadratic programming problem using the approach of (Goldfarb and

Idnani 1982, 1983). In this fashion, we are able to simulate different trajectories for the mean opinions, and the evolution of the opinion distributions of the agents.

Results

Simulation Setting

A computational study is conducted to test the estimation methodology and asymptotic properties of our model on midsize networks. First, we simulate the contact network structure and the opinion process using the SODM outlined in Section 2. For this purpose, a total of 18 scenarios as described below are simulated.

- Three types of contact network topologies: we use complete (C), random (RN) and free scale (BA) network structures with an average connectivity of 100 for a fixed size of 10,000 agents.
- Two different initial opinion conditions are used.
    - In Type (a) the mean initial opinion of all the agents range between 0 and 1 (emulating a potential opinion index). The mean is drawn from $Unif\sim(0,1)$, and the variance from $Unif\sim(0.1,0.25)$.
    - In Type (b) the initial opinion distributions come from two groups, where group 1 is modelled using a $N(5,0.05)$ and group 2 using a $N(0.75,0.05)$. This election is made so the two groups start apart from each other representing a central and extreme opinion with no common region at $t = 0$.
- Three types of influence matrices are used:

- Type (1) where all the weights for agent i are similar between his neighbors, but the weights are not close to the boundary values
- Type (2) where all the weights for agent i are different between his neighbors, and the weights are not close to the boundary values.
- Type (3) where all the weights for agent i are different between his neighbors, and his own weight $a_{i,i}$ is closed to the boundary values.

Note that these influence matrices capture all the potential variety and influence among agents while allowing us to test numerically cumbersome parameter values, especially with values close to the boundaries when using M-estimation methods (Hamilton 1994).

We simulate a total of 1000 periods of interaction; drawing 10 measurements per each agent at each time period. For each scenarios we simulate 100 replications. Since for each network, we have approximately a total of 1 million parameters to estimate; we evaluate the performance of our method by analyzing the bias distribution of all the parameters. In this sense, a distribution centered in zero and with a low variance is the preferred distribution.

Identification

Based on our algorithm, mainly Eq. (5), we can see that the main identification condition is given by the matrix $X'X$, which needs to be a full ranked matrix for the model to be identified. This condition is true in the cases of the random and scale-free network since more periods of interaction $(t = 1000)$ than parameters (on average $101\, a_{ij}$ coeffiecients). We fail to comply with the rank condition for the complete network case, therefore we cannot identify all the parameters for the 10,000 nodes network. In this case,

only a subgraph of size less than 1,000 of the complete network can be identified. For this reason, we label the complete graph case as unidentifiable under our method, and proceed with the analysis for this and the remaining sections only for the other two network topologies.

The impact of influence network type and contact network is presented in Fig.24 and Fig.25. All of the 95% confidence intervals for the all the parameters include the true parameter value, indicating that the parameters are identifiable. In the case of the type (3) influence network, thought the parameters are still identified, the confidence interval is wider than when the influence parameters are equal or vary slightly. The width of the confidence interval for the parameters of the random network is slightly greater than the interval for the scale-free network.

Figure 25 Bias distribution of the estimated parameters for a Random Network of 10,000 agents under type (a)

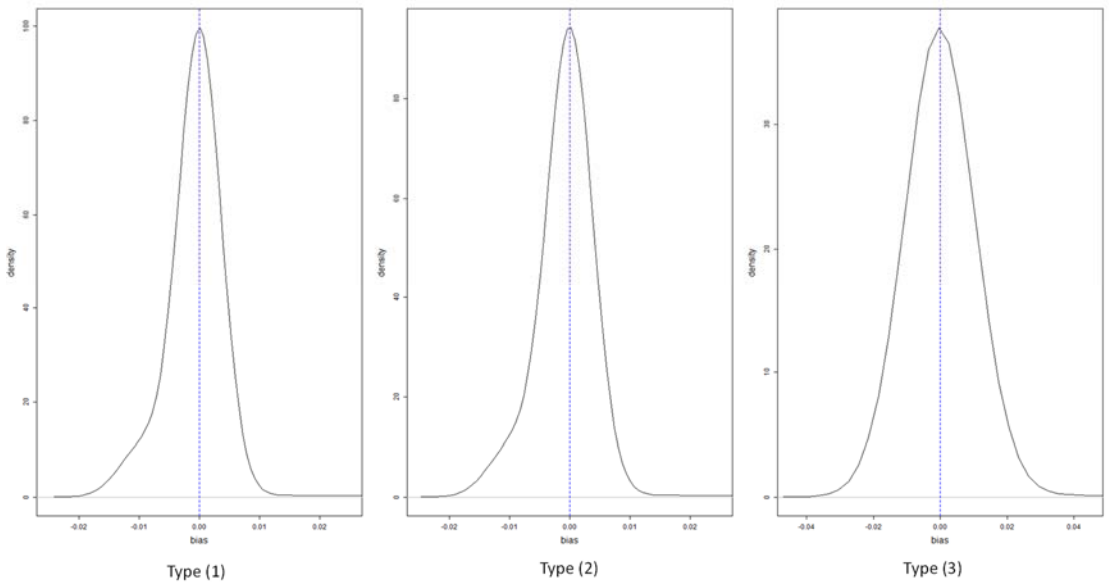
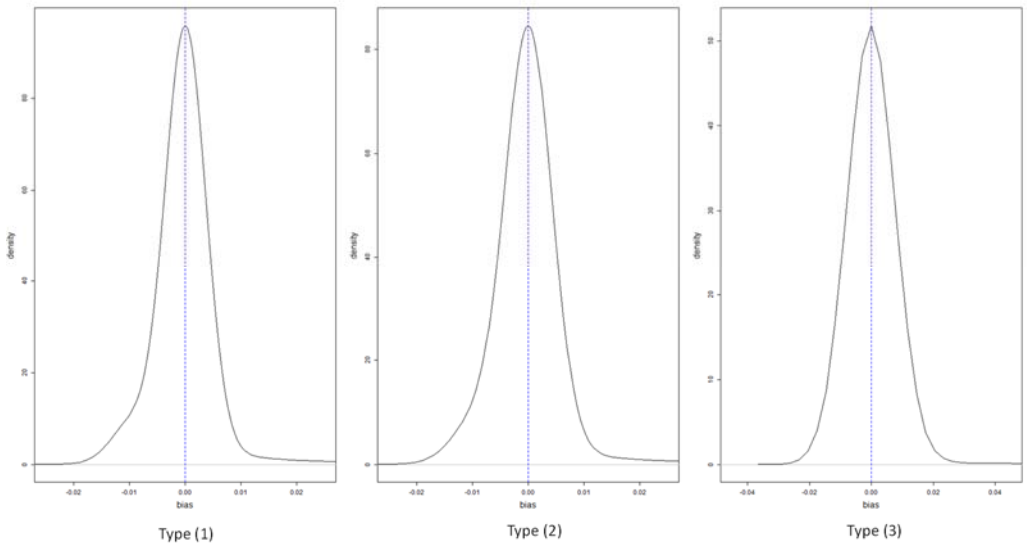


Figure 26 Bias distribution of the estimated parameters for a Scale-free of 10,000 agents under type (a)



1.1. Asymptotic properties

We extend our simulation study to test the asymptotic properties of our estimation algorithm. We generate data for 250, 500 and 1,000 periods of interaction. The asymptotic behavior of our estimation algorithm can be seen in Fig. 26 and Fig. 27 .As more information is available, the results show that the estimation recovers the true parameters and its precision improves asymptotically. For both contact networks and the three types of influence networks, the model provides consistent estimates of the true parameters. The 95% confidence intervals in the case of the influence network type 3 is the widest when only 250 periods of information is available.

Figure 27 Asymptotic behavior of the estimated parameters for a Random Network of 10,000 agents at the 95% confidence level (--) under type (a)

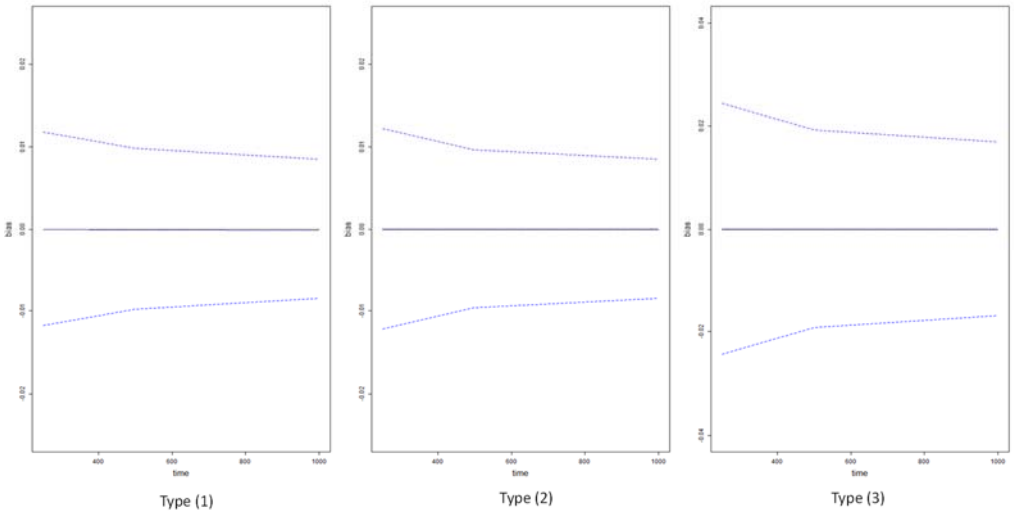
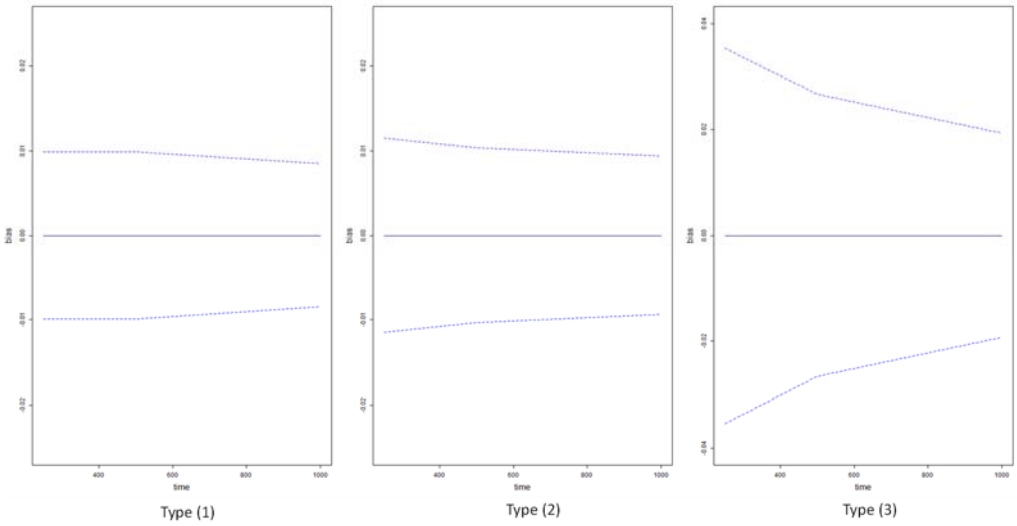


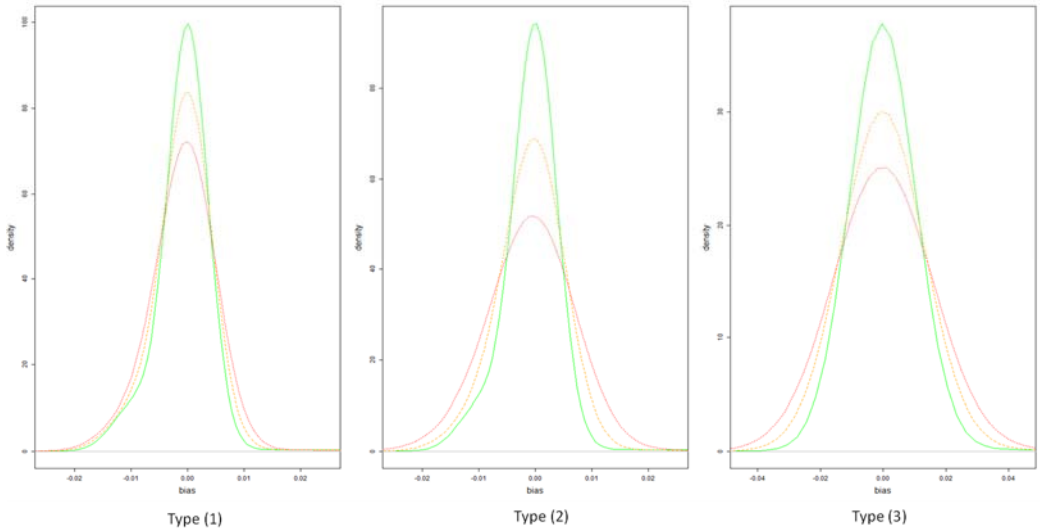
Figure 28 Asymptotic behavior of the estimated parameters for a Scale-free Network of 10,000 agents at the 95% confidence level (--) under type (a)



The variance bias distribution and its evolution as the information increases in time is depicted in Fig. 28 and Fig. 29. These results show that the precision of the parameters improves as more information is available. In addition, the parameters estimated from a

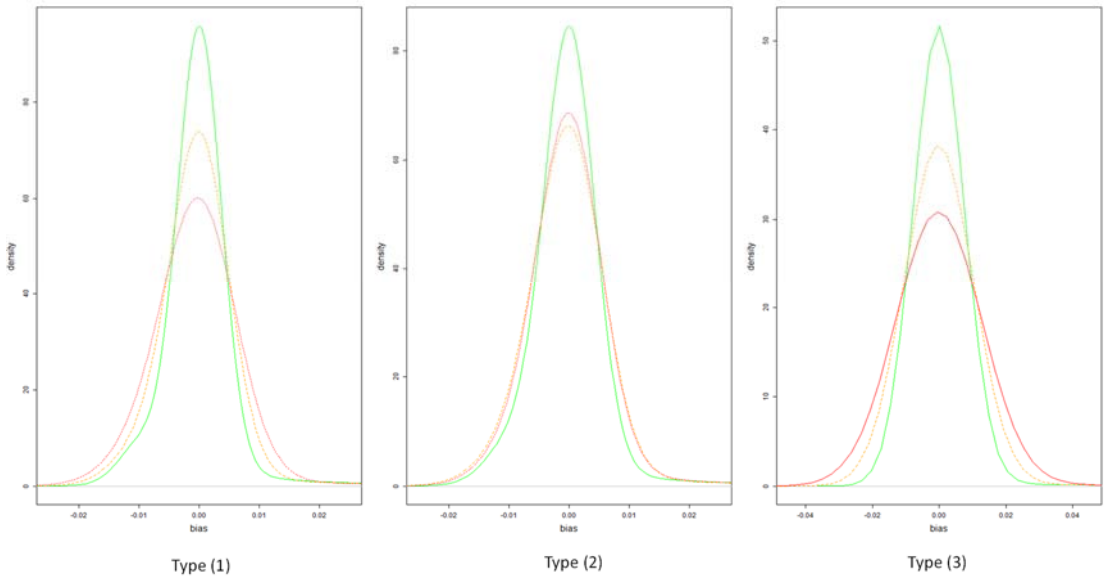
scale-free network are slightly more precise than the parameters estimated from the random network.

Figure 29 Asymptotic behavior of the bias distribution of the estimated parameters for a Random Network of 10,000 agents under type (a)



250 -- , 500 -- and 1,000 -- interaction periods

Figure 30 Asymptotic behavior of the bias distribution of the estimated parameters for a Scale-free Network of 10,000 agents under type (a)



250 -- , 500 -- and 1,000 -- interaction periods

Robustness

We extend our simulation analysis to study the impact of the amount of information per agent per period of time. In this case, we collect three types of opinion measurements for each t. For type l, only 5 measurements are observed per each period t. For type m we observe 10 measurements. Finally, for type h we collect 50 measurements per period.The results in Fig. 7 and Fig. 8 show that the estimation algorithm is robust when medium and large numbers of measurements are observed per period. The 95% confidence intervals in all cases include the true parameter; in addition, the precision increases as the amount of available information per agent increases. Once again, when the influence matrix is of type 3, the confidence intervals are wider.

When the amount of information is low, the confidence intervals of the bias suggest that the parameters cannot be fully identified. Notice that on average, the true value of the influence weight stays around 0.01; this means that when the amount of information per period per agent is low, the precision of this parameter is lower than in the rest of the cases. Then, the zero value is included inside the confidence interval for the parameters. This result can be expected when the agents of the network hardly emit visible measurements.

Figure 31 Asymptotic behavior of the estimated parameters for a Random Network of 10,000 agents at the 95% confidence level (--) under type (a) as the amount of measurement per period of time increases

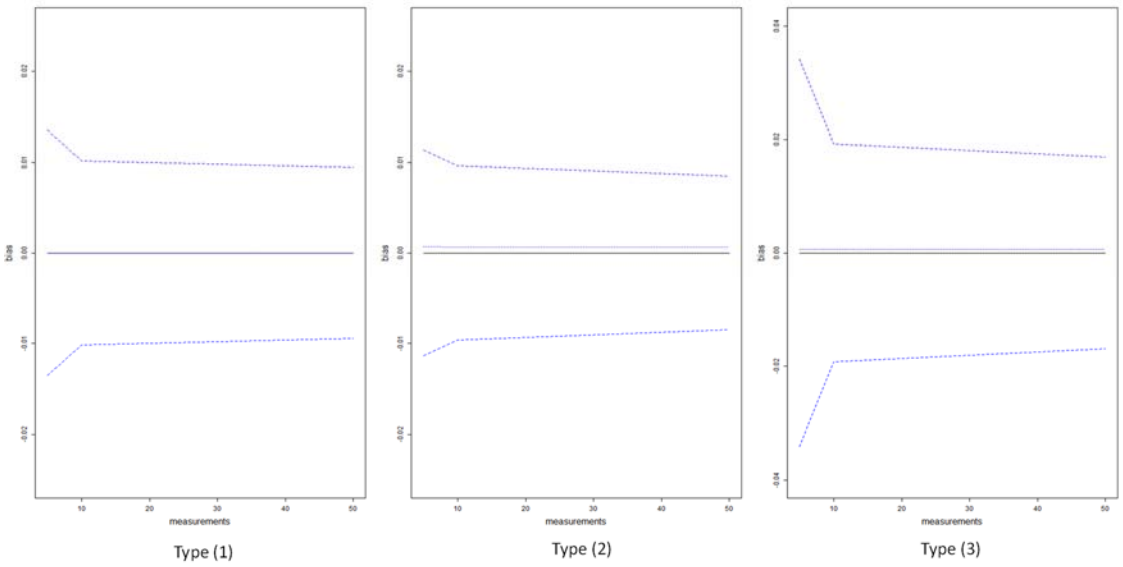
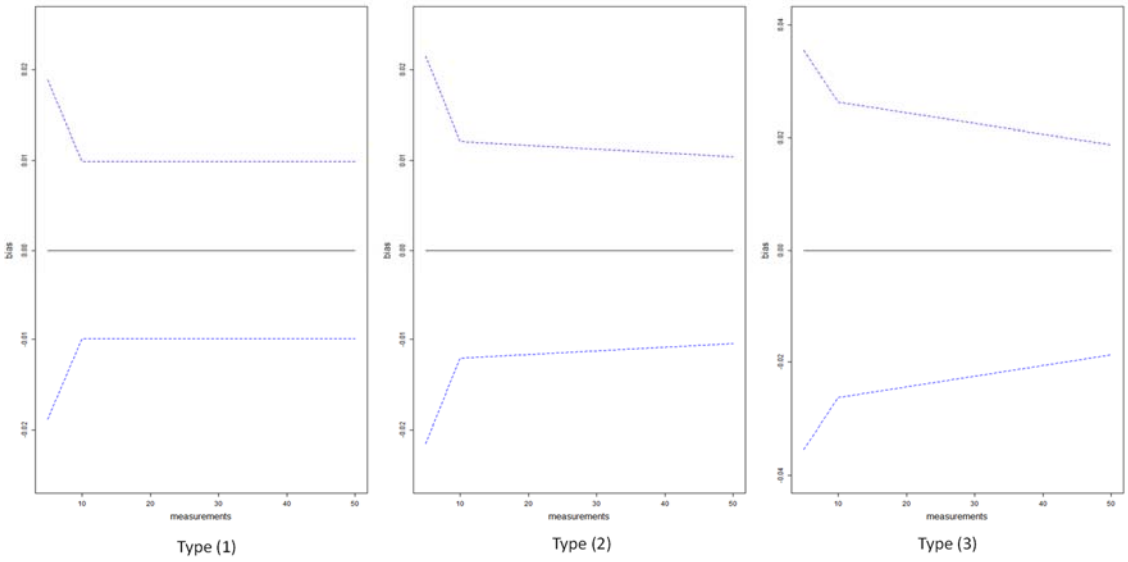


Figure 32 Asymptotic behavior of the estimated parameters for a Scale-free Network of 10,000 agents at the 95% confidence level (--) under type (a) as the amount of measurement per period of time increases



Conclusion

We propose a latent variable model for the estimation of the influence matrix and tracking of the mean opinion distributions for each agent of a social network. Our theoretical opinion model is grounded on the idea that the opinions of each agent can be modelled as individual probability distributions. Our theoretical model is functional; then, we develop an expected value transformation to estimate the parameters of the model. Given that opinions cannot be directly observed, only noise measurements are observed; we build a latent variable method to estimate the unknown influence network. Our algorithm offers the following computational advantages: (a) the estimation of the influence parameters for each agent only requires local information for deriving efficient estimates; (b) it only requires to model the mean parameter of the mean opinion distribution to recover the influence weights; (c) the evolution of the mean opinion distribution can be recovered and track by Monte Carlo simulation once the parameters has been estimated.

The model is estimated by restricted maximum likelihood. As noted by M-theory on restricted optimization (Enders 2001, Schoenberg 1997, Skrondal and Rabe-Hesketh 2004, Wooldridge 2010), the optimization can be solved as a quadratic programming problem where the function is convex and has a unique optimum (Goldfarb and Idnani 1982, 1983). The model was tested for identification, asymptotic, stability, and robustness. It was found that a full rank condition is required on the matrix $X'X$; this means that the estimation algorithm can only recover parameters when the number of time periods observed is greater than the number of influencers plus the external sources of information. This poses a limitation for the estimation of the influence matrix when the contact network is a complete network; in this case the parameters cannot be identified. For influence matrixes where the

contact network is random or free-scale, the estimation algorithm identifies completely the parameters. Simulation results also indicate that the estimation methodology yields unbiased, consistent and precise estimates of the true parameters if the underlying data generation process conforms the SODM framework.

Under both low and high time period information scenarios, the estimated parameters are consistent. When the influence weights are not too different, the asymptotic show a better performance. When we test our algorithm for robustness, the results suggest that medium and high levels of per time period information are required. The numerical experiment shows that $10\times2(neighbors(i)\times\#external\ information\ sources)$ data points are the minimum size requirement for each agent to estimate our model and have good small sample asymptotic results. In the case of a network with 10,000 agents, this condition translates into approximately 2020 observations. Notice that this condition is changes with the size of the social network, the number of neighbors per agent and the opinion process that is being modelled; therefore, it is case specific. These results suggest that the convergence ratio depends not only on large t, but also on the number n and reliability of the opinions at each t. It is preferable to have large n (more opinions during a time frame) than t. From a computational perspective, the estimation algorithm is scalable and compatible with requirements for distributed computing. The estimation procedure for N agents can be estimated as N loosely synchronized problems and this enables the efficient use of all cores and processors available for computation.

Chapter 7: A Particle Filter Based Approach to Estimate Influence and Track Opinions over Social Networks

Overview

The ability to track how opinions about people, place, issues or things are evolving over time is important in a multitude of domains such as engineering, social sciences, and management science. This ability enables firms to identify emerging trends and dominant opinions, opinion leaders, influencers and stubborn agents, and take proactive/predictive actions. This opinions are formed and propagated over different network structures. In this context, it is necessary to identify not only the opinion trends, but how these opinions are formed, who are the influential members of the network and how opinions evolved over time.

Unfortunately, opinions and influence are not directly observed. Online learning particle filter algorithms offer a flexible framework to recover both from available network and opinion data. This method allows the estimation of a posterior conditional distributions of any variable and its parameters, given a set of explanatory variables and measurements. It is built to tackle the curse of dimensionality due to its sampling strategy (Carvalho, Johannes, et al. 2010, Vaswani 2008b), a problem that arises in social networks where the number of parameters is large and the available dataset is sparse. In this paper, we present a procedure that enables us (a) identifying who is influencing whose opinions and to what degree when the complete and partial structure of the contact network is observed and (b) tracking the opinions of every member of a network.

From the previous chapter we have learned that an estimation that fully relies on restricted maximum likelihood demands great amount of information. Online learning particle filter algorithms offer a flexible framework to recover posterior conditional

distributions of any variable and its parameters, given a set of explanatory variables and measurements; and it is built to tackle the curse of dimensionality due to its sampling strategy (Carvalho et al. 2010, Vaswani 2008). In this chapter, we present a procedure that enables us (a) identifying who is influencing whose opinions and to what degree when the complete and partial structure of the contact network is observed and (b) tracking the opinions of every member of a network.

The chapter is organized into four sections. In section 2 we operationalize the SODM into an Online Particle Filter Model. In this setting, the probabilistic opinion functions of the agents are unobserved and the influence structure is unknown. We have two cases: (i) when it is only observed noise measurements from the opinion space and the contact network of the agents is partially known; and (ii) when it is only observed partially noise measurements from the opinion space and the contact network of the agents is fully known. The contact network captures the information about whether individual i is connected to individual j whereas the influence network captures information about the strength of the influence that individual i exerts on j. Based on these features, we develop an estimation procedure based on the concept of online particle filter to simultaneously estimate the influence parameters and track the opinions of the agents individually. We propose a model that can be estimated only considering local information for each agent. In Section 3; we use different SODM scenarios to simulate the opinion process and then test our algorithm on the synthetic data. Section 4 outlines the conclusions, limitations and topics for future research.

Estimation of the SODM

Online Particle Filter Model

Each agent at time t has its own probability distribution, and this function is evolving over time. The functional form of each agent's probability distribution is unknown and potentially different across agents. In this sense, a classical regression estimation cannot be directly performed. We develop an estimation procedure based on the ideas of online learning and tracking of parameters and signals from online particle filters. This approach allow us to track the opinion of the agents individually while estimating the influence weights of the opinion process.

Online particle filters offer a flexible way to estimate a large number of parameters in a model Lopes et al. 2010. The main idea is that having available T periods of information at hand, and starting at some time $t = t_0 < T$ the model learns the unknown parameters and latent mean opinion period after period. Online learning particle filter algorithms offer a flexible framework to recover posterior conditional distributions of any variable and its parameters, given a set of explanatory variables and measurements. Under a linear or non-linear model structure and a set of prior distributions for the parameters and the variables, the estimation procedure relies on Bayes rule to obtain an updated conditional distribution of the outcome under analysis. In this sense, online learning is built to tackle the curse of dimensionality given its sampling strategy, and the problem of partial and sparse information due to its Bayesian estimation framework (Carvalho, Johannes, et al. 2010, Carvalho, Lopes, et al. 2010, Vaswani 2008a).

Particle filter generalizes the idea of a measurement and state equation. In the general case, let's define $\widetilde{X_t}$ as the vector of unobserved parameters of the unobserved opinion distributions of every agent and unobserved opinion realizations at time t, $\widetilde{O_t}$ is the observed measurements available to the researcher, $\widetilde{Y_t}$ is the control input distributions representing external information, and let θ be the vector collecting all the influence parameters $a_{i,j}$.The general setting of SODM in online learning is given by:

$$\widetilde{X_t} \sim g\left(\widetilde{X_t}, \theta \middle| \check{X}_{t-1}, \check{Y}_t\right) \tag{1a}$$

$$\widetilde{O_t} \sim g(\widetilde{O_t} | \check{X}_t, \check{Y}_t) \tag{1b}$$

The sequential derivation of the filtering distribution is given by an adaptation of Bayes rule:

$$p(\widetilde{X_t}, \check{X}_{t-1}, \theta | \widetilde{O_{1:t}}, \check{Y}_{1:t}) \propto \tag{2}$$
$$p(\widetilde{X_t}, \theta | \check{X}_{t-1}, \check{Y}_t) p(\widetilde{O_t} | \check{X}_{t-1}, \check{Y}_t, \theta) p(\check{X}_{t-1}, \theta | \check{O}_{1:t-1}, \check{Y}_{1:t-1})$$

The estimation of Eq. (5) is done following (Carvalho, Lopes, et al. 2010). In this case, the estimation strategy only require us to resample and propagate the conditional likelihood. This is the additional construction $S_t(\check{X}_t, \theta)$ which is the state sufficient statistic. The advantage of his procedure is its wider applicability to dynamic models with lag dependencies, and it offers greater improvements in performance (amount of data and convergence rate) when compared to usual importance sampling methods. In addition, the

likelihood of Eq. (5) has the theoretical value of reflecting that the opinion of the any given agent only depends on the availability and amount of local information.

Estimation and opinion tracking under complete information

In many real life setting, for each agent of the network, her opinions and the opinion of her neighbors and all the information about her contact network is available. This information is taken as inputs. In this context, we can rewrite Eq. (1) under the general setting of online particle filter. We have:

$$\begin{bmatrix} g_1(x_{1,t,1}, \dots, x_{1,t,n}) \\ \dots \\ g_J(x_{J,t,1}, \dots, x_{J,t,n}) \end{bmatrix}_{J\times 1} = L_{J\times J} \begin{bmatrix} g_1(x_{1,t-1,1}, \dots, x_{1,t-1,n}) \\ \dots \\ g_J(x_{J,t-1,1}, \dots, x_{J,t-1,n}) \end{bmatrix}_{J\times 1} + \tag{3a}$$

$$B_{J\times K} \begin{bmatrix} y_1(w_{1,t,1}, \dots, w_{1,t,r}) \\ \dots \\ y_K(w_{K,t,1}, \dots, w_{K,t,r}) \end{bmatrix}_{K\times 1} + \varphi_{t\,J\times 1}$$

$$\begin{bmatrix} g_1(o_{1,t,1}, \dots, o_{1,t,n}) \\ \dots \\ g_J(o_{J,t,1}, \dots, o_{J,t,n}) \end{bmatrix}_{J\times 1} = H_{J\times J} \begin{bmatrix} g_1(x_{1,t,1}, \dots, x_{1,t,n}) \\ \dots \\ g_J(x_{J,t,1}, \dots, x_{J,t,n}) \end{bmatrix}_{J\times 1} + \vartheta_{t\,J\times 1} \tag{3b}$$

Opinions are still not observed, the function $g(.)$ is the unobserved pdf of opinions for each agent, and we collect J measurements $o_{j,t,n}$ at time t. We do not observe the value of the influence weight $a_{i,j}$ that agent i exerts on agent j; but we observe the structure of the contact network C such that its elements $c_{i,j} = \begin{cases} 1 & if\ i\ is\ neighbor\ of\ j \\ 0 & otherwise \end{cases}$. Given this structure, we can impose the constraint on the elements of L matrix such that $l_{i,j} = \begin{cases} 0 & if\ e_{i,j} = 0 \\ a_{i,j} & otherwise \end{cases}$ $for\ 0 \le a_{i,j} \le 1$. Finally, we observe r values $w_{k,t-1,r}$ from the exogenous function inputs $y_{k,t-1}$, but not their associated coefficients b_k.

Online learning can be applied to track opinions over mid-size (networks with less than 5 million nodes) or large size social networks. We tackle this issue, by proposing a modified two-step algorithm based on online learning. The first step is the estimation of the influence parameters and unknown states. We follow (Liu and West 2001) and (Carvalho, Lopes, et al. 2010) with our own node-by-node modification. In this stage we (i) resample the parameters; (ii) propagate the states and parameters, and (iii) resample states and parameters with importance weights to correct for potential lack of observations for parameter inference. The second step is opinion tracking over time; for this we use the SODM structure recovered to perform a Monte Carlo simulation given the influence parameters. Our algorithm is given by:

Table 23 SODM estimation and opinions' tracking algorithm under full information

Estimation Step

1. Determine C, the number of cores available for computing.

2. Split the J agents of the contact network in batches of size $s_C = \frac{J}{C}$ among the C cores

3. For each group s_C collect only the $e_{i,j}$ information of their immediate neighbours, and their respective measurements.

4. On each core, for each i node in the set s_C let's define $\mathrm{X}_{i,t} = [x_{1,t,1} \dots x_{1,t,n}]$, $\mathrm{O}_{i,t} = [o_{1,t,1} \dots o_{1,t,n}]$, $\Phi_{k,t} = [w_{1,t,1} \dots w_{1,t,r}]$ and $\theta_{i,t} = [a_{i,1} \dots$

$a_{i,J}\ b_{i,1}\ \ldots\ b_{i,K}$] and solve the online filtering problem at each time t:

$$p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t})\middle|g_{i,t}(\mathrm{O}_{i,t}), y_{1:K}(\Phi_{k,t}), \theta_{i,t}\right) \propto$$

$$p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t})\middle|g_{i,t}(\mathrm{X}_{i,t-1}), y_{1:K}(\Phi_{k,t}), \theta_{i,t}\right)p_{i,t}\left(g_{i,t}(o_{i,t})\middle|g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t}\right)p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t-1}\right.$$

with $a_{i,j} = 0\ if\ e_{i,j} \notin \mathrm{Neighbors(i)}$

Using for this procedure:

(i) Set the empirical pdf and cdf prior of $g(.)$ and y based on the observed measurements $\mathrm{O}_{i,t}$ and $\Phi_{k,t}$.

(ii) Use as initial prior for $\theta_i \sim N\left(\widetilde{\theta_{i,0}} = \left[\frac{1}{M+K} \ldots \frac{1}{M+K}\right], V = diag\left(\frac{1}{(M+K)^2}\right)\right)$

(iii) The learning starts at period 25

(iv) Resample the state and parameters using the index $\tau \sim Multinomial(\omega, \mathrm{T})$ where $\omega_i = \frac{p_{i,t}\left(g_{i,t}(o_{i,t})\middle|g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t}\right)}{\sum_{t=1}^{\mathrm{T}} p_{i,t}\left(g_{i,t}(o_{i,t})\middle|g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t}\right)}$

(v) Propagate forward using $p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t}\middle|y_{1:K}(\Phi_{k,t})\right)$ and $p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t})\middle|g_{i,t}(\mathrm{X}_{i,t-1}), y_{1:K}(\Phi_{k,t}), \theta_{i,t}\right)$

(vi) Learn $\theta_{i,t}$ via $p_{i,t}\left(\theta_{i,t}|g_{i,t}(\mathrm{X}_{i,t})\right) = \frac{1}{T}\sum\ p_{i,t}\left(\theta_{i,t}|\ g_{i,t}(\mathrm{X}_{i,t})y_{1:K}(\Phi_{k,t})\right)$

5. At each time t, verify the conditions for the mean value of the parameters

$0 \leq a_{i,j}, b_k \leq 1$ and $\sum_{j=1}^{M} a_{i,j} + \sum_{k=0}^{K} b_k = 1$

and if needed re-scale the results from the parameter vector θ_i to comply with both.

6. Store C lists on each one of the cores with the estimated parameters and states.

Tracking Step

1. Collect all the parameters from the C lists in the first core and arrange it in a matrix form $\hat{A}$ and $\hat{B}$.

2. Set $\widehat{\mathrm{M}}_0 = \begin{bmatrix} g_1(\mathrm{X}_{1,t}) \\ \cdots \\ g_1(\mathrm{X}_{J,t}) \end{bmatrix}$ and $\Lambda_{t+1} = \begin{bmatrix} y_1(\Phi_{k,t+1}) \\ \cdots \\ y_K(\Phi_{k,t+1}) \end{bmatrix}$

3. Simulate $t + m$ periods of the opinion process by using the recursion

$$\widehat{\mathrm{M}}_{t+1} = \widehat{\mathrm{A}}\widehat{\mathrm{M}}_t + \widehat{\mathrm{B}}\Lambda_{t+1}$$

This result approximates the opinion distributions of each agent at time $t + 1, t + 2, \ldots t + m$

4. If needed, for each $t + m$, draw a sample of a desired size m to approximate the mean opinion for each agent at.

We use the statistical software R to construct the estimation algorithm.

Estimation and opinion tracking under incomplete information

In many real life settings, we do not have access to the entire C matrix or to the entire set of opinion measurements for all the agents. In this case, opinions are still not observed, and we can only collect measurements for a subset $\mathcal{J}$ of agents, with cardinality $N < J$, such that for every $j \in \mathcal{J}$ at time t we observe n measurements $o_{j,t,n}$. We do not observe the value of the influence weight $a_{i,j}$ that agent i exerts on agent j; we only observe a partial structure of the contact network C. In this case, we observe the connections between $j \in \mathcal{J}$ agents and their immediate neighbors. This means that we observe a $\tilde{C}$ matrix with dimensions (JXN) and elements $\tilde{c}_{i,j} = \begin{cases} 1 & if\ i\ is\ neighbor\ of\ j \\ 0 & otherwise \end{cases}$. Finally, we observe r values $w_{k,t-1,r}$ from the exogenous function inputs $y_{k,t-1}$, but not their associated coefficients b_k. The general setting of online particle filter is given by.

$$\begin{bmatrix} g_1(x_{1,t,1}, \dots, x_{1,t,n}) \\ \dots \\ g_J(x_{J,t,1}, \dots, x_{J,t,n}) \end{bmatrix}_{J\times 1} = \tilde{L}_{J\times J} \begin{bmatrix} g_1(x_{1,t-1,1}, \dots, x_{1,t-1,n}) \\ \dots \\ g_J(x_{J,t-1,1}, \dots, x_{J,t-1,n}) \end{bmatrix}_{J\times 1} + \tag{4a}$$

$$B_{J\times K} \begin{bmatrix} y_1(w_{1,t,1}, \dots, w_{1,t,r}) \\ \dots \\ y_K(w_{K,t,1}, \dots, w_{K,t,r}) \end{bmatrix}_{K\times 1} + \varphi_{t\,J\times 1}$$

$$\begin{bmatrix} g_1(o_{1,t,1}, \dots, o_{1,t,n}) \\ \dots \\ g_J(o_{J,t,1}, \dots, o_{J,t,n}) \end{bmatrix}_{N\times 1} = \widetilde{H}_{N\times J} \begin{bmatrix} g_1(x_{1,t,1}, \dots, x_{1,t,n}) \\ \dots \\ g_J(x_{J,t,1}, \dots, x_{J,t,n}) \end{bmatrix}_{J\times 1} + \vartheta_{t\,N\times 1} \tag{4b}$$

Since we observe only $\tilde{C}$, we can impose the constraint on the elements of L matrix such that $l_{i,j} = \begin{cases} 0 & if\ e_{i,j} = 0\ and\ e_{i,j} \in \mathcal{J} \\ a_{i,j} & otherwise \end{cases}$ $for\ 0 \le a_{i,j} \le 1$ f. in this case, we have additional $a_{i,j}$ terms to estimate.

In this case, we introduce extra estimation steps to our full information online learning algorithm so we can tackle the incomplete information case. The modified algorithm is given by:

Table 24 SODM estimation and opinions' tracking algorithm under incomplete information

Estimation Step

1. Determine C, the number of cores available for computing.

2. Split the J agents of the contact network in batches of size $s_C = \frac{J}{C}$ among the C cores

3. For each group s_C collect only the available $e_{i,j}$ information of their immediate neighbours, and their respective measurements.

4. On each core, for each i node in the set s_C let's define $\mathrm{X}_{i,t} = [x_{1,t,1} \dots x_{1,t,n}]$, $\mathrm{O}_{i,t} = [o_{1,t,1} \dots o_{1,t,n}]$, $\Phi_{k,t} = [w_{1,t,1} \dots w_{1,t,r}]$ and $\theta_{i,t} = [a_{i,1} \dots a_{i,J}\ b_{i,1} \dots\ b_{i,K}\ ]$ and solve the online filtering problem at each time t:

$$p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t})\middle|g_{i,t}(\mathrm{O}_{i,t}), y_{1:K}(\Phi_{k,t}), \theta_{i,t}\right) \propto$$

$$p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t})\middle|g_{i,t}(\mathrm{X}_{i,t-1}), y_{1:K}(\Phi_{k,t}), \theta_{i,t}\right)p_{i,t}\left(g_{i,t}(o_{i,t})\middle|g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t}\right)p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t-1}\right.$$

with $a_{i,j} = 0\ if\ e_{i,j} \notin \mathrm{Neighbors(i)}$

Using for this procedure:

(i) For the opinion measurements that are observed $i \in \mathcal{J}$, set the empirical pdf and cdf prior of $g(.)$ and y based on the observed

measurements $O_{i,t}$ and $\Phi_{k,t}$

(ii) For the opinion measurements that are not observed $i \notin \mathcal{J}$, set the empirical pdf of each node i by constructing a pool kernel $\sum_{s\ in\ Neighbors(i)} \alpha g_s(.)$ with $\alpha = \frac{1}{card(Neighbors(i))}$ based on the empirical $g(.)$ from the observed measurements of the neighbors of i. If no information is available for the neighbors of i, set $g_i(.) = beta(1,1)$

(iii) Use as initial prior for $\theta_i \sim N\left(\widetilde{\theta_{i,0}} = \left[\frac{1}{M+K} \ldots \frac{1}{M+K}\right], V = diag\left(\frac{1}{(M+K)^2}\right)\right)$

(iv) The learning starts at period 25

(v) Resample the state and parameters using the index $\tau \sim Multinomial(\omega, \mathrm{T})$ where $\omega_i = \frac{p_{i,t}(g_{i,t}(o_{i,t})|g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t})}{\sum_{t=1}^{\mathrm{T}} p_{i,t}(g_{i,t}(o_{i,t})|g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t})}$

(vi) Propagate forward using $p_{i,t}\left(g_{i,t}(\mathrm{X}_{i,t-1}), \theta_{i,t} \middle| y_{1:K}(\Phi_{k,t})\right)$ and $p_{i,t}(g_{i,t}(\mathrm{X}_{i,t})|g_{i,t}(\mathrm{X}_{i,t-1}), y_{1:K}(\Phi_{k,t}), \theta_{i,t})$

(vii) Learn $\theta_{i,t}$ via $p_{i,t}\left(\theta_{i,t}|g_{i,t}(\mathrm{X}_{i,t})\right) = \frac{1}{T}\sum\ p_{i,t}\left(\theta_{i,t}|\ g_{i,t}(\mathrm{X}_{i,t}) y_{1:K}(\Phi_{k,t})\right)$

5. At each time t, verify the conditions for the mean value of the parameters $0 \leq a_{i,j}, b_k \leq 1$ and $\sum_{j=1}^{M} a_{i,j} + \sum_{k=0}^{K} b_k = 1$

and if needed re-scale the results from the parameter vector θ_i to comply with both.

6. Store C lists on each one of the cores with the estimated parameters and states.

Tracking Step

1. Collect all the parameters from the C lists in the first core and arrange it in a matrix form $\hat{A}$ and $\hat{B}$.

2. Set $\widehat{\mathrm{M}}_0 = \begin{bmatrix} g_1(\mathrm{X}_{1,t}) \\ \dots \\ g_1(\mathrm{X}_{J,t}) \end{bmatrix}$ and $\Lambda_{t+1} = \begin{bmatrix} y_1(\Phi_{k,t+1}) \\ \dots \\ y_K(\Phi_{k,t+1}) \end{bmatrix}$

3. Simulate $t + m$ periods of the opinion process by using the recursion

$$\widehat{\mathrm{M}}_{t+1} = \widehat{\mathrm{A}}\widehat{\mathrm{M}}_t + \widehat{\mathrm{B}}\Lambda_{t+1}$$

This result approximates the opinion distributions of each agent at time $t + 1, t + 2, \dots t + m$

4. If needed, for each $t + m$, draw a sample of a desired size m to approximate the mean opinion for each agent at.

We use the statistical software R to construct the estimation algorithm.

Results

Simulation setting

A computational study is conducted to test the estimation methodology and asymptotic properties of our model on midsize networks. First, we simulate the contact network structure and the opinion process using the SODM outlined in section two. For this purpose, a total of 96 scenarios are simulated.

a) Contact Network topology: we use complete (C), random (RN) and free scale (BA) network structures with an average connectivity of 100 nodes, for a fixed size of 10,000 agents.

b) Influence matrix: different influence matrixes are used to capture all the potential variety and influence among agents:

- Type (1) where all the weights for agent i are similar between his neighbours.
- Type (2) where all the weights for agent i are different between his neighbours.

c) One initial opinion conditions are used. The initial opinion of all the agents range between 0 and 1 (emulating a potential opinion index) and it is simulated using a beta distribution. The parameters of the initial beta distribution $beta(a, b)$ are drawn from $a \sim Unif(1,10)$, and the variance from $b \sim Unif(1,10)$

d) Amount of information at hand: there are three types of scenarios.

- Type (a) where the C matrix is observed for levels 5%,25%, 75%, and 100%. The opinion vector for each agent at every period is fully observed
- Type (b) where only for subset of agent J the opinion vector is observed for levels 5%,25%, 75% and 100%. The contact network C is fully observed.
- Type (c) where the combinations of different levels of observed opinions and observed contact network are used.

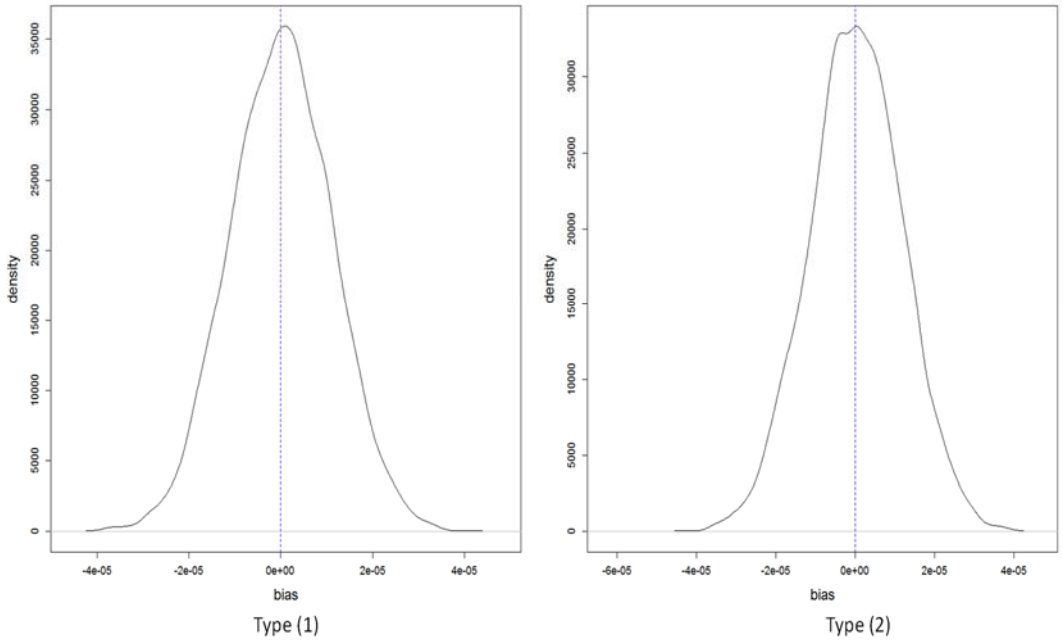
We simulate a total of 1000 periods of interaction; drawing 10 measurements per each agent at each time period.

Identification

Since for each network, we have approximately a total of 1 million parameters to estimate; we evaluate the performance of our method by analyzing the bias distribution of all the parameters. In this sense, a distribution centered in zero and with a low variance is the preferred distribution. We have three cases of analysis.

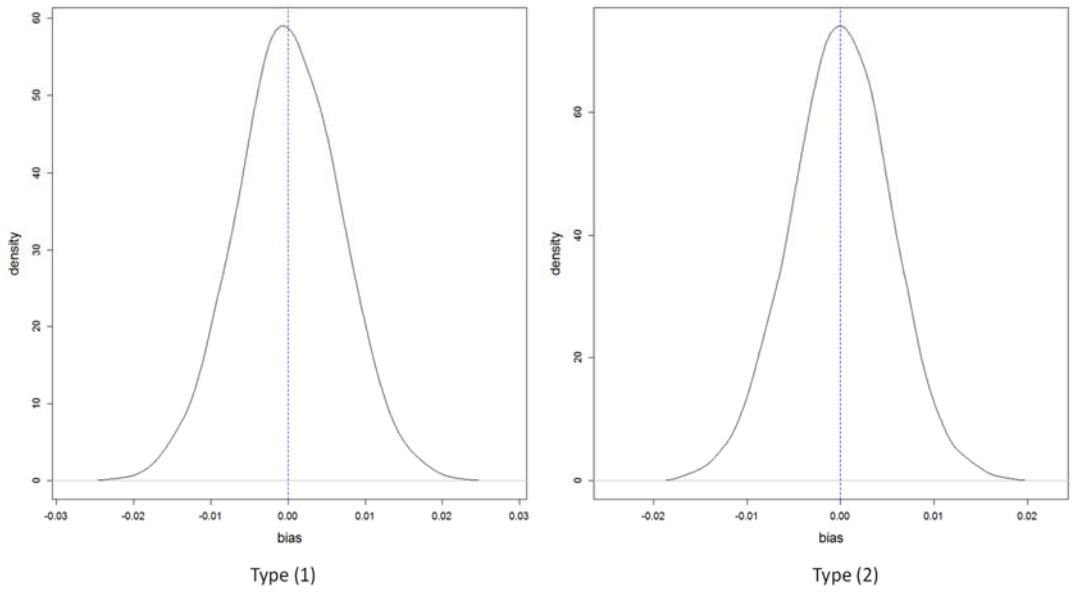
The first case is when we have access to the complete opinion measurements and contact network. In this case, we applied the online learning algorithm of table 1. The results show that the learning is possible for all the three network topologies. The main difference lies in the variance of the estimates. The impact of influence network type and contact network is presented in Fig.32, Fig.33 and Fig.34.

Figure 33 Bias distribution of the estimated mean parameters for a Complete Network of 10,000 agents after 1,000 periods of online learning with complete information



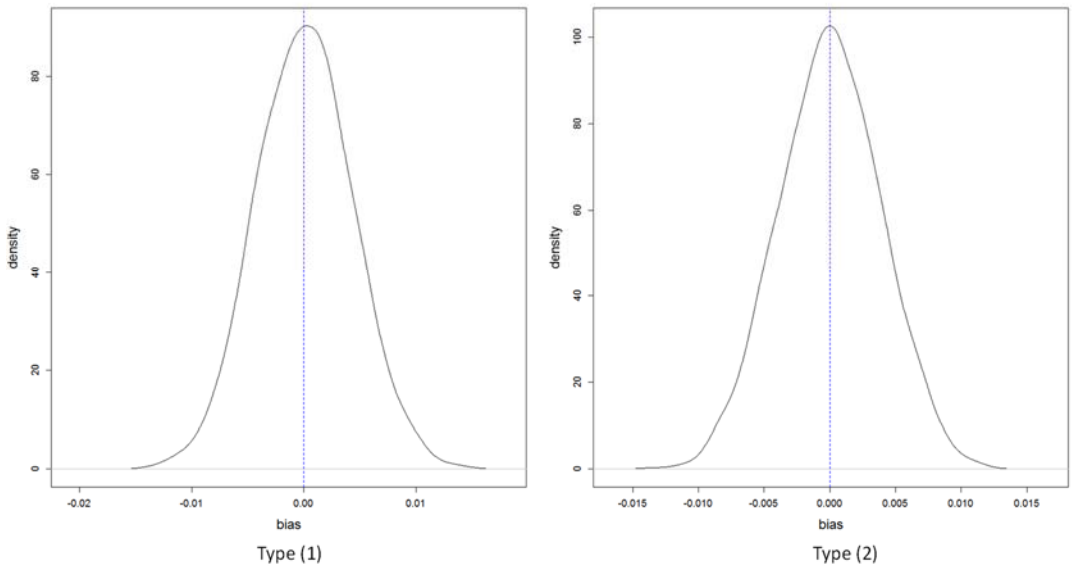
All of the 95% confidence intervals for the all the parameters include the true parameter value, indicating that the parameters are identifiable. In the case of the complete influence network, thought the parameters are still identified, the confidence interval is the widest (when compared as a ratio against the actual true values) among all the network types used. The main reason for this behavior is the larger number of parameters that need to be estimated in this case.

Figure 34 Bias distribution of the estimated mean parameters for a Random Network of 10,000 agents after 1,000 periods of online learning with complete information



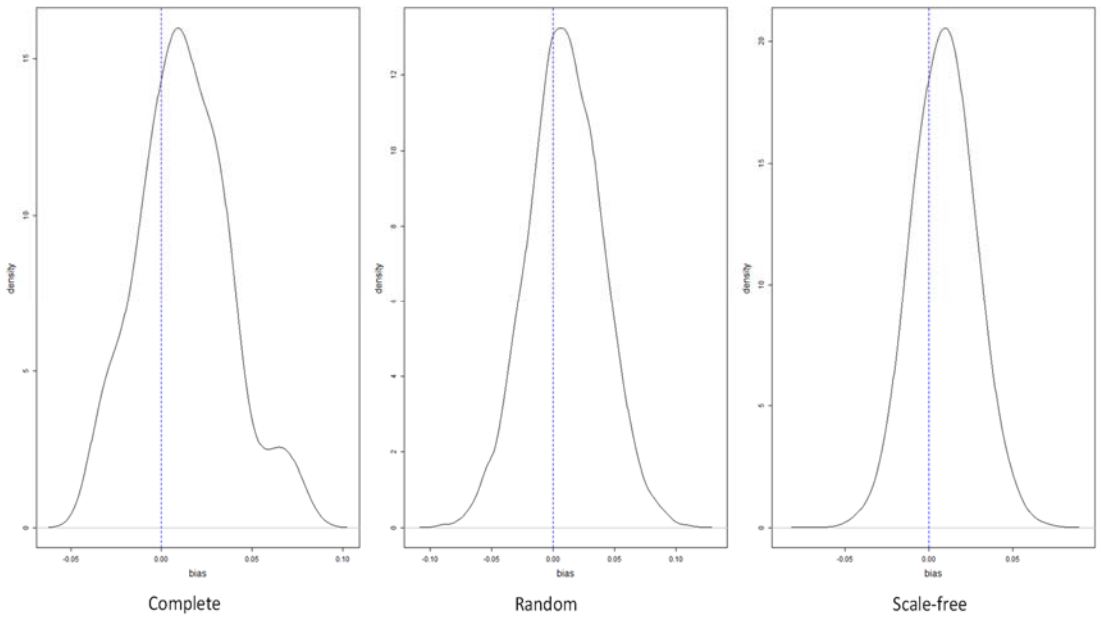
The width of the confidence interval for the parameters of the random network is slightly greater than the interval for the scale-free network. In addition, there is no noticeable difference between type (1) and type (2) values for the influence weights.

Figure 35 Bias distribution of the estimated mean parameters for a Scale-free Network of 10,000 agents after 1,000 periods of online learning with complete information



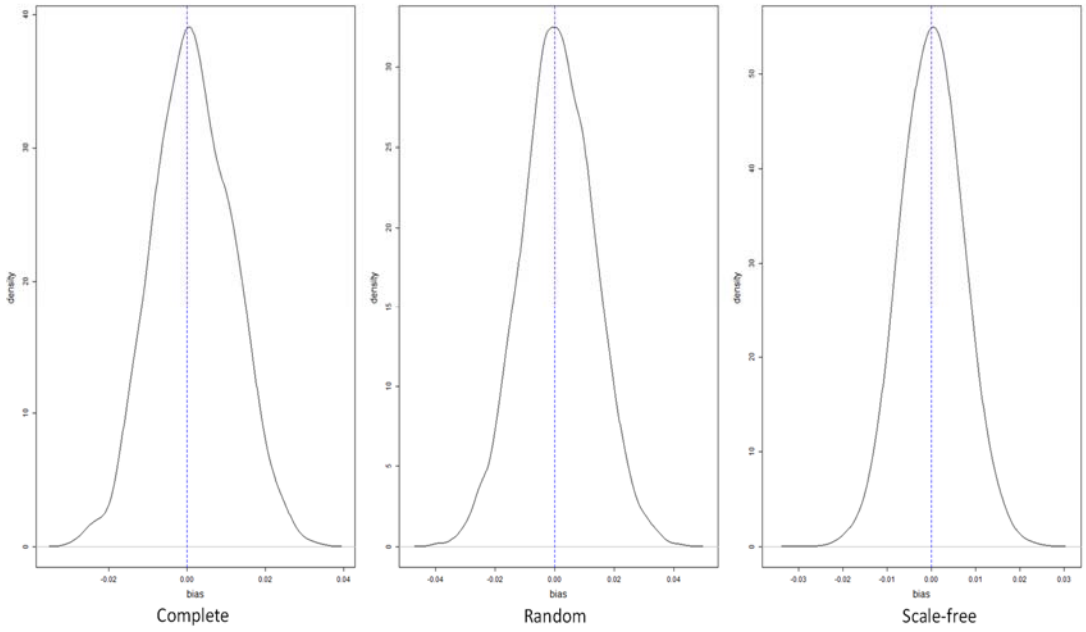
Our second case of analysis occurs when partial information of the contact network is observed, but full information of the opinion measurements can be collected. In this case, it is not possible (worst performance) to recover parameters under complete network when the observed contact network is 5%. When only 25% of C is observed, the parameters have high variability. This result is similar for all type of networks. Since there is no noticeable difference between type (1) and type (2) values for the influence weights, and type (2) represents a more general case, we restricted ourselves to report for the rest of the analysis the bias distribution for type (2).

Figure 36 Bias distribution of type (2) estimated mean parameters for a network of 10,000 agents after 1,000 periods of online learning observing 25% of C



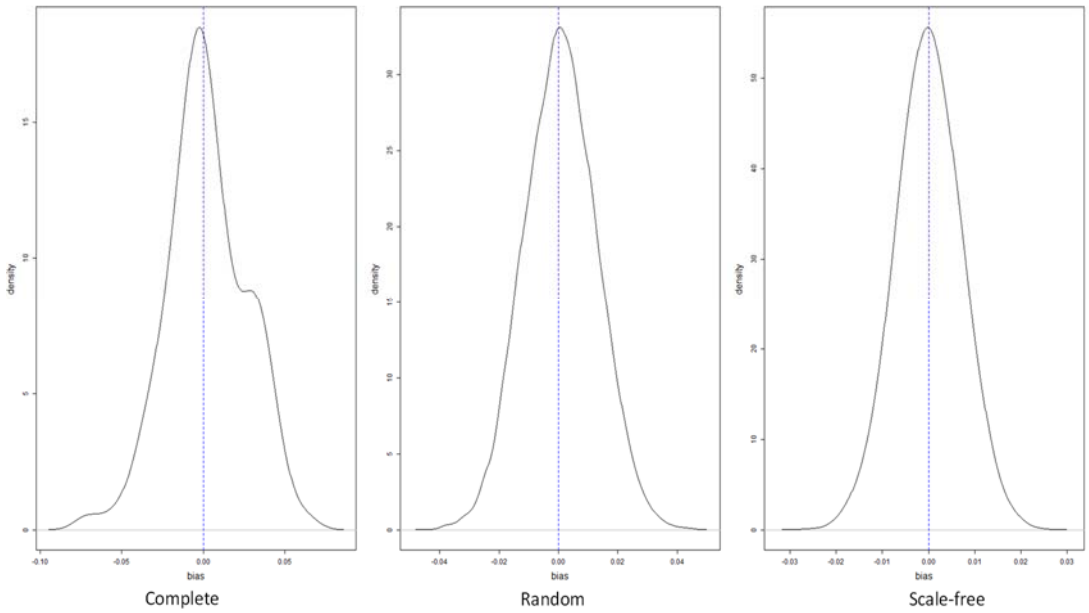
The parameters has a moderate variability when the 75% of C is observed when the true structure is a complete network. This variation decreases in the case of random and scale-free network.

Figure 37 Bias distribution of type (2) estimated mean parameters for a network of 10,000 agents after 1,000 periods of online learning observing 75% of C



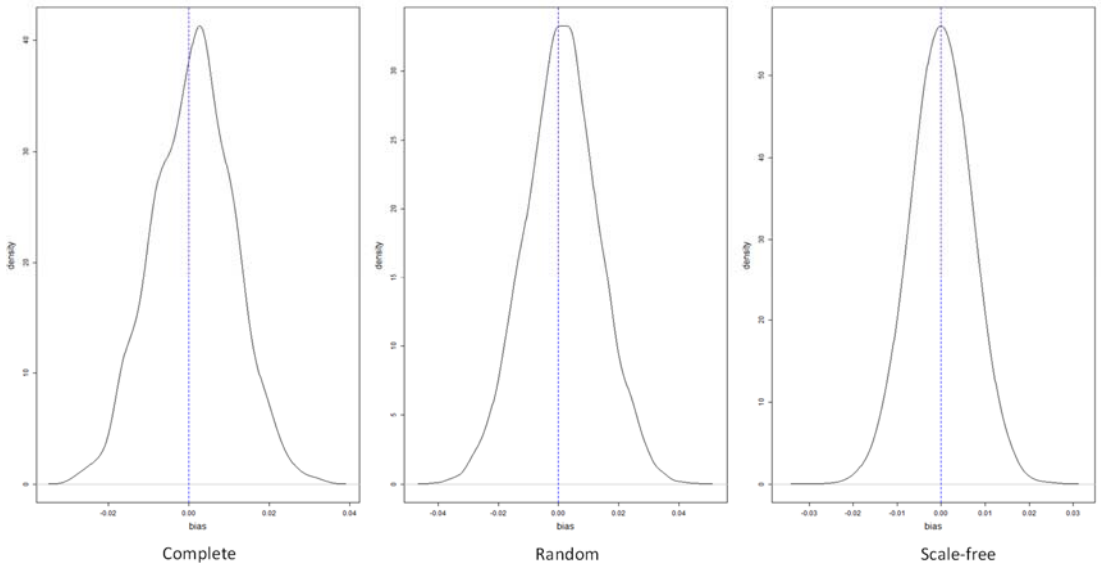
The third case of analysis is when we have incomplete information for the opinion measurements and the contact network simultaneously. In this situation it is still not possible (worst performance) to recover the parameters under combinations of opinion (5%, 25%) and observed C (5%, 25%). The results in this cases are non-informative. When only 25% of C is observed and 75% of the measurements are collected, the parameters show a high variability for the complete network, and high to moderate variability for the random and scale-free network. These results are shown in Fig.37.

Figure 38 Bias distribution of type (2) estimated mean parameters for a network of 10,000 agents after 1,000 periods of online learning observing 25% of C and 75% of opinion measurements



When we can only access 75% of C and 75% of the measurements, our second algorithm recovers the influence parameters with a low (random and scale-free network) to moderate (complete network) variability. Fig. 38 depict these findings.

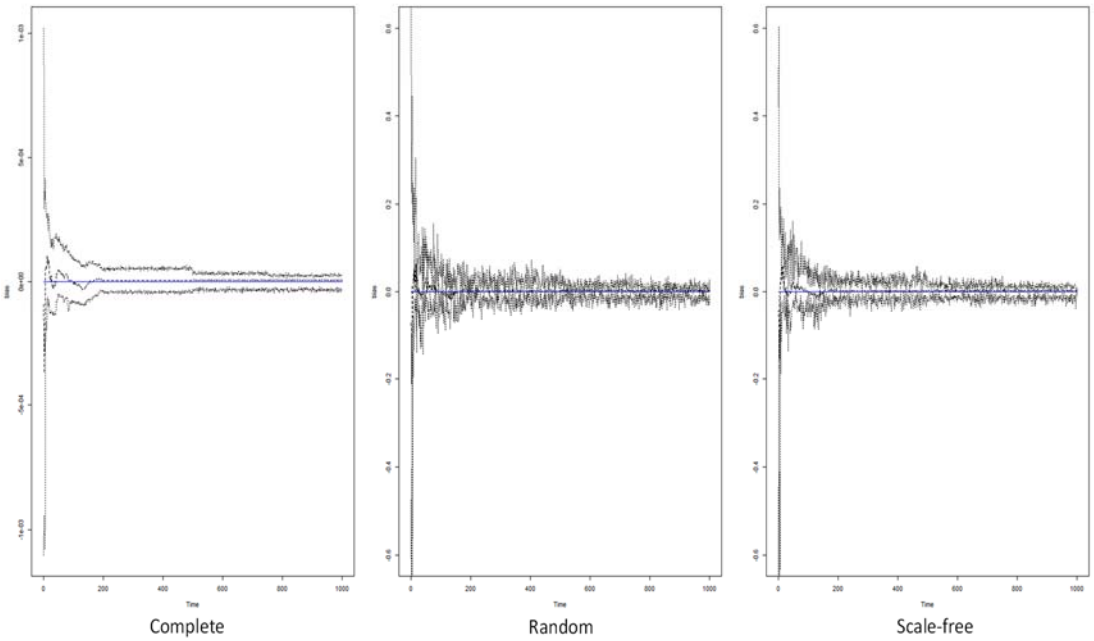
Figure 39 Bias distribution of type (2) estimated mean parameters for a network of 10,000 agents after 1,000 periods of online learning observing 25% of C and 75% of opinion measurements



5.3 Asymptotic properties

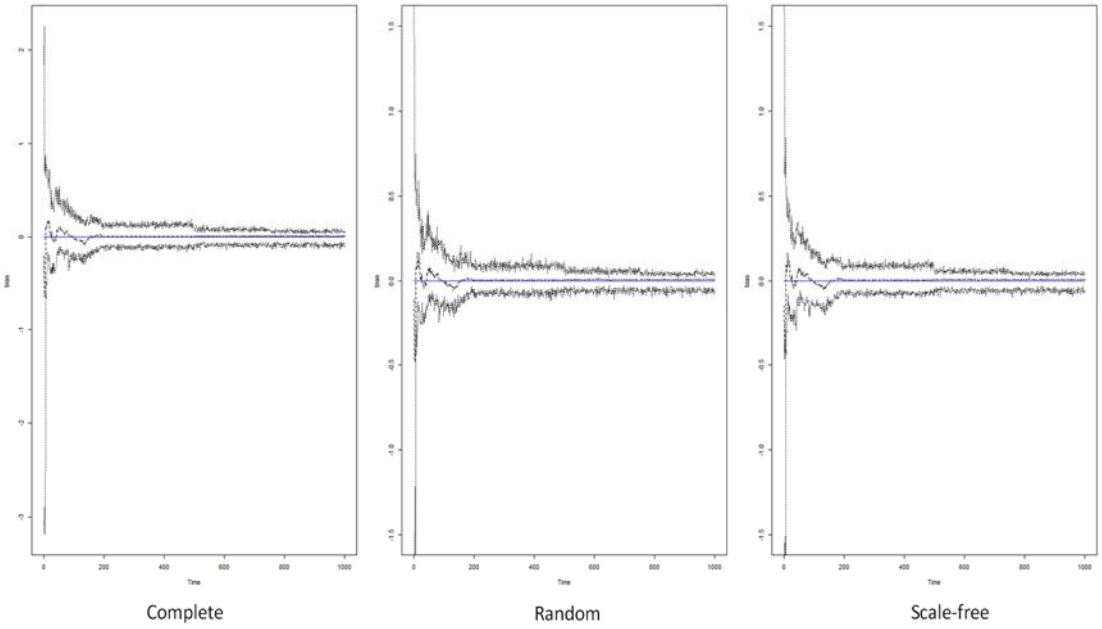
We analyze the asymptotic properties of our estimation algorithm by looking at the learning trajectory of our three different data cases. When we have complete information, our algorithm requires approximately 50 periods to learn accurately the parameters of a random and free scale network. In the case of the complete network, the accurate learning starts around 200 periods. As more information is available, the results show that the estimation recovers the true parameters and its precision improves asymptotically. The model provides consistent estimates of the true parameters for all the different contact networks. In this case we use evolution-trend graphs to show the properties of our algorithm. Fig. 8 shows the trend of the 95% confidence intervals and mean trajectory of the parameters' bias.

Figure 40 Evolution of the bias distribution of type (2) estimated mean parameters and 95% C.I for a network of 10,000 agents with complete information



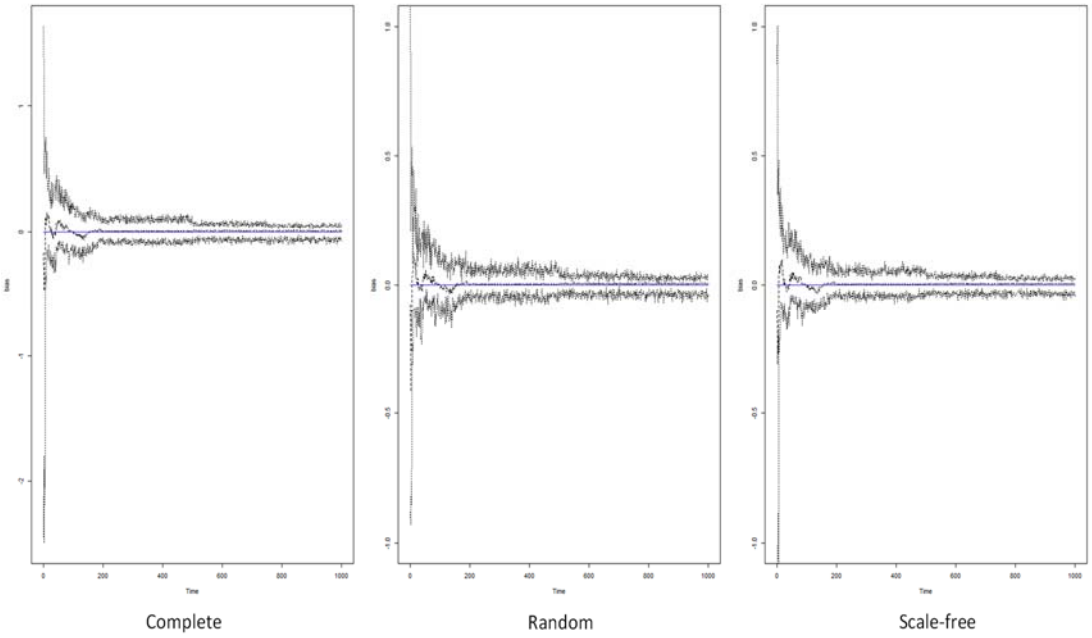
In the second case of analysis, the results are mixed. When only 25% of C is observed, the parameters show a slow learning rate. This result is similar for all type of networks. The parameter learning improves as new info is available, but not that much.

Figure 41 Evolution of the bias distribution of type (2) estimated mean parameters and 95% C.I for a network of 10,000 agents observing 25% of C



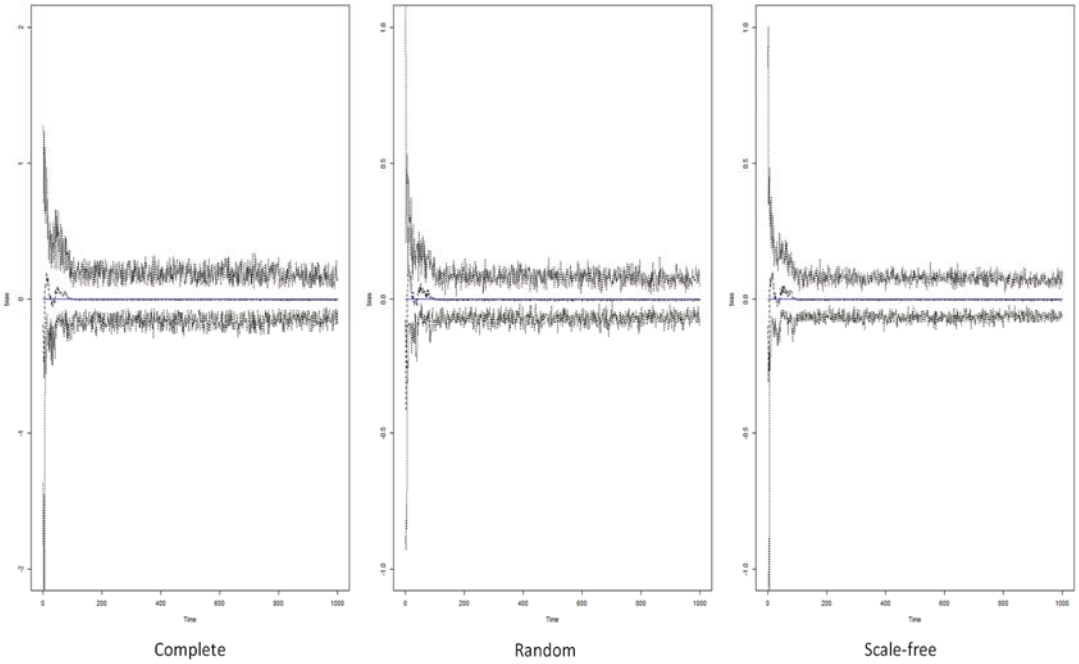
When 75% of C is observed, the learning trend improves over time. The parameters show a learning evolution from high to moderate variability. This improvement is the greatest in the case of random and scale-free network.

Figure 42 Evolution of the bias distribution of type (2) estimated mean parameters and 95% C.I for a network of 10,000 agents observing 75% of C



Unfortunately the lowest learning rate is present when we observed partially information from opinion measurements and contact network. When only 75% of the CN is observed and 75% of the measurements are collected, the learning rate is slow. The estimated parameters show a transition between high variability to low variability for all type of networks. Our second algorithm recovers the influence parameters with a low (random and scale-free network) to moderate (complete network) variability only after all the available information has been used (1000 periods). This findings are collected in Fig. 42.

Figure 43 Evolution of the bias distribution of type (2) estimated mean parameters and 95% C.I for a network of 10,000 agents observing 25% of C and 75% of opinion measurements



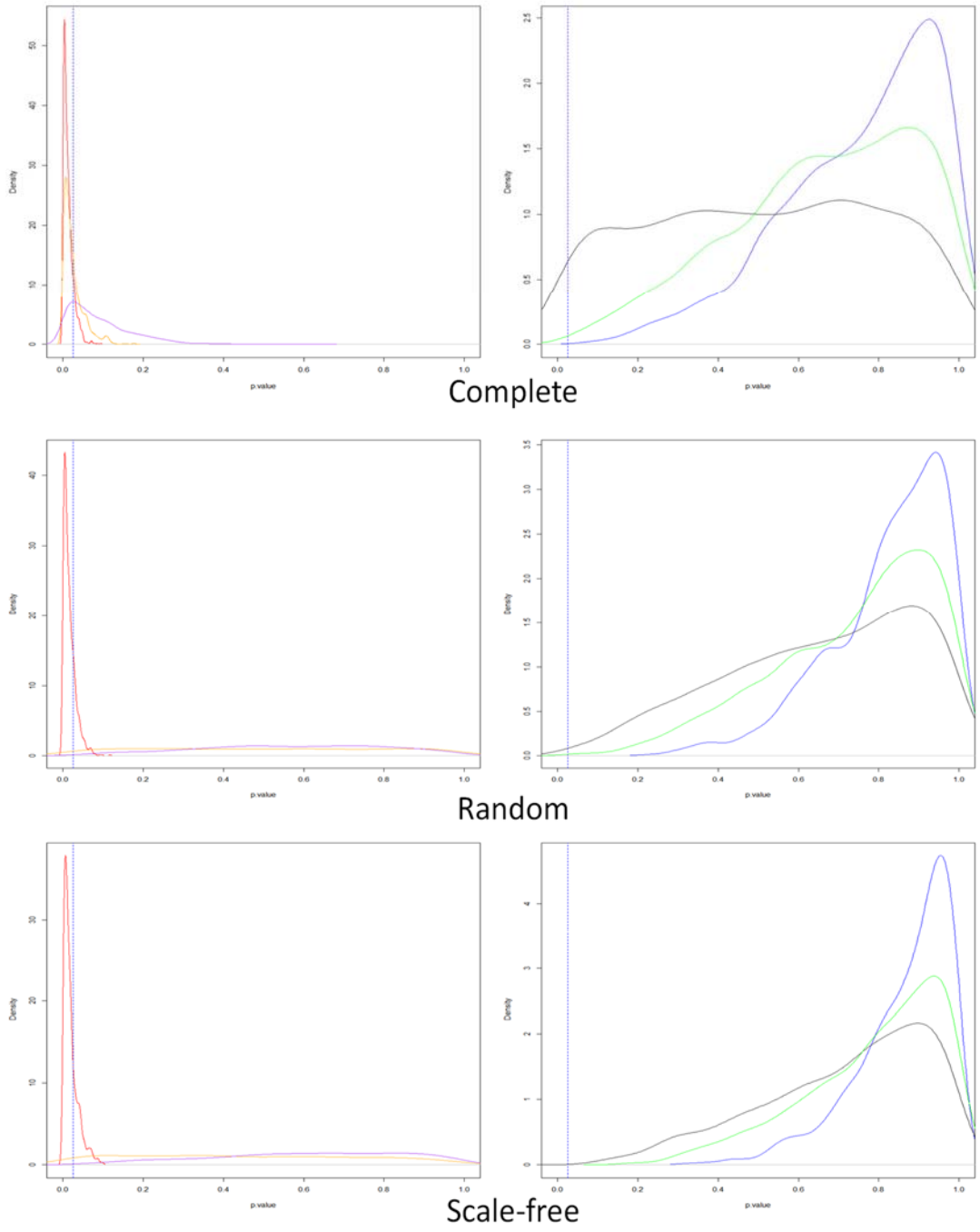
Opinion tracking

Finally, we extend our simulation analysis to study the accuracy of our method in recovering the opinion distribution of the agents. From our simulation experiment, we know the true empirical pdf of the opinions of each agent and from our algorithms we know the estimated $g(.)$ functions. For comparison and reporting purposes we plot the distribution of p-value of the Kolmogorov-Smirnov statistic for all the agents, at different time periods. A distribution with most of its density close to one and with almost no area below the critical p-value (we choose 0.025 as a reference mark) represents an accurate estimation of the individual opinion pdfs.

When we have complete information, our algorithm requires approximately 50 periods to learn accurately the opinion distributions when the network topology is random or free scale. In the case of the complete network, the accurate learning starts around 250 periods.

Fig. 43 shows that as the amount of information increases, the KS distribution shows that the online learning recovers asymptotically the true unobserved opinion distributions.

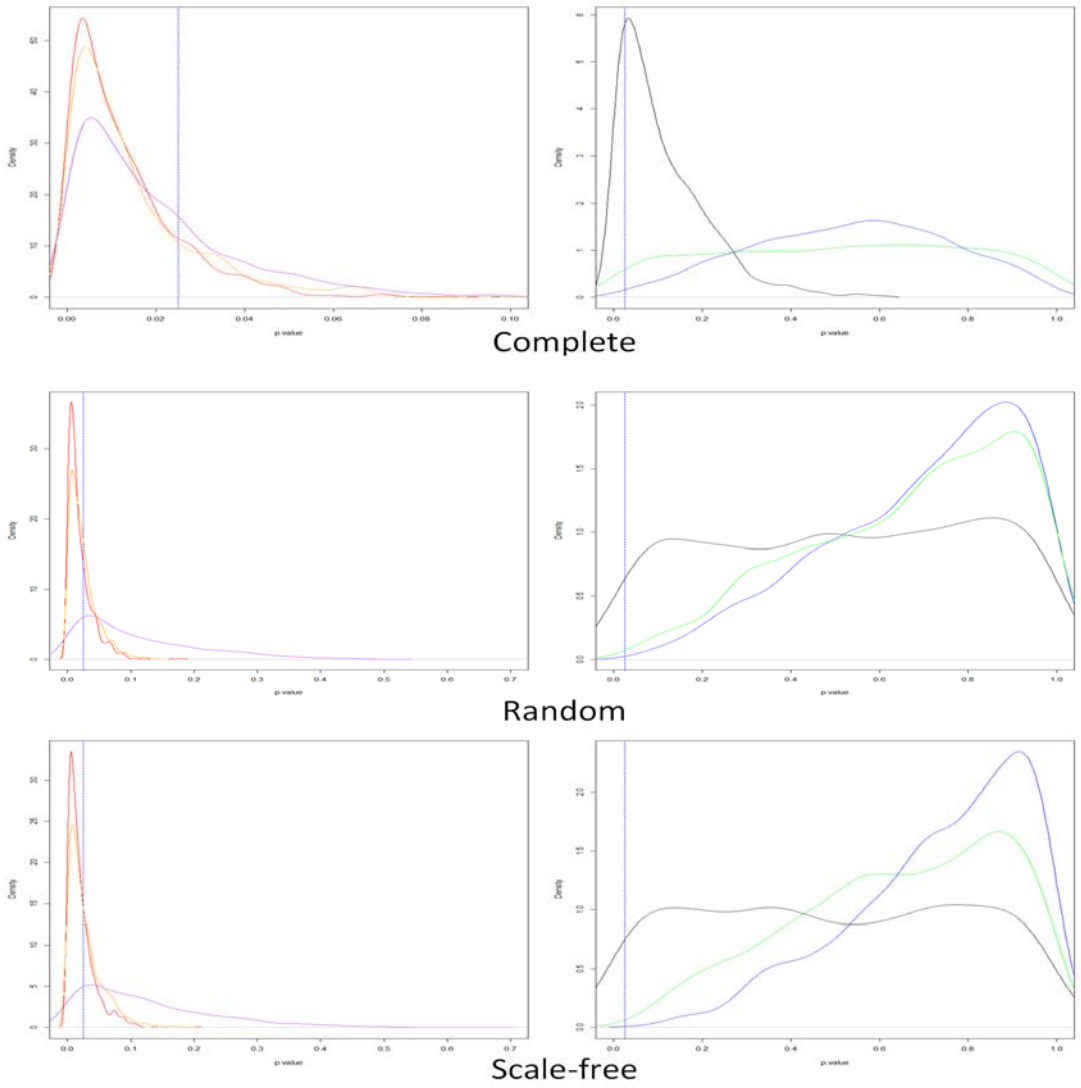
Figure 44 Evolution of the KS distribution of the estimated opinion distributions for type (2) under complete information



25 -- , 50 --, 100 --, 250 --, 500 -- and 1,000 -- interaction periods

In the second case of analysis, the results are not promising when only 25% of C is observed. The unobserved opinions show a slow learning rate. This result is similar for all type of networks. Nevertheless, when 75% of C is observed, the learning trend improves over time. The KS distribution show a learning evolution from having p-values close to the critical value to moderate values. The true opinion distribution is recovered more accurately in the case of random and scale-free network.

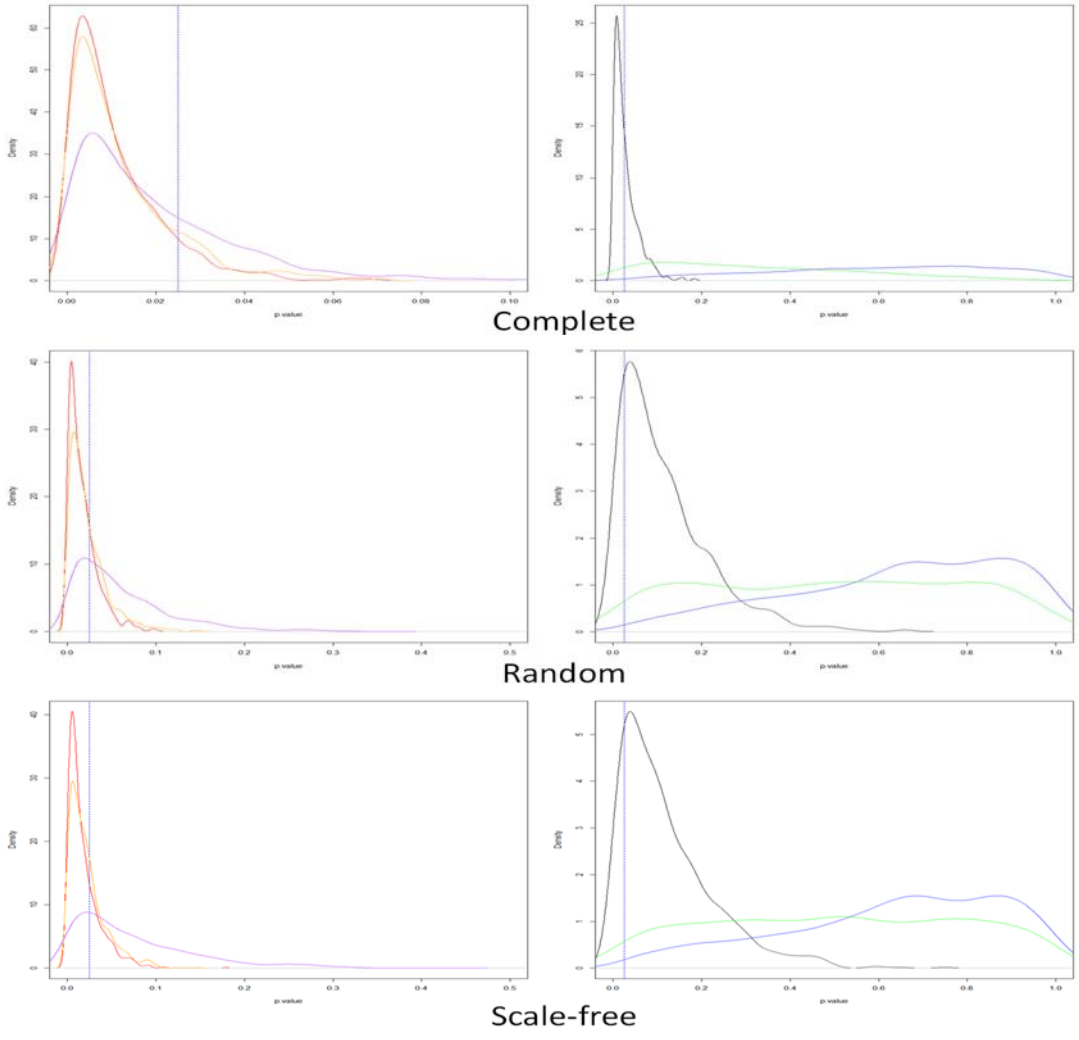
Figure 45 Evolution of the KS distribution of the estimated opinion distributions for type (2) under complete information observing 75% of C



25 -- , 50 --, 100 --, 250 --, 500 -- and 1,000 -- interaction periods

The lowest learning rate is present when we observed partially information from opinion measurements and contact network. When only 75% of the CN is observed and 75% of the measurements are collected, the learning rate of the unobserved opinion distribution is slow. The estimated parameters show a transition between a distribution cluster around the critical value to a distribution with a small area below the critical value but with a moderate variability for all the network topologies. Fig. 45 shows these results.

Figure 46 Evolution of the KS distribution of the estimated opinion distributions for type (2) under complete information observing 25% of C and 75% of opinion measurements



25 -- , 50 --, 100 --, 250 --, 500 -- and 1,000 -- interaction periods

Conclusion

We propose an online particle filter learning algorithm for the estimation of the influence matrix and tracking of opinion distributions for each agent of a social network. Our theoretical opinion model is grounded on the idea that the opinions of each agent can be modelled as individual probability distributions. Our theoretical model is functional; then, we develop a filter algorithm to learn the parameters of the model as new information becomes available.

The model was tested for identification, asymptotic, stability, and online tracking. Given the particle filter strategy, the results are contingent on the amount of information available for the estimation procedure. We propose two general online filtering algorithms, depending whether we observe complete or incomplete information. In the case of complete information we are able to recover all the parameters and opinion distributions accurately. In addition, the learning rate of the parameters and states is fast, learning them accurately only after 25 periods of information. For the incomplete information case, we can only learn the parameters and unobserved opinion distributions if we observed at least 75% of the contact network, measurement opinions or both. The learning rate demands at least 250 periods of information to recover the parameters and states with moderate to high precision.

Both algorithms offer the following computational advantages: (a) the estimation of the influence parameters for each agent only requires local information for deriving efficient estimates; (b) (c) the evolution of the opinion distribution can be recovered and tracked by Monte Carlo simulation once the parameters has been estimated. From a computational perspective, the estimation algorithm is scalable and compatible with requirements for

distributed computing. The estimation procedure for N agents can be estimated as N loosely synchronized problems and this enables the efficient use of all cores and processors available for computation.

From an information perspective, the numerical experiment shows that $10\times(0.75\ neighbors(i))\times\#external\ information\ sources$ data points are the minimum size requirement for each agent to estimate our model with accurate mean point estimates and asymptotic results. In the case of a network with 10,000 agents and average connectivity of 100, this condition translates into approximately 750 observations. This condition is contingent upon the size of the social network, the number of neighbours per agent and the opinion process that is being modelled; therefore, it is case specific. These results confirm that the convergence ratio depends on a large t, on the number n of measurements , and on the reliability of the measurement opinions at each t.

WORKS CITED

Acemoglu, D., M. A. Dahleh, and I. Lobel. "Ozdaglar.(2008). Bayesian learning in social networks (Working Paper No. 14040)." (2009).

Acemoğlu, Daron, Giacomo Como, Fabio Fagnani, and Asuman Ozdaglar. 2013. "Opinion Fluctuations and Disagreement in Social Networks." *Mathematics of Operations Research 38* (1):1-27. doi: 10.1287/moor.1120.0570.

Admiraal, Ryan, and Mark S Handcock. 2008. "Networksis: A Package to Simulate Bipartite Graphs with Fixed Marginals through Sequential Importance Sampling." *Journal of Statistical Software* 24 (8).

Albert, Réka, and Albert-László Barabási. 2002. "Statistical Mechanics of Complex Networks." *Reviews of Modern Physics* 74:47.

Anderson, A, S Goel, J Hofman, and D Watts. 2013. "The Structural Virality of Online Diffusion." Under review.

Anderson, Theodore Wilbur. 1989. "Linear Latent Variable Models and Covariance Structures." *Journal of Econometrics* 41 (1):91-119.

Apolloni, A., K. Channakeshava, L. Durbeck, M. Khan, C. Kuhlman, B. Lewis, and S. Swarup. 2009. "A Study of Information Diffusion over a Realistic Social Network Model." *International Conference on Computational Science and Engineering, 2009*. CSE '09, 2009.

Aral, Sinan, and Dylan Walker. 2014. "Tie Strength, Embeddedness, and Social Influence: A Large-Scale Networked Experiment." *Management Science* 60 (6):1352-1370.

Askari-Sichani, Omid, and Mahdi Jalili. 2013. "Large-Scale Global Optimization Through Consensus of Opinions Over Complex Networks." *Complex Adaptive Systems Modeling* 1 (1). doi: 10.1186/2194-3206-1-11.

Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. 2010. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." *LREC*.

Backstrom, Lars, Paolo Boldi, Marco Rosa, Johan Ugander, and Sebastiano Vigna. "Four Degrees of Separation." *In Proceedings of the 4th Annual ACM Web Science Conference*, pp. 33-42. ACM, 2012.

Bader, David A, and Kamesh Madduri. 2006. "Parallel algorithms for evaluating centrality indices in real-world networks." *2006 International Conference on Parallel Processing (ICPP'06)*.

Bailey, Norman TJ. 1975. The mathematical theory of infectious diseases and its applications: Charles Griffin & Company Ltd, 5a Crendon Street, High Wycombe, Bucks HP13 6LE.

Balmer, Michael, Kai Nagel, and Bryan Raney. "Large-scale Multi-agent Simulations for Transportation Applications." *Intelligent Transportation Systems*, vol. 8, no. 4, pp. 205-221. Taylor & Francis Group, 2004.

Banerjee, Abhijit V. 1992. "A Simple Model of Herd Behavior." *The Quarterly Journal of Economics* 107 (3):797-817. doi: 10.2307/2118364.

Bapna, Ravi, and Akhmed Umyarov. 2015. "Do your online friends make you pay? A Randomized Field Experiment on Peer Influence in Online Social Networks." *Management Science* 61 (8):1902-1920.

Barabási, Albert-László, and Réka Albert. 1999. "Emergence of Scaling in Random Networks." *Science* 286 (5439):509-512. doi: 10.1126/science.286.5439.509.

Barrett, Christopher L, Keith R Bisset, Stephen G Eubank, Xizhou Feng, and Madhav V Marathe. 2008. "EpiSimdemics: an Efficient Algorithm for Simulating the Spread of Infectious Disease over Large Realistic Social Networks." *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*.

Bass, Frank M. 1969. "A New Product Growth for Model Consumer Durables." *Management science* 15 (5):215-227.

Benczik, I. J., S. Z. Benczik, B. Schmittmann, and R. K. P. Zia. 2008. "Lack of Consensus in social systems." *EPL (Europhysics Letters)* 82 (4). doi: 10.1209/0295-5075/82/48006.

Bhatt, Sandeep, Richard Fujimoto, Andy Ogielski, and Kalyan Perumalla. 1998. "Parallel Simulation Techniques for Large-scale Networks." *IEEE Communications Magazine* 36 (8):42-47.

Bickel, Peter J, and Aiyou Chen. 2009. "A Nonparametric View of Network Models and Newman–Girvan and other Modularities." *Proceedings of the National Academy of Sciences* 106 (50):21068-21073.

Bikhchandani, Sushil, David Hirshleifer, and Ivo Welch. 1992. "A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades." *Journal of Political Economy*:992-1026.

Bisset, Keith R, Jiangzhuo Chen, Xizhou Feng, VS Kumar, and Madhav V Marathe. 2009. "EpiFast: a Fast Algorithm for Large Scale Realistic Epidemic Simulations on Distributed Memory Systems." *Proceedings of the 23rd international conference on Supercomputing*.

Boyles, Stephen D., and Tarun Rambha. "A Note on Detecting Unbounded Instances of the Online Shortest Path Problem." *Networks* (2016).

Brandes, Ulrik, and Christian Pich. 2007. "Centrality Estimation in Large Networks." *International Journal of Bifurcation and Chaos* 17:2303-2318.

Budescu, David V, and Eva Chen. 2014. "Identifying Expertise to Extract the Wisdom of Crowds." *Management Science* 61 (2):267-280.

Budescu, David V, and Adrian K Rantilla. 2000. "Confidence in Aggregation of Expert Opinions." *Acta Psychologica* 104 (3):371-398.

Budescu, David V, and Hsiu-Ting Yu. 2007. "Aggregation of Opinions based on Correlated Cues and Advisors." *Journal of Behavioral Decision Making* 20 (2):153-177.
Butts, Carter T. 2008a. "network: a Package for Managing Relational Data in R." *Journal of Statistical Software* 24 (2):1-36.

Butts, Carter T. 2008b. "Social Network Analysis with sna." *Journal of Statistical Software* 24 (6):1-51.

Campbell, CA. 1967. "Towards a Definition of Belief." *The Philosophical Quarterly* (1950-) 17 (68):204-220.

Carvalho, Carlos, Michael S Johannes, Hedibert F Lopes, and Nick Polson. 2010. "Particle Learning and Smoothing." *Statistical Science* 25 (1):88-106.

Carvalho, Carlos M, Hedibert F Lopes, Nicholas G Polson, and Matt A Taddy. 2010. "Particle Learning for General Mixtures." *Bayesian Analysis* 5 (4):709-740.

Cha, Meeyoung, Alan Mislove, and Krishna P. Gummadi. "A Measurement-driven Analysis of Information Propagation in the Flickr Social Network." *Proceedings of the 18th International Conference on World Wide Web*, pp. 721-730. ACM, 2009.

Chambers, Simone. 2003. "Deliberative Democratic Theory." *Annual Review of Political Science* 6 (1):307-326.

Chen, Wei, Chi Wang, and Yajun Wang. "Scalable Influence Maximization for Prevalent Viral Marketing in Large-scale Social Networks." *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1029-1038. ACM, 2010.

Chen, Xinlei, Yuxin Chen, and Ping Xiao. 2013. "The Impact of Sampling and Network Topology on the Estimation of Social Intercorrelations." *Journal of Marketing Research* 50 (1):95-110.

Cohen, Jacob, Patricia Cohen, Stephen G West, and Leona S Aiken. 1983. "Multiple Regression/Correlation Analysis for the Behavioral Sciences." *Hillsdale, NJ: Earlbaum*.

Coleman, James, Elihu Katz, and Herbert Menzel. 1957. "The Diffusion of an Innovation Among Physicians." *Sociometry* 20:253-270.

Condon, Anne, and Michael Saks. 2004. "A Limit Theorem for Sets of Stochastic Matrices." *Linear Algebra and its Applications* 381:61-76.

Cooksey, Ray W. 1996. "The Methodology of Social Judgement Theory." *Thinking & Reasoning* 2 (2-3):141-174.

Costa, Paolo, Luca Mottola, Amy L. Murphy, and Gian Pietro Picco. "Programming wireless sensor networks with the TeenyLIME middleware." In ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing, pp. 429-449. Springer Berlin Heidelberg, 2007.

Crobak, Joseph R., Jonathan W. Berry, Kamesh Madduri, and David A. Bader. "Advanced Shortest Paths Algorithms on a Massively-multithreaded Architecture." *IEEE International Parallel and Distributed Processing Symposium*, pp. 1-8. IEEE, 2007.

Csardi, Gabor, and Tamas Nepusz. 2006. "The igraph Software Package for Complex Network Research." *InterJournal, Complex Systems* 1695 (5):1-9.

Dabarera, Ranga, Kamal Premaratne, Manohar N Murthi, and Dilip Sarkar. 2016. "Consensus in the Presence of Multiple Opinion Leaders: Effect of Bounded Confidence." *IEEE Transactions on Signal and Information Processing over Networks* 2 (3).

Dabek, Frank, Russ Cox, Frans Kaashoek, and Robert Morris. "Vivaldi: A Decentralized Network Coordinate System." *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 15-26. ACM, 2004.

de Solla Price, Derek J. 1965. "Networks of Scientific Papers." *Science* 149 (3683):510-515.

Deffuant, Guillaume, Frederic Amblard, and Gerard Weisbuch. 2004. "Modelling Group Opinion Shift to Extreme : the Smooth Bounded Confidence Model." arXiv:cond-mat/0410199.

Deffuant, Guillaume, Frédéric Amblard, Gérard Weisbuch, and Thierry Faure. 2002. "How Can Extremism Prevail? A Study based on the Relative Agreement Interaction Model." *Journal of artificial societies and social simulation* 5 (4).

Deffuant, Guillaume, David Neau, Frederic Amblard, and Gérard Weisbuch. 2000. "Mixing Beliefs Among Interacting Agents." *Advances in Complex Systems* 03 (01n04):87-98. doi: 10.1142/S0219525900000078.

DeGroot, Morris H. 1974. "Reaching a Consensus." *Journal of the American Statistical Association* 69 (345):118-121.

DeGroot, Morris H, and Julia Mortera. 1991. "Optimal Linear Opinion Pools." *Management Science* 37 (5):546-558.

Dekker, A. H. "Network Centrality and Super-spreaders in Infectious Disease Epidemiology." *20th International Congress on Modelling and Simulation (MODSIM2013)*. 2013.

Dempster, Arthur P. 1967. "Upper and Lower Probabilities Induced by a Multivalued Mapping." *The Annals of Mathematical Statistics*:325-339.

Dijkstra, Edsger W. 1959a. "A Note on Two Problems in Connexion with Graphs." *Numerische mathematik* 1 (1):269-271.

Douglass, Rodney B, Martin Fishbein, and Icek Ajzen. 1977. Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research. JSTOR.

Ebbes, Peter, Zan Huang, and Arvind Rangaswamy. 2013. "Subgraph Sampling Methods for Social Networks: The Good, the Bad, and the Ugly." *HEC Paris Research Paper* No. MKG-2014-1027.

Ehrhardt, George, Matteo Marsili, and Fernando Vega-Redondo. 2006. "Diffusion and Growth in an Evolving Network." *International Journal of Game Theory* 34 (3):383-397. doi: 10.1007/s00182-006-0025-6.

Ellison, Glenn, and Drew Fudenberg. 1995. "Word-of-mouth Communication and Social Learning." *The Quarterly Journal of Economics*:93-125.

Enders, Craig K. 2001. "A Primer on Maximum Likelihood Algorithms Available for Use with Missing Data." *Structural Equation Modeling* 8 (1):128-141.

Erdős, Paul, and Miklós Simonovits. 1965. "A Limit Theorem in Graph Theory." *Studia Sci. Math. Hung*.

Ferraty, Frédéric, and Philippe Vieu. Nonparametric functional data analysis: theory and practice. Springer Science & Business Media, 2006.

Fishbein, Martin, and Bertram H. Raven. "The AB Scales: An Operational Definition of Belief and Attitude." *Human Relations* (1962).

Floyd, Robert W. 1962. "Algorithm 97: Shortest Path." *Communications of the ACM* 5:345.

Fortunato, Santo. 2004. "Universality of the Threshold for Complete Consensus for the Opinion Dynamics of Deffuant et al." *International Journal of Modern Physics C* 15 (09):1301-1307.

Frank, Ove. 1979. "Sampling and Estimation in Large Social Networks." *Social networks* 1:91-101.

Fu, Feng, Lianghuan Liu, and Long Wang. 2008. "Empirical Analysis of Online Social Networks in the Age of Web 2.0." *Physica A: Statistical Mechanics and its Applications* 387:675-684.

Fujimoto, Richard M, Kalyan Perumalla, Alfred Park, Hao Wu, Mostafa H Ammar, and George F Riley. 2003. "Large-scale Network Simulation: how big? how fast?" *Modeling, Analysis and Simulation of Computer Telecommunications Systems*, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on.

Galam, Serge. 1997. "Rational Group Decision Making: A Random Field Ising Model at T = 0." *Physica A: Statistical Mechanics and its Applications* 238 (1–4):66-80. doi: 10.1016/S0378-4371(96)00456-6.

Gale, Douglas, and Shachar Kariv. 2003. "Bayesian Learning in Social Networks." *Games and Economic Behavior* 45 (2):329-346.

Gelper, Sarah, and Stefan Stremersch. 2014. "Variable Selection in International Diffusion Models." *International Journal of Research in Marketing* 31 (4):356-367.

Gerard, Harold B, and Ruben Orive. 1987. "The Dynamics of Opinion Formation." *Advances in Experimental Social Psychology* 20:171-202.

Goel, Sharad, and Daniel G Goldstein. 2013. "Predicting Individual Behavior with Social Networks." *Marketing Science* 33 (1):82-93.

Goldenberg, Anna, Alice X Zheng, Stephen E Fienberg, and Edoardo M Airoldi. 2010. "A Survey of Statistical Network Models." *Foundations and Trends® in Machine Learning* 2 (2):129-233.

Goldfarb, Donald, and Ashok Idnani. 1982. "Dual and Primal-dual Methods for Solving Strictly Convex Quadratic Programs." *In Numerical Analysis*, 226-239. Springer.

Goldfarb, Donald, and Ashok Idnani. 1983. "A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs." *Mathematical programming* 27 (1):1-33.

Golub, Benjamin, and Matthew O Jackson. 2010. "Naive Learning in Social Networks and the Wisdom of Crowds." *American Economic Journal: Microeconomics* 2 (1):112-149.

Golub, Benjamin, and Matthew O. Jackson. 2012. "How Homophily Affects the Speed of Learning and Best-Response Dynamics." *The Quarterly Journal of Economics* 127 (3):1287-1338. doi: 10.1093/qje/qjs021.

Goodman, Leo A. 1961. "Snowball Sampling." *The Annals of Mathematical Statistics*:148-170.

Goodreau, Steven M, Mark S Handcock, David R Hunter, Carter T Butts, and Martina Morris. 2008. "A statnet Tutorial." *Journal of Statistical Software* 24 (9):1.

Grandy, Richard. 1973. "Reference, Meaning, and Belief." *The Journal of Philosophy* 70 (14):439-452.

Haenlein, Michael, and Barak Libai. 2013. "Targeting Revenue Leaders for a New Product." *Journal of Marketing* 77 (3):65-80.

Hammond, Kenneth R, and David A Summers. 1972. "Cognitive Control." *Psychological Review* 79 (1):58.

Handcock, Mark S, Adrian E Raftery, and Jeremy M Tantrum. 2007. "Model-based Clustering for Social Networks." *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 170 (2):301-354.

Hegselmann, Rainer, and Ulrich Krause. "Opinion Dynamics and Bounded Confidence Models, Analysis, and Simulation." *Journal of Artificial Societies and Social Simulation* 5, no. 3 (2002).

Hoeffding, Wassily. 1963. "Probability Inequalities for Sums of Bounded Random Variables." *Journal of the American Statistical Association* 58:13-30.

Hu, Yansong, and Christophe Van den Bulte. 2014. "Nonmonotonic Status Effects in New Product Adoption." *Marketing Science* 33 (4):509-533.

Hunter, David R, Mark S Handcock, Carter T Butts, Steven M Goodreau, and Martina Morris. 2008. "ergm: A Package to Fit, Simulate and Diagnose Exponential-family Models for Networks." *Journal of Statistical Software* 24 (3):nihpa54860.

Icard, Thomas, Eric Pacuit, and Yoav Shoham. 2010. "Joint Revision of Belief and Intention." *Proc. of the 12th International Conference on Knowledge Representation*.

Ihaka, Ross, and Robert Gentleman. 1996. "R: a Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics* 5 (3):299-314.

Iñiguez, Gerardo, János Kertész, Kimmo K. Kaski, and R. A. Barrio. 2009. "Opinion and Community Formation in Coevolving Networks." *Physical Review E* 80 (6). doi: 10.1103/PhysRevE.80.066119.

Iyengar, Raghuram, Christophe Van den Bulte, and Jae Young Lee. 2015. "Social Contagion in New Product Trial and Repeat." *Marketing Science* 34 (3):408-429.

Jackson, Matthew O. 2010. "An Overview of Social Networks and Economic applications." *The handbook of social economics* 1:511-85.

Jackson, Matthew O, and Brian W Rogers. 2007. "Meeting Strangers and Friends of Friends: How Random are Social Networks?" *The American Economic Review* 97 (3):890-915.

Jackson, Matthew O, and Leeat Yariv. 2007. "Diffusion of Behavior and Equilibrium Properties in Network Games." *The American Economic Review* 97 (2):92-98.

Jackson, Matthew O., and Dunia López-Pintado. 2013. "Diffusion and Contagion in Networks with Heterogeneous Agents and Homophily." *Network Science* 1 (01):49-67. doi: 10.1017/nws.2012.7.

Jadbabaie, Ali, Pooya Molavi, Alvaro Sandroni, and Alireza Tahbaz-Salehi. 2012. "Non-Bayesian Social Learning." *Games and Economic Behavior* 76 (1):210-225.

Jalili, Mahdi. 2013. "Social Power and Opinion Formation in Complex Networks." *Physica A: Statistical Mechanics and its Applications* 392 (4):959-966. doi: 10.1016/j.physa.2012.10.013.

Johnson, Donald B. 1977. "Efficient Algorithms for Shortest Paths in Sparse Networks." *Journal of the ACM (JACM)* 24:1-13.

Jöreskog, Karl G. 1967. "A General Approach to Confirmatory Maximum Likelihood Factor Analysis." *ETS Research Bulletin Serie*s 1967 (2):183-202.

Kacperski, Krzysztof. 1999. "Opinion Formation Model with Strong Leader and External Impact: a Mean Field Approach." *Physica A: Statistical Mechanics and its Applications* 269 (2):511-526.

Karrer, Brian, and Mark EJ Newman. 2011. "Stochastic Blockmodels and Community Structure in Networks." *Physical Review E* 83 (1):016107.

Kermack, William O, and Anderson G McKendrick. 1927. "A Contribution to the Mathematical Theory of Epidemics." *Proceedings of the Royal Society of London A: mathematical, physical and engineering sciences*.

Kleinberg, Jon. "The Small-world Phenomenon: An Algorithmic Perspective." *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pp. 163-170. ACM, 2000.

Klovdahl, Alden S, Zamakhsyari Dhofier, G Oddy, J O'Hara, S Stoutjesdijk, and A Whish. 1977. "Social Networks in an Urban Area: First Canberra Study1." *Journal of Sociology* 13 (2):169-172.

Klunder, G. A., and H. N. Post. 2006. "The Shortest Path Problem on Large-scale Real-road Networks." *Networks* 48:182-194.

Koenker, Roger, and Pin Ng. 2011. "SparseM: A Sparse Matrix Package for R." *CRAN Package Archive*.

Kourtellis, Nicolas, Tharaka Alahakoon, Ramanuja Simha, Adriana Iamnitchi, and Rahul Tripathi. 2013. "Identifying High Betweenness Centrality Nodes in Large Social Networks." *Social Network Analysis and Mining* 3 (4):899-914.

Kovalev, Anatolii Aleksandrovich, Vladimir Borisovich Kolmanovskii, and LE Shaikhet. 1998. "The Riccati Equations in the Stability of Stochastic Linear Systems with Delay." *Avtomatika i Telemekhanika* (10):35-54.

Lakhina, Anukool, John W. Byers, Mark Crovella, and Peng Xie. "Sampling Biases in IP Topology Measurements." *INFOCOM 2003 Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 1, pp. 332-341. IEEE, 2003.

Lanchier, Nicolas. 2012. "The Axelrod Model for the Dissemination of Culture Revisited." *The Annals of Applied Probability* 22 (2):860-880. doi: 10.1214/11-AAP790.

Lee, Keeheon, Shintae Kim, Chang Ouk Kim, and Taeho Park. 2013. "An Agent-based Competitive Product Diffusion Model for the Estimation and Sensitivity Analysis of Social Network Structure and Purchase Time Distribution." *Journal of Artificial Societies and Social Simulation* 16 (1):3.

Lee, Sang Hoon, Pan-Jun Kim, and Hawoong Jeong. 2006. "Statistical Properties of Sampled Networks." *Physical Review E* 73 (1):016102.

Leskovec, Jure, and Christos Faloutsos. "Sampling from Large Graphs." *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 631-636. ACM, 2006.

Leskovec, Jure, Jon Kleinberg, and Christos Faloutsos. "Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations." *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 177-187. ACM, 2005.

Lewenstein, Maciej, Andrzej Nowak, and Bibb Latané. 1992. "Statistical Mechanics of Social Impact." *Physical Review A* 45 (2):763.

Li, Lin, A. Scaglione, A. Swami, and Qing Zhao. 2013. "Consensus, Polarization and Clustering of Opinions in Social Networks." *IEEE Journal on Selected Areas in Communications* 31 (6):1072-1083. doi: 10.1109/JSAC.2013.130609.

Li, Qian, Lidia A. Braunstein, Huijuan Wang, Jia Shao, H. Eugene Stanley, and Shlomo Havlin. 2013. "Non-consensus Opinion Models on Complex Networks." *Journal of Statistical Physics* 151 (1-2):92-112. doi: 10.1007/s10955-012-0625-4.

Li, Qing, Sung Hyon Myaeng, and Byeong Man Kim. 2007. "A Probabilistic Music Recommender Considering User Opinions and Audio Features." *Information processing & management* 43 (2):473-487.

Libai, Barak, Eitan Muller, and Renana Peres. 2013. "Decomposing the Value of Word-of-mouth Seeding Programs: Acceleration Versus Expansion." *Journal of marketing research* 50 (2):161-176.

Liu, Bing. 2012. "Sentiment Analysis and Opinion Mining." *Synthesis Lectures on Human Language Technologies* 5 (1):1-167.

Liu, Jane, and Mike West. 2001. "Combined Parameter and State Estimation in Simulation-based Filtering." *In Sequential Monte Carlo methods in practice*, 197-223. Springer.

Liu, Xiaodong, Mo Li, Shanshan Li, Shaoliang Peng, Xiangke Liao, and Xiaopei Lu. 2014. "IMGPU: GPU-accelerated Influence Maximization in Large-scale Social Networks." *IEEE Transactions on Parallel and Distributed Systems* 25 (1):136-145.

Lopes, Hedibert F., and Carlos M. Carvalho. "Online Bayesian Learning in Dynamic Models: An Illustrative Introduction to Particle Methods." *Hierarchical Models and Markov Chain Monte Carlo* (2013).

Lu, Yingda, Kinshuk Jerath, and Param Vir Singh. 2013. "The Emergence of Opinion Leaders in a Networked Online community: A Dyadic Model with Time Dynamics and a Heuristic for Fast Estimation." *Management Science* 59 (8):1783-1799.

Lumsdaine, Andrew, Douglas Gregor, Bruce Hendrickson, and Jonathan Berry. 2007. "Challenges in Parallel Graph Processing." *Parallel Processing Letters* 17 (01):5-20.

Ma, Liye, Ramayya Krishnan, and Alan L Montgomery. 2014. "Latent Homophily or Social Influence? An Empirical Analysis of Purchase within a Social Network." *Management Science* 61 (2):454-473.

Ma, Zhenfeng, Zhiyong Yang, and Mehdi Mourali. 2014. "Consumer Adoption of New Products: Independent versus Interdependent Self-perspectives." *Journal of Marketing* 78 (2):101-117.

Madduri, Kamesh, David A. Bader, Jonathan W. Berry, and Joseph R. Crobak. "An Experimental Study of a Parallel Shortest Path Algorithm for Solving Large-scale Graph Instances." *Proceedings of the Meeting on Algorithm Engineering & Expermiments*, pp. 23-35. Society for Industrial and Applied Mathematics, 2007.

Mao, Guoyong, and Ning Zhang. 2013. "Analysis of Average Shortest-path Length of Scale-free Network." *Journal of Applied Mathematics* 2013.

Mao, Guoyong, and Ning Zhang. 2014. "A Multilevel Simplification Algorithm for Computing the Average Shortest-Path Length of Scale-Free Complex Network." *Journal of Applied Mathematics* 2014.

Marino, Marina, and Agnieszka Stawinoga. 2011. "Statistical Methods for Social Networks: a Focus on Parallel Computing." *MetodoloSki zvezki* 8 (1):57.

Martins, André C. R., Carlos de B. Pereira, and Renato Vicente. "An Opinion Dynamics Model for the Diffusion of Innovations." *Physica A: Statistical Mechanics and its Applications* 388 (15–16):3225-3232. doi: 10.1016/j.physa.2009.04.007.

McAuley, Julian J., and Jure Leskovec. "Learning to Discover Social Circles in Ego Networks." *In NIPS*, vol. 2012, pp. 548-56. 2012.

Miao, Qingliang, Qiudan Li, and Daniel Zeng. 2010. "Mining Fine Grained Opinions by using Probabilistic Models and Domain Knowledge." *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2010 IEEE/WIC/ACM International Conference on.

Miller, Grant, and A Mushfiq Mobarak. 2014. "Learning about New Technologies Through Social Networks: Experimental Evidence on Nontraditional Stoves in Bangladesh." *Marketing Science* 34 (4):480-499.

Molavi, P., A. Jadbabaie, K. R. Rad, and A. Tahbaz-Salehi. 2013. "Reaching Consensus With Increasing Information." *IEEE Journal of Selected Topics in Signal Processing* 7 (2):358-369. doi: 10.1109/JSTSP.2013.2246764.

Muthén, Bengt O. 2002. "Beyond SEM: General Latent Variable Modeling." *Behaviormetrika* 29 (1):81-117.

Nakata, Hiroyuki. 2003. "Modelling Exchange of Probabilistic Opinions." *Economic Theory* 21 (2-3):697-727.

Nasukawa, Tetsuya, and Jeonghee Yi. "Sentiment analysis: Capturing Favorability Using Natural Language Processing." *Proceedings of the 2nd international conference on Knowledge capture*, pp. 70-77. ACM, 2003.

Newman, M. E. J. 2003. "The Structure and Function of Complex Networks." *SIAM Review* 45 (2):167-256. doi: 10.1137/S003614450342480.

Newman, Mark EJ. 2000. "Models of the Small World." *Journal of Statistical Physics* 101:819-841.

Newman, Mark EJ. 2006. "Modularity and Community Structure in Networks." *Proceedings of the National Academy of Sciences* 103 (23):8577-8582.

Newman, Mark EJ. 2013. "Spectral Methods for Community Detection and Graph Partitioning." *Physical Review E* 88 (4):042822.

Newman, Mark EJ, and Duncan J Watts. 1999. "Renormalization Group Analysis of the Small-world Network Model." *Physics Letters A* 263 (4):341-346.

Nicol, David M., Jason Liu, Michael Liljenstam, and Guanhua Yan. "Simulation of Large Scale Networks Using SSF." In Simulation Conference, 2003. Proceedings of the 2003 Winter, vol. 1, pp. 650-657. IEEE, 2003.

Nowak, Andrzej, and Maciej Lewenstein. 1996. "Modeling Social Change with Cellular Automata." *In Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View*, edited by Rainer Hegselmann, Ulrich Mueller and Klaus G. Troitzsch, 249-285. Springer Netherlands.

Nowak, Andrzej, Jacek Szamrej, and Bibb Latané. 1990. "From Private Attitude to Public Opinion: A Dynamic Theory of Social Impact." *Psychological Review* 97 (3):362-376. doi: 10.1037/0033-295X.97.3.362.

Olfati-Saber, R. 2007. "Evolutionary Dynamics of Behavior in Social Networks." *46th IEEE* Conference on *Decision and Control*, , pp. 4051-4056. IEEE, 2007.

Olfati-Saber, R., J. A. Fax, and R. M. Murray. 2007. "Consensus and Cooperation in Networked Multi-Agent Systems." *Proceedings of the IEEE 95* (1):215-233. doi: 10.1109/JPROC.2006.887293.

Olfati-Saber, R., and R. M. Murray. 2004. "Consensus Problems in Networks of Agents with Switching Topology and Time-delays." *IEEE Transactions on Automatic Control* 49 (9):1520-1533. doi: 10.1109/TAC.2004.834113.

Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." *In LREc*, vol. 10, pp. 1320-1326. 2010.

Paternoster, Beatrice, and L Shaikhet. 2000. "About Stability of Nonlinear Stochastic Difference Equations." *Applied Mathematics Letters* 13 (5):27-32.

Pearl, Judea. 1988. "On Probability Intervals." *International Journal of Approximate Reasoning* 2 (3):211-216.

Pearl, Judea. 1990. "Reasoning with Belief Functions: an Analysis of Compatibility." *International Journal of Approximate Reasoning* 4 (5-6):363-389.

Potamias, Michalis, Francesco Bonchi, Carlos Castillo, and Aristides Gionis. "Fast Shortest Path Distance Estimation in Large Networks." *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 867-876. ACM, 2009.

Premaratne, Kamal, Manohar N Murthi, Jinsong Zhang, Matthias Scheutz, and Peter H Bauer. 2009. "A Dempster-Shafer Theoretic Conditional Approach to Evidence Updating for Fusion of Hard and Soft data." *In Information Fusion, 2009. FUSION'09. 12th International Conference on*, pp. 2122-2129. IEEE, 2009.

Racine, Jeffrey, Chirstopher F Parmeter, and Pang Du. 2009. "Constrained nonparametric kernel regression: Estimation and inference." In Working paper.

Reghbati, Eshrat, and Derek G Corneil. 1978. "Parallel Computations in Graph Theory." *SIAM Journal on Computing* 7 (2):230-237.

Ren, Wei, Randal W Beard, and Ella M Atkins. 2005. "A survey of consensus problems in multi-agent coordination." Proceedings of the 2005, American Control Conference, 2005.

Ren, Wei, Randal W. Beard, and Derek B. Kingston. "Multi-agent Kalman Consensus with Relative Uncertainty." *In Proceedings of the 2005, American Control Conference, 2005*., pp. 1865-1870. IEEE, 2005.

Resnick, Sidney I. *A Probability Path*. Springer Science & Business Media, 2013.

Rétvári, Gábor, József J. Bíró, and Tibor Cinkler. 2007. "On Shortest Path Representation." *IEEE/ACM Transactions on Networking (TON)* 15:1293-1306.

Riondato, Matteo, and Evgenios M. Kornaropoulos. "Fast Approximation of Betweenness Centrality Through Sampling." *In Proceedings of the 7th ACM international conference on Web search and data mining*, pp. 413-422. ACM, 2014.

Risselada, Hans, Peter C Verhoef, and Tammo HA Bijmolt. 2014. "Dynamic Effects of Social Influence and Direct Marketing on the Adoption of High-technology Products." *Journal of Marketing* 78 (2):52-68.

Robertson, Thomas S. *Innovative Behavior and Communication*. Holt McDougal, 1971.

Rogers, Carl R. "The Interpersonal Relationship: The Core of Guidance." *Harvard Educational Review* (1962).

Rosenberg, Dinah, Eilon Solan, and Nicolas Vieille. 2007. "Social Learning in One-Arm Bandit Problems." *Econometrica* 75 (6):1591-1611.

Rosenblatt, Murray. 1956. "A Central Limit Theorem and a Strong Mixing Condition." *Proceedings of the National Academy of Sciences* 42 (1):43-47.

Ryan, Bryce, and Neal C Gross. 1943. "The Diffusion of Hybrid Seed Corn in two Iowa Communities." *Rural Sociology* 8 (1):15.

Schoenberg, Ronald. 1997. "Constrained Maximum Likelihood." *Computational Economics* 10 (3):251-266.

Sentz, Kari, and Scott Ferson. *Combination of Evidence in Dempster-Shafer Theory.* Vol. 4015. Albuquerque: Sandia National Laboratories, 2002.

Shafer, Glenn. *A Mathematical Theory of Evidence*. Vol. 1. Princeton: Princeton University Press, 1976.

Shaikhet, L. 1996. "Stability of Stochastic Hereditary Systems with Markov Switching." *Theory of Stochastic Processes* 2 (18):180-184.

Shriver, Scott K, Harikesh S Nair, and Reto Hofstetter. 2013. "Social Ties and User-generated Content: Evidence from an Online Social Network." *Management Science* 59 (6):1425-1443.

Skrondal, Anders, and Sophia Rabe-Hesketh. *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Crc Press, 2004.

Sobkowicz, Pawel. 2009. "Modelling Opinion Formation with Physics Tools: Call for Closer Link with Reality." *Journal of Artificial Societies and Social Simulation* 12 (1):11.

Sommer, Christian. "Shortest-path Queries in Static Networks." *ACM Computing Surveys (CSUR)* 46, no. 4 (2014): 45.

Sousa, A. O. 2005. "Consensus Formation on a Triad Scale-free Network." *Physica A: Statistical Mechanics and its Applications* 348:701-710. doi: 10.1016/j.physa.2004.09.027.

Stadtfeld, Christoph. 2013. "NetSim: A Social Networks Simulation Tool in R." *project website), URL http://www. christoph-stadtfeld. com/netsim.*

Stephen, Andrew T, Peter Pal Zubcsek, and Jacob Goldenberg. 2016. "Lower Connectivity is Better: The Effects of Network Structure on Redundancy of Ideas and Customer Innovativeness in Interdependent Ideation Tasks." *Journal of Marketing Research* 53 (2):263-279.

Stocker, Rob, David Cornforth, and Terry RJ Bossomaier. "Network Structures and Agreement in Social Network Simulations." *Journal of Artificial Societies and Social Simulation* 5, no. 4 (2002).

Stone, Mervyn. 1961. "The Opinion Pool." *The Annals of Mathematical Statistics* 32 (4):1339-1342.

Suo, Shuguang, and Yu Chen. 2008. "The Dynamics of Public Opinion in Complex Networks." *Journal of Artificial Societies and Social Simulation* 11 (4):2.

Sznajd-Weron, Katarzyna, and Jozef Sznajd. 2000. "Opinion Evolution in Closed Community." *International Journal of Modern Physics C* 11 (06):1157-1165.

Tabachnick, B. G., and L. S. Fidell. *Using Multivariate Statistics*. HarperCollins Publishers, 1989.

Thorup, Mikkel. 1999. "Undirected Single-source Shortest Paths with Positive Integer Weights in Linear Time." *Journal of the ACM (JACM)* 46:362-394.

Toubia, Olivier, Jacob Goldenberg, and Rosanna Garcia. 2014. "Improving Penetration Forecasts Using Social Interactions Data." *Management Science* 60 (12):3049-3066.

Trusov, Michael, William Rand, and Yogesh V Joshi. 2013. "Improving Prelaunch Diffusion Forecasts: Using Synthetic Networks as Simulated Priors." *Journal of Marketing Research* 50 (6):675-690.

Vaswani, Namrata. "Kalman Filtered Compressed Sensing." In 2008 15th IEEE International Conference on Image Processing, pp. 893-896. IEEE, 2008.

Vaswani, Namrata. 2008b. "Particle Filtering for Large-dimensional State Spaces with Multimodal Observation Likelihoods." *IEEE Transactions on Signal Processing* 56 (10):4583-4597.

Verma, Anurag, Austin Buchanan, and Sergiy Butenko. 2015. "Solving the Maximum Clique and Vertex Coloring Problems on Very Large Sparse Networks." *INFORMS Journal on Computing* 27 (1):164-177.

Visser, Marco D, Sean M McMahon, Cory Merow, Philip M Dixon, Sydne Record, and Eelke Jongejans. 2015. "Speeding up Ecological and Evolutionary Computations in R; Essentials of High Performance Computing for Biologists." *PLoS Comput Biol* 11 (3):e1004140.

Wang, David Eppstein Joseph. 2006. "Fast Approximation of Centrality." *Graph Algorithms and Applications* 5 5:39.

Weisbuch, Gérard, Guillaume Deffuant, and Frédéric Amblard. 2005. "Persuasion Dynamics." *Physica A: Statistical Mechanics and its Applications* 353:555-575. doi: 10.1016/j.physa.2005.01.054.

Weisbuch, Gérard, Guillaume Deffuant, Frédéric Amblard, and Jean-Pierre Nadal. 2002. "Meet, Discuss, and Segregate!" *Complexity* 7 (3):55-63. doi: 10.1002/cplx.10031.

Wickramarathne, Thanuka L, Kamal Premaratne, Manohar N Murthi, and Nitesh V Chawla. 2014. "Convergence Analysis of Iterated Belief Revision in Complex Fusion Environments." *IEEE Journal of Selected Topics in Signal Processing* 8 (4):598-612.

Wickramarathne, TL, K Premaratne, MN Murthi, and M Scheutz. 2010. "A Dempster-Shafer theoretic evidence updating strategy for non-identical frames of discernment." *Proc. Workshop on the Theory of Belief Functions (WTBF'10), Brest, France*.

Wooldridge, Jeffrey M. 2005. "Simple Solutions to the Initial Conditions Problem in Dynamic, Nonlinear Panel Data Models with Unobserved Heterogeneity." *Journal of Applied Econometrics* 20 (1):39-54.

Wooldridge, Jeffrey M. 2010. *Econometric Analysis of Cross Section and Panel Data*. MIT press.

Yeom, Jae-Seung, Abhinav Bhatele, Keith Bisset, Eric Bohm, Abhishek Gupta, Laxmikant V Kale, Madhav Marathe, Dimitrios S Nikolopoulos, Martin Schulz, and Lukasz Wesolowski. 2014. "Overcoming the Scalability Challenges of Epidemic Simulations on Blue Waters." *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*.

Yildiz, Ercan, Daron Acemoglu, Asuman E. Ozdaglar, Amin Saberi, and Anna Scaglione. 2011. "Discrete Opinion Dynamics with Stubborn Agents." *Available at SSRN 1744113* (2011)

Zadeh, Lotfi A. 1986. "A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination." *AI Magazine* 7 (2):85.

Zhang, Zhongzhi, Lichao Chen, Shuigeng Zhou, Lujun Fang, Jihong Guan, and Tao Zou. 2008. "Analytical Solution of Average Path Length for Apollonian Networks." *Physical Review E* 77:017102.

Ziliaskopoulos, Athanasios, Dimitrios Kotzinos, and Hani S. Mahmassani. 1997. "Design and Implementation of Parallel Time-dependent Least Time Path Algorithms for Intelligent Transportation Systems Applications." *Transportation Research Part C: Emerging Technologies* 5:95-107.