AGENT-BASED DYNAMICS MODELS FOR OPINION SPREADING AND COMMUNITY DETECTION IN LARGE-SCALE SOCIAL NETWORKS

By

Jierui Xie

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Approved by the
Examining Committee:

Boleslaw K. Szymanski, Thesis Adviser

Gyorgy Korniss, Thesis Adviser

Mark K. Goldberg, Member

Mohammed J. Zaki, Member

Rensselaer Polytechnic Institute
Troy, New York

May 2012
(For Graduation August 2012)

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I would like to thank all the people that continuously support and encourage me during my study at RPI. Without their help, I would not be able to complete my thesis.

First of all, I would like to thank my advisor Prof. Szymanski, who has been giving me a lot of guidances. He is always there when I encounter a problem and need help, even during weekends and his travels. I am happy that I can work on interesting, exciting and challenging projects guided by him over these years. Prof. Szymanski not only has high academic standard, but also has a good sense of humor. I am very lucky to have him as my supervisor, colleague and friend.

I would like to thank my committee members. Prof. Korniss, from whom I have learned a lot, has been my co-advisor since I joined the social network project. He is very nice and always accessible. He inspires many ideas and models in this thesis. Both Prof. Goldberg and Zaki are experts in data mining and social network analysis. Their work inspires my study in the community detection. As my committee members, their have taught me a lot not only about knowledge but also how to be a researcher. I am grateful for all the guidances and help from my committee members. Without them, my thesis would not be as good as it is.

I am very lucky that I have many colleagues and friends, who give me help and good company. I would like to thank Sameet Sreenivasan, a very active and talented researcher. As a postdoc in my group, he has been giving me a lot of help in completing this thesis. I really enjoy working with him for the past three years. I would also like to thank Prof. Chjan Lim, Apirak Hoonlor, Zijian Wang, Andrea Asztalos, David Hunt, Pramesh Singh, Ferenc Molnar, Weituo Zhang, Jeffrey Emenheiser, Matthew Kirby, Sahin Cem Geyik, Andy Liu and Yousaf Shah.

Finally, I would like to thank my parents, my sister and my girlfriend. Even though they may be far away from me, I know that they are and would always be there supporting and encouraging me.

ABSTRACT

Human behavior is profoundly affected by the interaction of individuals and the social networks that link them together. In this thesis, two topics in the context of social network analysis (SNA) are studied. One is the opinion dynamics, and the other is the community structure discovery.

Opinion dynamics. In order to provide useful insights into understanding the evolution of opinions, ideologies or attitudes, this thesis explores a simple, abstract opinion dynamics model, called *binary agreement model*, where there are two *competing* opinions. The contribution of this thesis is to quantify the effect of *committed minorities* who hold unshakable opinions. In particular, such effect is investigated in two scenarios.

The first scenario is one committed group competing with the opposing uncommitted majority. The study shows how the prevailing majority opinion in a population can be rapidly reversed by a small fraction p of randomly distributed committed agents. When the committed fraction grows beyond a critical value p_c, there is a dramatic decrease in the time, T_c, taken for the entire population to adopt the committed opinion. For complete graphs, when $p < p_c$, $T_c \sim \exp(\alpha(p)N)$, where α is a function of p and N is the network size, while for $p > p_c$, $T_c \sim \ln N$. Simulation results for sparse networks (e.g., Erdős-Rényi (ER) random graphs, Barabasi-Albert (BA) scale-free networks and real-world social networks) show qualitatively similar behavior.

The second scenario is the more general case where two groups committed to distinct opinions A and B, and constituting fractions p_A and p_B, coexist. The study shows using mean-field theory that the phase diagram in parameter space (p_A, p_B), consists of two regions, one where two opinions coexist, and the remaining where one opinion always dominates the other. The scaling exponent associated with the exponential growth of switching times is found to be a function of the distance from the second-order transition point. Lastly, the nature (i.e., discontinuous and continuous) of transitions on sparse networks is explored by decomposing the system

into linear trajectories and deriving appropriate order parameters. The simulation results show that sparse networks are also characterized by the same qualitative phase diagram as the fully connected networks. As a comparison, the influence of global social media that serves as a kind of committed opinion is briefly discussed.

Community detection. Mining communities that allow multiple memberships is challenging especially in large-scale networks. This thesis presents a fast algorithm, called Speaker-listener Label Propagation Algorithm (SLPA), for *overlapping* community detection. SLPA is an extension of Label Propagation Algorithm (LPA). It spreads labels according to dynamic interaction rules and maintains label distributions in nodes' memories. Experiments in both synthetic and real-world networks show that SLPA has an excellent performance in identifying both node and community level overlapping structures. The performance is remarkably stable under different quality measures including normalized mutual information (NMI), Omega Index and F-score. SLPA can be applied to various structures, including weighted, unweighted, directed and undirected networks. With time complexity that scales linearly with the number of edges in the network, SLPA is successfully applied to networks with million nodes.

Detecting and tracking communities in a dynamic network where changes arrive as a stream is another challenging issue in real-world applications. Instead of computing communities on each snapshot independently, algorithms that *incrementally* update communities are very useful in the case of real time monitoring of huge data streams such as the Internet traffic or online social interactions. This thesis proposes *LabelRankT*, a decentralized online algorithm, to detect evolving communities in large-scale dynamic networks. LabelRankT by its own is based on a stabilized label propagation algorithm proposed in this thesis. It maintains the previous partitioning and dynamically updates only nodes involved in changes. As compared to other static algorithms including MCL and Infomap, LabelRankT achieves similar performance but with lower computational cost. Furthermore, it significantly outperforms and is more than 100 times faster than dynamic detection algorithms such as facetNet and iLCD. Importantly, LabelRankT is highly parallelizable allowing the computation to be distributed to each individual node. Such property will be

particularly useful for applications like wireless sensor networks and mobile ad hoc networks, where each node in the network corresponds to a physical platform.

CHAPTER 1
INTRODUCTION

This chapter introduces backgrounds, motivations, model selections, basic definitions and related work for both opinion dynamics models and community detection.

1.1 Social Dynamics, Binary Agreement Model and Commitment

1.1.1 Consensus with Committed Agents

Since the seminal work of Gabriel Tarde [1] in the late 1800s, the shaping of public opinion through interpersonal influence and conformity has been a subject of significant interest in sociology. This topic is especially relevant today due to the proliferation of online social media where individuals can influence and be influenced by their numerous and geographically spread contacts. The dynamics of social influence has been heavily studied in sociological, physics, and computer science literature [2–6].

In this thesis, we first study how a population moves towards *consensus* in the adoption of *competing* ideologies, traditions, and attitudes [7, 8]. Examples of disputable opinions attempting to convey to the public include the change of climate, the need of the flu vaccine, the advantage of hybrid cars, etc. In the sociological context, work on diffusion of innovations [9] has emphasized how individuals adopt new *states* in behavior, opinion or consumption through the influence of their neighbors. Commonly used models for this process include the threshold model [10] and the Bass diffusion model [11]. A key feature in both these models is that once an individual adopts a new state, his state remains unchanged at all subsequent times. While appropriate for modeling the diffusion of innovation where investment in a new idea comes at a cost (e.g., buying a new computer or car), these models are less suited to studying the dynamics of competing opinions where switching one's state has little overhead (e.g., adopting a new public opinion or following other topics on

Twitter[1]).

Here we focus on the case where the cost of adoption is low [12], or where changes in state are not deliberate or calculated, but unconscious [13]. We study a two-opinion variant [14] of the Naming Game (NG) [15–19] that we refer to as the *binary agreement model*. In the model, two opinions A and B consist of three states A, B and AB. The evolution of the system in this model takes place through the usual NG dynamics, wherein at each simulation time step, a randomly chosen *speaker* voices a random opinion from his list to a randomly chosen neighbor, designated the *listener*. If the listener has the spoken opinion in his list, both speaker and listener retain only that opinion, else the listener adds the spoken opinion to his list.

An important difference between the binary agreement model and the predominantly used opinion dynamics models [3,5,20–22], such as the voter model [23], is that an agent is allowed to possess two opinions simultaneously in the former. Unlike the simple voter model where a node adopts one of the two states (i.e., 1/0), and fluctuation is the driving force of evolution, with an intermediate state (i.e., AB), the binary agreement model allows us to study the *tipping points* and coarse-graining phenomena observed in empirical data. Another merit of the binary agreement dynamics in modeling social opinion change compared with Epidemic models (e.g., SIS and its variants [24]) and epidemic-like models of social "contagion" (examples in [25]) is that the latter suffer from the drawback that the rules governing the conversion of a node from a given state to the other are not symmetric for the two states or opinions. In contrast, in the binary agreement model, both singular opinion states are treated symmetrically in their susceptibility to change.

In particular, this thesis studies the evolution of opinions in the presence of *committed* agents [26], defined as nodes that can influence other nodes to alter their states through the usual prescribed rules, but which themselves are immune to influence. We start from an *initial state* where all agents adopt a given opinion B, except for a finite fraction p of the total number of agents who are *committed agents* and have state A. Since p is small, it defines the *minorities*. In the presence of committed agents adopting state A, the only absorbing fixed point of the system

[1] http://twitter.com.

is the consensus state where all influenceable nodes adopt opinion A - the opinion of the committed agents.

The questions of interest are: *what is the condition under which an inflexible set of minority opinion holders can win over the rest of the population and how does the consensus time vary with the size of the committed fraction?*

The effect of having un-influencable agents has been considered to some extent in prior studies. Biswas et al. [27] considered for two-state opinion dynamics models in one dimension, the case where some individuals are "rigid" in both segments of the population, and studied the time evolution of the magnetization and the fraction of domain walls in the system. Mobilia et al. [28] considered the case of the voter model with some fraction of spins representing "zealots" who never change their states, and studied the magnetization distribution of the system on the complete graph, and in one and two dimensions. Similarly to [28], Yildiz et al. [29] studied the properties of steady state opinion distribution for the voter model with *stubborn* agents, but additionally considered the optimal placement of stubborn agents so as to maximally affect the steady-state opinion on the network. Again, unlike in the model studied here, in the voter model, no transitions in steady-state magnetization are observed as the committed fraction pair values are smoothly varied.

The study in this thesis differs from the above models not only in the particular model of opinion dynamics considered, but also in its explicit consideration of different network topologies and of finite sized networks, specifically in its derivation of how consensus times scale with network size for the case of the complete graph. Furthermore, the above mentioned studies do not explicitly consider the initial state that we care about - one where the entire minority set is un-influencable. A notable exception is the study by Galam and Jacobs [30, 31] in which the authors considered the case of "inflexibles" in a two-state model of opinion dynamics with opinion updates obeying a majority rule. While that study provides several useful insights into understanding the effect of committed minorities, its analysis is restricted to the mean-field case, and has no explicit consideration of consensus times for finite systems.

1.1.2 Presence of Competing Committed Groups

Public opinion on an issue is often shaped by the actions of groups that rigidly advocate competing points of view. The most evident example of such a process occurs during elections when multiple parties campaign to influence and win over the majority of the voters. Specifically in the context of rural campaigns, there is evidence that interpersonal channels constitute the dominant pathways for effecting individual behavior change, even when direct external influence is present [32]. Motivated by these observations, we generalize the above model to incorporate the presence of two groups within the network that are committed to distinct, competing opinions on an issue. Within the limits of the proposed model, similar questions are addressed including: *what should be the minimal fractional size of a competing committed group in order to effect a fast reversal in the majority opinion, and is there any property different from the previous single committed opinion case.*

1.2 Community Detection and Label Propagation

Modular structure is considered to be a significant property of real-world social networks as it often accounts for the functionality of the system. Despite the ambiguity in the definition of *community*, numerous techniques have been developed for community detection. Random walks, spectral clustering, modularity maximization, differential equations, and statistical mechanics have been used previously. Much of the focus has been on identifying *disjoint* and *static* communities [33–36].

1.2.1 Overlapping Community Detection

In reality, it is well understood that people in a social network are naturally characterized by *multiple* community *memberships* (i.e., the existence of *overlap* [37]). For example, a person usually has connections to several social groups like family, friends, and colleagues; a researcher may be active in several areas.

There has been growing interest in overlapping community detection algorithms since 2005 [38, 39]. We have categorized existing algorithms into five major classes in the survey paper [40], summarized as follows: (I) clique percolation based algorithms [38,41,42], which are based on the assumption that a community consists

of fully connected subgraphs and detecting overlapping communities by searching for *adjacent* cliques; (II) line graph and link partitioning based algorithms [43–48], whose idea is to partition links instead of nodes to allow multiple memberships; (III) local expansion and optimization based algorithms [39, 49–61] are based on growing a *natural* community [51] by maximizing a benefit function that characterizes the quality of a densely connected group of nodes; (IV) fuzzy algorithms [62–71], which construct a soft membership vector or belonging factor [72] explicitly; (V) dynamical algorithms [26, 72–79], which are based on principles from statistical mechanism or simulations.

On one hand, the extension from disjoint to overlapping increases the difficulty of detection. On the other hand, the rapid emergence of large-scale social networks[2] today brings more challenges for social network analysis. However, most of these algorithms are not efficient enough for very large networks (e.g., up to million nodes), such as CFinder [38], EAGLE [58], NMF [70], MOSES [68], UEOC [54], OSLOM [53] and so on (the worst-case running times of commonly used algorithms are shown in Table 4.1.), while some are ineffective, such as Link [43] and iLCD [57]. We seek both effective and efficient algorithms for large-scale networks in Chapter 4.

1.2.2 Dynamic Community Detection

With the rapid emergence of large-scale online social networks, there is a high demand for an efficient community detection algorithm that is able to handle a large amount of new data on a daily basis. For example, with 900 millions users on Facebook[3] as of 2012, more than 250 million photos are uploaded to Facebook every day. However, most of the community detection algorithms rely on a *static* and *global* view of the network. Structure information other than local connections is vital for these algorithms, and the temporal correlation between different snapshots is also ignored. Such algorithms are less suitable for a dynamically evolving network, especially in the case where new data come in continuously.

[2]According to the recent eBizMBA Rank*, the top four most popular social networking sites are Facebook with 900 million active users since 2004, Twitter with 300 million active users since 2006, LinkedIn 116 million active users since 2002 and Myspace with 30 active million users since 2003.

[3]http://www.facebook.com.

More importantly, an algorithm that can take advantage of the cutting-edge cloud computing (e.g., EC2 [21]) or the ubiquitous mobile computing [33] (e.g., a human-centralized mobile phone network or an ad hoc wireless sensor network) would be even more valuable for future applications. This essentially requires that a detection algorithm is parallelizable or distributed. We seek solutions that meet these two needs, online and parallelizable detection, in Chapter 5.

1.2.3 Label Propagation

Multi-state spin models [26, 76], in which a spin is assigned to each node, can be applied to community detection. One of such models is q-state Potts model [76, 77, 80], where q is the number of states that a spin may take, indicating the maximum number of communities. The community detection problem is equivalent to the problem of minimizing the Hamiltonian of the model. In the ground states (i.e., local minima of the Hamiltonian), the set of nodes with the same spin state form a community.

The work relevant to our study is the Label Propagation Algorithm (LPA) [81–83], which belongs to the family of agent-based community detection algorithms. LPA is shown to be equivalent to a Potts model in [82]. LPA can also be viewed as a simple opinion spreading model but with many competing opinions. During the iterative process, each node adopts the *label* or opinion (i.e., an artificial unique id) in agreement with the majority of its neighbors. At the end of the algorithm, connected nodes with the same label form a community. This algorithm provides desirable qualities such as easy implementation and fast execution.

The first attempt to extend LPA for overlapping community detection was made in [72] by allowing a node to have multiple labels. However, it is shown to be unstable for some networks by producing a number of small sized communities. This problem is alleviated with new strategies proposed in Chapter 4.

Since LPA-based community detection algorithms require only *local* information, they can potentially serve as good candidates for a parallel or distributed detection. However, due to the use of random tie breaking strategy, label propagation based algorithms are nondeterministic in nature. They produce different partitions

in different runs. Such randomness becomes an undesired property when tracking the evolution of communities in a dynamic network. This problem is addressed by new operators proposed in Chapter 5.

1.3 Contributions and Organization

The contributions of this thesis are two-fold. First, this thesis quantifies the effect of committed agents in the binary opinion dynamics model in two scenarios. Chapter 2 studies one committed group (minorities) competing with the opposing majority [84]. Chapter 3 generalizes the model to the case where two committed groups with distinct competing opinions coexist [85]. The effect of global social media which serves as a kind of committed opinion is also discussed. For each of these scenarios, both the transitions (i.e., tipping points) and the time scale of the evolution in various network structures are identified.

Second, this thesis extends the LPA algorithm for effective and efficient community detection. Chapter 4 proposes SLPA [73, 86], a linear time algorithm for identifying overlapping structure by introducing a label distribution in a node's memory, which provides better performance and can be applied to various structures, including weighted, unweighted, directed and undirected networks. Chapter 5 proposes LabelRankT for detecting temporal evolution of communities. LabelRankT is based on a *stabilized* label propagation algorithm proposed in this thesis. It maintains the previous partitioning and dynamically updates only nodes involved in changes. As compared to other static algorithms including MCL and Infomap, LabelRankT achieves similar performance but with much lower computational cost. Furthermore, it significantly outperforms and is more than 100 times faster than dynamic detection algorithms such as facetNet and iLCD.

The rest of thesis is mainly composed of materials from the following papers [84], [85], [40, 73, 86], [87]. I was the main author of these papers to which I contributed a majority of the technical work with the supervision and help of the other co-authors.

CHAPTER 2
INFLUENCE OF COMMITTED MINORITIES

2.1 The Binary Agreement Model

In a dynamical opinion model, the interactions between individuals are the key. In this particular model, there are only two distinct opinions denoted as A and B. Each node (i.e., agent) stores opinions in his list, and takes one of the three states A, B or AB. AB is the mixed state that both opinions coexist. The dynamic rules at each simulation step are as follows* :

- Randomly select a node from the system as the *speaker*.
- Randomly select one of the speaker's neighbors as the *listener*.
- The speaker randomly selects an opinion from his list and sends it to the listener.
- If the listener already has this opinion, both the speaker and listener retain only this opinion; Otherwise, the listener adds the opinion to his list.

All the possible interactions are shown in Table 2.1. The order of selecting speakers and listeners is known to influence the dynamics, and we stick to choosing the speaker first, followed by the listener. The *unit* time consists of N speaker-listener interactions (steps), where N is the total number of nodes.

We start from an initial state where all nodes adopt a given opinion B, except for a finite *committed fraction* p of the total number of agents who are *committed agents* and have state A. We designate the densities of *uncommitted* nodes in states A, B as n_A, n_B respectively. Consequently, the density of nodes in the mixed state AB is $n_{AB} = 1 - p - n_A - n_B$.

*Portions of this chapter previously appeared as: J. Xie, S. Sreenivasan, G. Korniss, W. Zhang, C. Lim, and B. K. Szymanski, Social consensus through the influence of committed minorities, Phys. Rev. E, vol. 84, no. 1, p. 011130, Jul. 2011 [84].

Table 2.1: Possible interactions in the binary agreement model. The Speaker is on the left of the arrow, and the listener is on the right. The opinion voiced by the speaker is shown above the arrow.

Before interaction	After interaction	Before interaction	After interaction
A $\xrightarrow{A}$ A	A - A	AB $\xrightarrow{A}$ A	A - A
A $\xrightarrow{A}$ B	A - AB	AB $\xrightarrow{A}$ B	AB - AB
A $\xrightarrow{A}$ AB	A - A	AB $\xrightarrow{A}$ AB	A - A
B $\xrightarrow{B}$ A	B - AB	AB $\xrightarrow{B}$ A	AB - AB
B $\xrightarrow{B}$ B	B - B	AB $\xrightarrow{B}$ B	B - B
B $\xrightarrow{B}$ AB	B - B	AB $\xrightarrow{B}$ AB	B - B

2.2 The Effect on Complete Graphs

2.2.1 Infinite Network Size Limit: the Tipping Point

We start along similar lines as [30] by considering the case where the social network connecting agents is a complete graph with the size of the network $N \to \infty$. Neglecting fluctuations and correlations between nodes, one can write the following rate equations for the evolution of densities:

$$\begin{aligned} \frac{dn_A}{dt} &= -n_A n_B + n_{AB}^2 + n_{AB} n_A + \frac{3}{2} p n_{AB} \\ \frac{dn_B}{dt} &= -n_A n_B + n_{AB}^2 + n_{AB} n_B - p n_B. \end{aligned} \tag{2.1}$$

The terms in these equations are obtained by considering all interactions which increase (decrease) the density of agents in a particular state and computing the probability of that interaction occurring. Table 2.1 lists all possible interactions. As an example, the probability of the interaction listed in row eight is equal to the probability that a node in state AB is chosen as speaker and a node in state B is chosen as listener ($n_{AB} n_A$) times the probability that the speaker voices opinion A ($\frac{1}{2}$).

The fixed-point and stability analyses (see Appendix A.1) of these *mean-field* equations show that for any value of p, the consensus state in the committed opinion ($n_A = 1 - p$, $n_B = 0$) is a stable fixed point of the mean-field dynamics. However, below $p = p_c = \frac{5}{2} - \frac{3}{2}\left(\sqrt[3]{5+\sqrt{24}} - 1\right)^2 - \frac{3}{2}\left(\sqrt[3]{5-\sqrt{24}} - 1\right)^2 \approx 0.09789$, two

additional fixed points appear: one of these is an unstable fixed point (saddle point), whereas the second is stable and represents an *active* steady state where n_A, n_B and n_{AB} are all non-zero (except in the trivial case where $p = 0$) . Figure 2.1(a) shows (asterisks) the steady state density of nodes in state B obtained by numerically integrating the mean-field equations at different values of the committed fraction p and with initial condition $n_A = 0, n_B = 1 - p$. As p is increased, the stable density of B nodes n_B abruptly jumps from ≈ 0.6504 to zero at the critical committed fraction p_c. A similar abrupt jump also occurs for the stable density of A nodes from a value very close to zero below p_c, to a value of 1, indicating consensus in the A state (not shown). In the study of phase transitions, an "order parameter" is a suitable quantity changing (either continuously or discontinuously) from zero to a non zero-value at the critical point. Following this convention, we use n_B, the density of uncommitted nodes in state B, as the order parameter appropriate for our case, characterizing the transition from an active steady state to the absorbing consensus state.

In practice, for a complete graph of any finite size, consensus is always reached. However, we can still probe how the system evolves, conditioned on the system not having reached consensus. Figure 2.1(a) shows the results of simulating the binary agreement model on a complete graph for different system sizes (solid lines). For $p < p_c$, in each realization of agreement dynamics, neglecting the initial transient, the density of nodes in state B, n_B, fluctuates around a non-zero steady state value, until a large fluctuation causes the system to escape from this active steady state to the consensus state. Figure 2.1(a) shows these steady state values of n_B conditioned on survival, for several values of p. As expected, agreement of simulation results with the mean-field curve improves with increasing system size, since Eq. 2.1 represents the true evolution of the system in the asymptotic large network-size limit. Accordingly the critical value of the committed fraction obtained from the mean-field equations is designated as $p_c(\infty)$, although, for brevity, we refer to it simply as p_c throughout this chapter.

The existence of the transition as p is varied and when the initial condition for densities is $(n_A = 0, n_B = 1-p)$ can be further understood by observing the motion

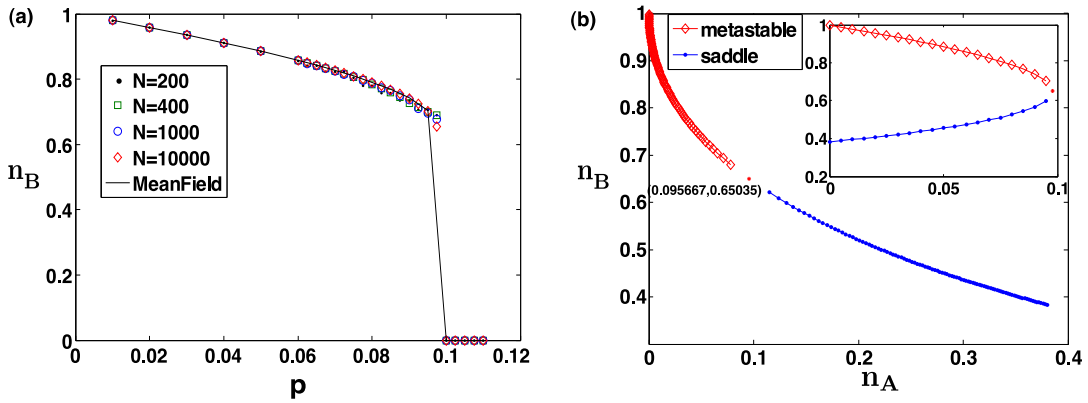

Figure 2.1: (a) The steady state density n_B of nodes in state B as a function of committed fraction p for complete graphs of different sizes, conditioned on survival of the system. Simulation results are from 100 realizations of the binary agreement dynamics. (b) Movement of the stable fixed point (diamonds) and the saddle point (filled circles) in phase space as a function of committed fraction p (see text). The point at which they meet (coordinates shown) is indicated by the asterisk. The location of these points in phase space was obtained through fixed point analysis of the mean field equations Eq. 2.1 (see Appendix A). The inset shows the density of nodes in state B at the stable (red) and unstable(blue) fixed points as p is varied.

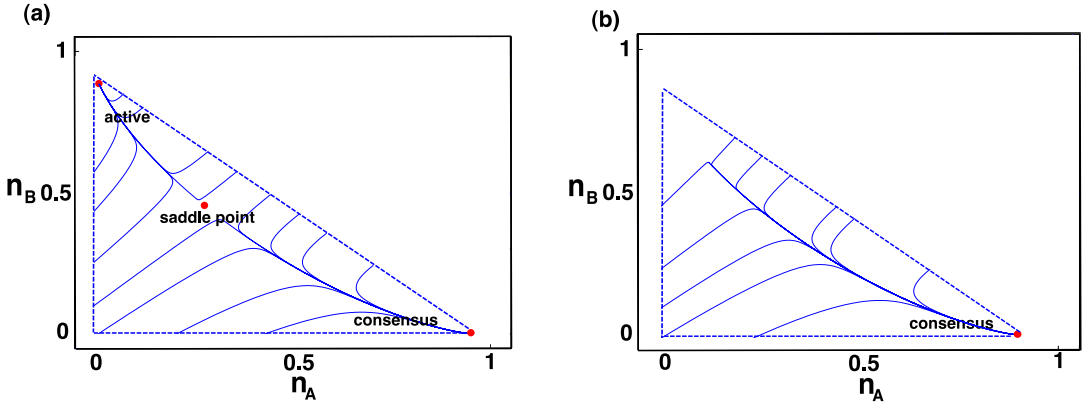

Figure 2.2: Trajectories (obtained from integration of the mean-field equations Eq. 2.1) in the phase-plane show the nature of flows from different regions of the phase-plane into existing fixed points for (a) $p = 0.05$ ($< p_c$) and (b) $p = 0.1$ ($> p_c$).

of the fixed points in phase space. Figure 2.1(b) shows how the stable fixed point and the unstable fixed point move in phase space as p is varied from 0 to p_c. The active steady state moves downward and right while the saddle point moves upwards and left. At the critical value p_c the two meet and the only remaining stable fixed point is the consensus fixed point. A similar observation was made in the model studied in [30]. The fact that the value of n_B converges to $\cong 0.65$ and does not smoothly approach zero as the stable fixed point and the saddle point approach each other, explains the origin of the *first-order* nature of the phase transition. Figure 2.2 shows the representative trajectories obtained by integrating the mean-field equations for the cases where $p = 0.05$ ($< p_c$) and $p = 0.1$ ($> p_c$).

2.2.2 Finite Network Size: the Consensus Times

Even though consensus is always reached for finite N, limits on computation time prohibit the investigation of the consensus time, T_c, for values of p below or very close to p_c. We therefore adopt a semi-analytical approach prescribed in [88] that allows us to estimate the consensus times for different N for an appreciable range of p including values below p_c. We start with the master equation which describes the evolution of the probability that the network of size N has n (m) uncommitted nodes in state A (B). We denote by c, the number of committed nodes, and by $l(= N - n - m - c)$, the number of uncommitted nodes in state AB.

$$\begin{aligned}
\frac{dp_{nm}}{dt}\frac{1}{N} &= \frac{1}{N^2}\Big[- p_{nm}\,(2ln + \frac{3}{2}lc + 2nm + l(l-1) + 2lm + mc) \\
&+ p_{n-1,m}\frac{3(l+1)(n-1+c)}{2} + p_{n+1,m}\frac{(n+1)(2m+l-1)}{2} \\
&+ p_{n-2,m}\frac{(l+2)(l+1)}{2} + p_{n,m-1}\frac{3(l+1)(m-1)}{2} \\
&+ p_{n,m+1}\frac{(m+1)(2n+2c+l-1)}{2} + p_{n,m-2}\frac{(l+2)(l+1)}{2}\Big]. \qquad (2.2)
\end{aligned}$$

The factor of $1/N$ in the LHS comes from the fact that a transition between states takes place in an interval of time $1/N$. The transition rates in each term are the product of two densities, which is responsible for the overall factor of $1/N^2$ in the RHS. The probabilities are defined over all allowed states of the system (i.e., $0 \leq$

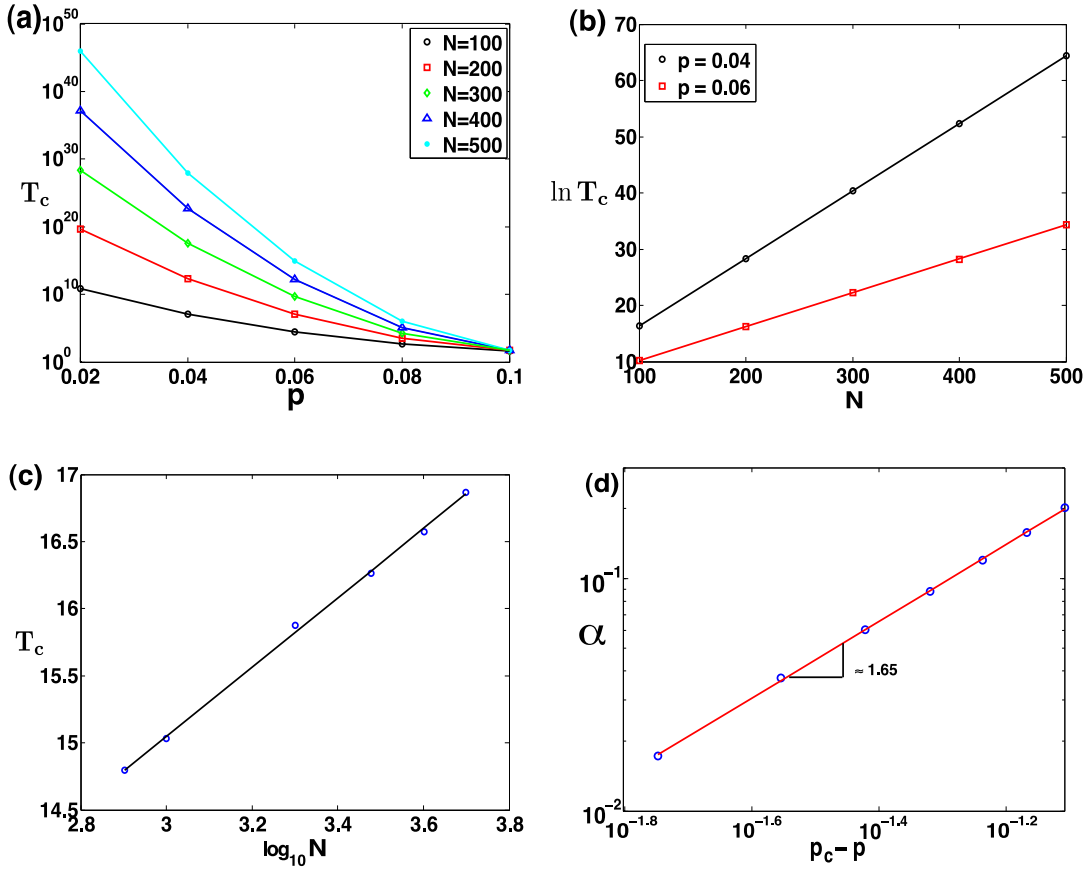

Figure 2.3: (a) Mean consensus time T_c for $p < p_c$ obtained by using the QS approximation. (b) Exponential scaling of T_c with N, for $p < p_c$; mean consensus times (circles, squares) are obtained using the QS approximation. The lines are guides to the eye. (c) Logarithmic scaling of T_c with N for $p = 0.3 > p_c$; mean consensus times are obtained from simulations. The line shows the best linear fit to the data. (d) The rate, $\alpha(p)$, of exponential growth of the consensus time with N as a function of $p - p_c$ (see text). Circles show the values of $\alpha(p)$ obtained for $p = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8$ by considering the scaling of T_c with N, for these values of p. The straight line shows a linear fit to the data plotted on a log-log scale.

$n \leq N-c$, and $0 \leq m \leq N-c-n$ for given n) and the allowed transitions from any point $\{n, m\}$ in the interior of this state-space are $\{n, m\} \to \{n, m \pm 1\}, \{n, m\} \to \{n \pm 1, m\}, \{n, m\} \to \{n, m+2\}, \{n, m\} \to \{n+2, m\}$.

We know from the mean field equations that in the asymptotic limit, and below a critical fraction of committed agents, there exists a stable fixed point. For finite

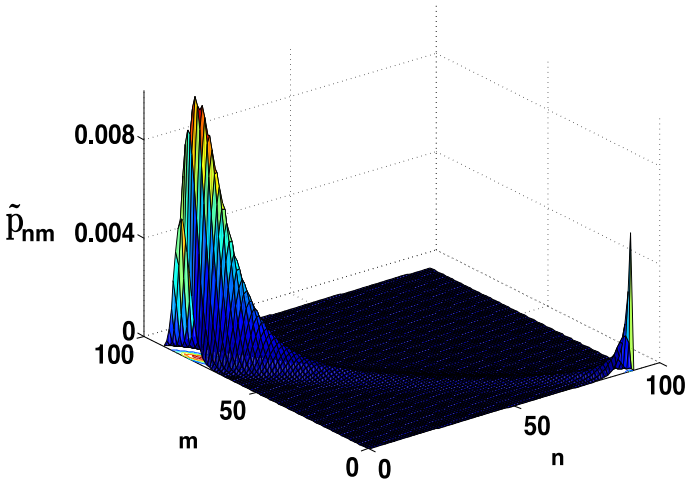

Figure 2.4: The quasistationary distribution $\tilde{p}_{nm}$ for $p = 0.09$ and $N = 100$.

stochastic systems, escape from this fixed point is always possible, and therefore it is termed *metastable*. For a finite system, the probability of having escaped to the metastable fixed point as a function of time is $P_e(t) = 1 - P_s(t)$ where $P_s(t)$ is the survival probability. The surviving fraction is constrained to be in the allowed region of n, m quadrant excluding the true fixed point $\{N - c, 0\}$. If the number of committed agents is far lower than $p_c\ N$ we expect that this surviving fraction will occupy configurations around the metastable fixed point, and the occupation probabilities $p_{n,m}$ will be peaked around the metastable fixed point. In systems which exhibit such long lived metastable states in addition to an absorbing fixed point, applying a quasistationary (QS) approximation has been found to be useful in computing quantities of interest [88–90]. This approximation assumes that after a short transient, the occupation probability, conditioned on survival, of allowed states excluding the consensus state, is stationary. Following this approximation, the distribution of occupation probabilities conditioned on survival can be written as, $\tilde{p}_{nm} = p_{nm}(t)/P_s(t)$ [88] and using this form in the master equation (Eq. 2.2),

we get:

$$\begin{aligned}\frac{dP_s(t)}{dt}\tilde{p}_{nm} &= -\frac{P_s(t)}{N}\Big[\tilde{p}_{nm}\,(2ln+\frac{3}{2}lc+2nm+l(l-1)+2lm+mc)\\ &- \tilde{p}_{n-1,m}\frac{3(l+1)(n-1+c)}{2}-\tilde{p}_{n+1,m}\frac{(n+1)(2m+l-1)}{2}\\ &- \tilde{p}_{n-2,m}\frac{(l+2)(l+1)}{2}-\tilde{p}_{n,m-1}\frac{3(l+1)(m-1)}{2}\\ &- \tilde{p}_{n,m+1}\frac{(m+1)(2n+2c+l-1)}{2}-\tilde{p}_{n,m-2}\frac{(l+2)(l+1)}{2}\Big]. \quad (2.3)\end{aligned}$$

Considering transitions from states $\{N-c-1,0\}$ and $\{N-c-2,0\}$ to the absorbing state $\{N-c,0\}$, we obtain the decay rate of the survival probability $dP_s(t)/dt$:

$$\frac{dP_s(t)}{dt} = -P_s(t)\left[\tilde{p}_{N-c-1,0}\left(\frac{3(N-1)}{2N}\right)+\tilde{p}_{N-c-2,0}\frac{2}{N}\right]. \quad (2.4)$$

Substituting Eq. 2.4 into Eq. 2.3, we finally obtain a condition that the occupation probabilities conditioned on survival must satisfy [91] :

$$\tilde{p}_{nm} = \frac{\tilde{Q}_{nm}}{W_{nm}-\tilde{Q}_0}, \quad (2.5)$$

where $\tilde{Q}_{nm} = Q_{nm}(t)/P_s(t)$ is obtained through explicit consideration of the terms in the master equation:

$$\begin{aligned}\tilde{Q}_{nm} &= \tilde{p}_{n-1,m}\frac{3(l+1)(n-1+c)}{2}+\tilde{p}_{n+1,m}\frac{(n+1)(2m+l-1)}{2}\\ &+ \tilde{p}_{n-2,m}\frac{(l+2)^2}{2}+\tilde{p}_{n,m-1}\frac{3(l+1)(m-1)}{2}\\ &+ \tilde{p}_{n,m-2}\frac{(l+2)^2}{2}+\tilde{p}_{n,m+1}\frac{(m+1)(2n+2c+l-1)}{2} \quad (2.6)\end{aligned}$$

and $\tilde{Q}_0 = \left[\tilde{p}_{N-c-1,0}\left(\frac{3(N-1)}{2}\right)+2\,\tilde{p}_{N-c-2,0}\right]$ is the term arising from the decay of the survival probability (Eq. 2.4). W_{nm} is the coefficient of p_{nm} ($\tilde{p}_{nm}$) within the brackets on the right hand side of Eq. 2.2 (Eq. 2.3) and is equal to the transition rate out of state $\{n,m\}$ times N^2.

Equation 2.4 indicates that the survival probability decays exponentially with a rate $\lambda = \tilde{Q}_0/N$. Since the mean lifetime of an exponentially decaying process is

the inverse of the decay rate, it follows that the mean consensus time (neglecting the short transient before the QS distribution is attained) is

$$T_c \approxeq \frac{1}{\lambda} = 1/\left[\tilde{p}_{N-c-1,0}\left(\frac{3(N-1)}{2N}\right) + \tilde{p}_{N-c-2,0}\frac{2}{N}\right]. \tag{2.7}$$

Thus, knowledge of $\tilde{p}_{nm}$'s (in particular, $\tilde{p}_{N-c-1,0}$ and $\tilde{p}_{N-c-2,0}$) would allow us to calculate T_c through Eq. 2.7. In order to obtain $\tilde{p}_{nm}$ (for all $0 \leq n, m$), we adopt the iterative procedure proposed in [91]. Following this procedure, we start with an arbitrary initial distribution $\tilde{p}^0_{nm}$, and obtain a new distribution using: $\tilde{p}^{i+1}_{nm} = \alpha\ \tilde{p}^i_{nm} + (1-\alpha)\ \frac{\tilde{Q}^i_{nm}}{W^i_{nm}-\tilde{Q}^i_0}$, where $0 \leq \alpha \leq 1$ is an arbitrary parameter, and $\tilde{Q}^i_{nm}, W^i_{nm}$ and $\tilde{Q}^i_0$ are all obtained using the probability distribution at the current iteration, $\tilde{p}^i_{nm}$. With sufficient number of iterations, this procedure is expected to converge to a distribution that satisfies Eq. 2.5 and which is thus, the desired QS distribution. In our case, we obtained acceptable convergence with a choice of $\alpha = 0.5$ and 30000 iterations.

Following the above method, we obtain the QS distribution, and consequently the mean consensus times T_c for different values of committed fraction p and system size N. Figure 2.3(a) shows how the consensus time grows as p is decreased beyond the asymptotic critical point p_c for finite N. For $p < p_c$, the growth of T_c is exponential in N (Fig. 2.3(b), consistent with what is known regarding escape times from metastable states. For $p > p_c$, the QS approximation does not reliably provide information on mean consensus times, since consensus times themselves are small and comparable to transient times required to establish a QS state. However, simulation results show that above p_c the scaling of mean consensus time with N is logarithmic (Fig. 2.3(c)). A snapshot of the QS distribution (Fig. 2.4) near p_c ($p = 0.09$) for a system of size $N = 100$ shows clearly the bimodal nature of the distribution, with the two modes centered around the stable fixed point, and the consensus fixed point.

The precise dependence of consensus times on p can also be obtained for $p < p_c$ by considering the rate of exponential growth of T_c with N. In other words, assuming $T_c \sim \exp(\alpha(p)N)$, we can obtain $\alpha(p)$ as a function of p. Figure 2.3(d) shows that

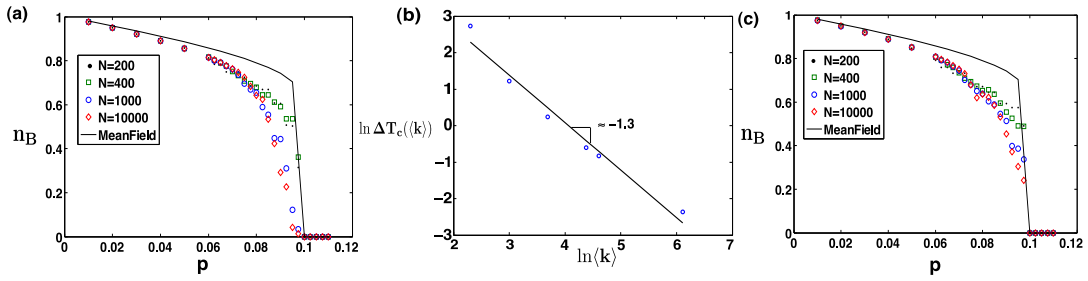

Figure 2.5: (a) The steady state density n_B of nodes in state B as a function of committed fraction p for Erdős-Rényi graphs of different sizes with $\langle k \rangle = 10$, conditioned on survival of the system. Symbols show mean values of n_B obtained from 100 simulations of different system sizes; the black line shows mean-field consensus times obtained by integrating Eq. 2.1 (b) Scaling of $\Delta T_c(\langle k \rangle)$ (defined in text) with $\langle k \rangle$; the line shows the best linear fit to the data. (c) The steady state density n_B of nodes in state B as a function of committed fraction p for Barabasi-Albert (BA) networks of different sizes with $\langle k \rangle = 10$, conditioned on survival of the system (symbols as in (a)).

$\alpha(p) \sim |p - p_c|^{\nu}$ where $\nu \approx 1.65$. Thus, below p_c, we have:

$$T_c(p < p_c) \sim \exp((p_c - p)^{\nu} N). \tag{2.8}$$

This exponential growth is presumably modulated by factors of $\log N$ which become dominant only when $p = p_c$. Above p_c, the dependance of T_c on p as seen from simulations is negligible (not shown).

2.3 The Effect on Sparse Networks

Next, we present simulation results for the case when the underlying network topology is chosen from an ensemble of Erdős-Rényi (ER) random graphs with given size N and given average degree $\langle k \rangle$. The qualitative features of the evolution of the system in this case are the same as that of the complete graph, although the critical fraction p_c displays some dependence on $\langle k \rangle$. For small $\langle k \rangle$ and fixed N, the drop in consensus times occurs slightly earlier in p for ER graphs than for a complete graph of the same size, as shown in Fig. 2.5(a). However for $p > p_c$, a complete graph has shorter consensus times (on average) than an ER graph of the same size. Above

p_c, the difference between consensus times for a graph with average degree $\langle k \rangle$ and the complete graph, ΔT_c, decays approximately as power law with increasing $\langle k \rangle$ (Fig. 2.5(b)). The deviation from a perfect power law is likely due to other weaker $\langle k \rangle$ dependent terms, presumably logarithmic in $\langle k \rangle$.

We also performed simulations of the binary agreement model on Barabasi-Albert (BA) networks (Fig. 2.5(c)), and found similar qualitative behavior as observed for ER networks including the difference from mean-field behavior. We leave a detailed analysis of the dependence of the critical fraction p_c and the consensus times T_c on the average degree $\langle k \rangle$ of sparse networks, for future work.

2.4 Tipping Points on Real-world Social Networks

Next, we present critical points determined by simulation (as for ER) for some real-world social network instances. These networks include High school friendship network[4], Email communication network [92], Facebook-like network [93] at University of California, Irvine and Citation network on high energy physics theory [94] from arXiv[5]. As shown in Figs. 2.7, 2.8, 2.9 and 2.6, all critical points on tested graphs are bounded by 10% (mean-field) and also bounded by that of BA with similar average degree. More specifically, p_c is found to be 4.8% and 5% for High school network and Email network respectively (compared with 6.6% for BA with $N = 1000, \langle k \rangle = 10$); p_c is found to be 7.8% for Facebook-like network (compared with 7.8% for BA with $N = 2000, \langle k \rangle = 14$); p_c is found to be 2% for Citation network (compared with 7.4% for BA with $N = 6000, \langle k \rangle = 10$). These results also verify the weak dependence on the degree (i.e., the smaller the degree, the smaller the tipping point) observed in [85, 95].

2.5 Summary

We show how the prevailing majority opinion in a population can be rapidly reversed by a small fraction p of randomly distributed *committed* agents who consistently proselytize the opposing opinion and are immune to influence. Specifically,

[4] A project funded by the National Institute of Child Health and Human Development.
[5] http://arxiv.org.

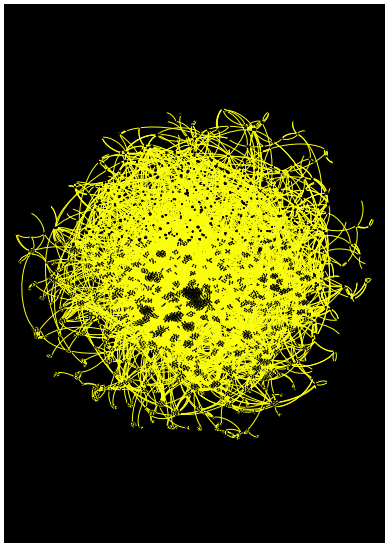

Figure 2.6: Citation network with $N = 5835$, $\langle k \rangle = 4.7$. p_c is found to be 2%.

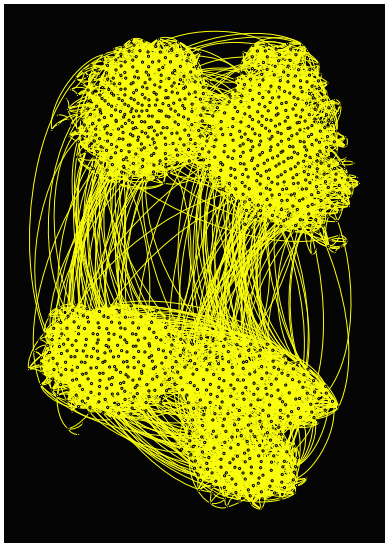

Figure 2.7: High school network with $N = 1127$, $\langle k \rangle = 9$. p_c is found to be 4.8%.

we show that when the committed fraction grows beyond a critical value $p_c \approx 10\%$, there is a dramatic decrease in the time, T_c, taken for the entire population to adopt the committed opinion. In particular, for complete graphs we show that when $p < p_c$, $T_c \sim \exp(c(p)N)$, while for $p > p_c$, $T_c \sim \ln N$. We conclude with simulation results for Erdős-Rényi random graphs which show qualitatively similar behavior.

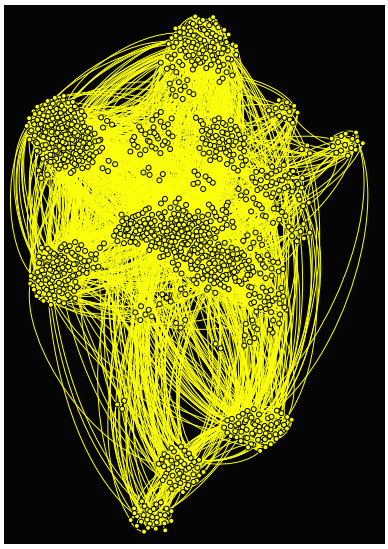

Figure 2.8: Email network with $N = 1133$, $\langle k \rangle = 9.6$. p_c is found to be 5%.

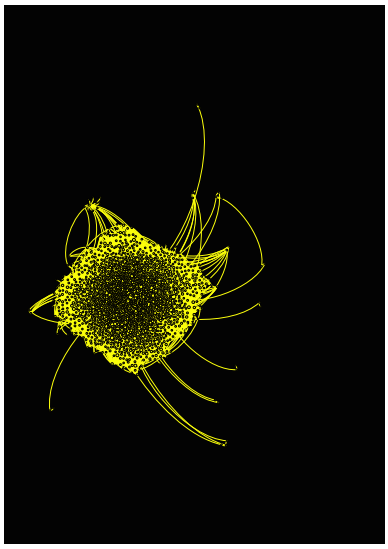

Figure 2.9: Facebook-like network with $N = 1893$, $\langle k \rangle = 15$. p_c is found to be 7.8%.

CHAPTER 3
INFLUENCE OF COMPETING COMMITTED GROUPS

3.1 The Model

Public opinion is often affected by the presence of committed groups of individuals dedicated to competing points of view. In this chapter, we study the more general case of opinion evolution when two groups committed to distinct opinions A and B are present in the network. The social dynamic of the system is the same as the one defined in Chapter 2. As before, we designate the densities of uncommitted agents in states A, B and AB by n_A, n_B and n_{AB}. We also designate the fraction of nodes committed to states A, B by p_A, p_B respectively. These quantities naturally obey the condition: $n_A + n_B + n_{AB} + p_A + p_B = 1$*.

3.2 The Effect on Complete Graphs

3.2.1 Phase Diagram in the Parameter Space

In the asymptotic limit of network size, and neglecting fluctuations and correlations, the system can be described by the following mean-field equations, for given values of the parameters p_A and p_B:

$$\begin{aligned} \frac{dn_A}{dt} &= -n_A n_B + n_{AB}^2 + n_A n_{AB} + \frac{3}{2} p_A n_{AB} - p_B n_A \\ \frac{dn_B}{dt} &= -n_A n_B + n_{AB}^2 + n_B n_{AB} + \frac{3}{2} p_B n_{AB} - p_A n_B. \end{aligned} \tag{3.1}$$

The evolution of n_{AB} follows from the constraint on densities defined above. In general, the evolution of the system depends on the relative values of p_A and p_B. In the case of $p_A > 0$, $p_B = 0$ (or equivalently, $p_B > 0$, $p_A = 0$) there is only a single group of committed nodes in the network, all of whom subscribe to the same opinion. This was the case studied in Chapter 2 and [26,30,84]. In this scenario, a transition

*Portions of this chapter previously appeared as: J. Xie, J. Emenheiser, M. Kirby, S. Sreenivasan, B. K. Szymanski, and G. Korniss, Evolution of opinions on social networks in the presence of competing committed groups, PLoS ONE, vol. 7, no. 3, p. e33215, Mar. 2012 [85].

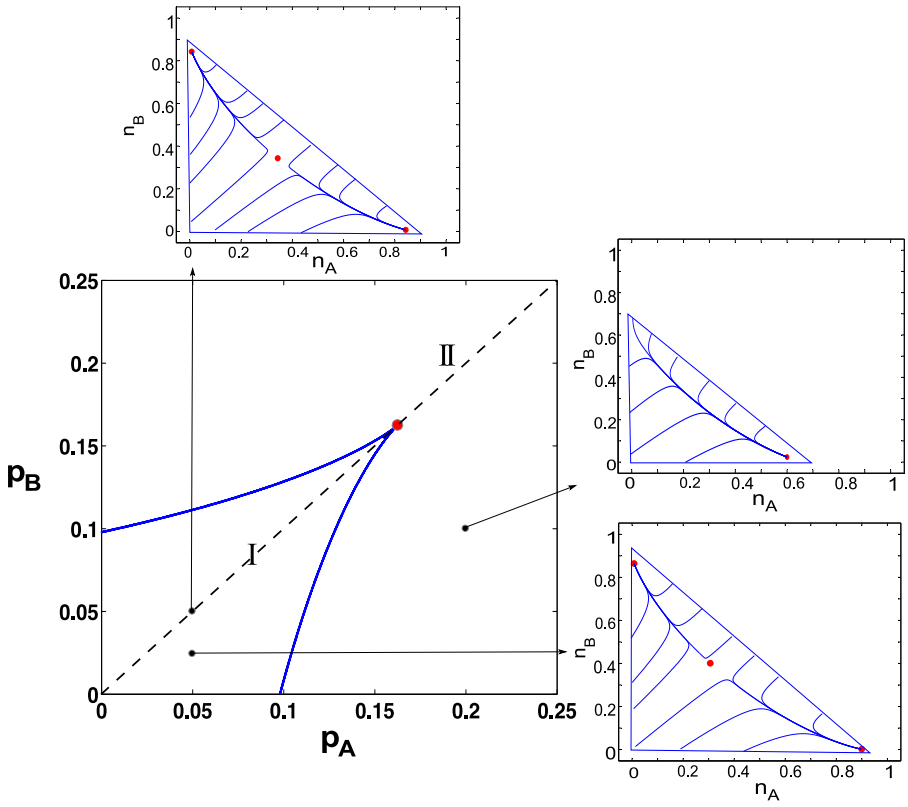

Figure 3.1: Mean-field picture in parameter space. The phase diagram obtained by integrating the mean-field Eqs. (3.1). The two lines indicate saddle-node bifurcation lines which form the boundary between two regions with markedly different behavior in phase space. For any values of parameters within the beak, denoted as region I, the system has two stable fixed points separated by a saddle point. Outside of the beak, in region II, the system has a single stable fixed point. The saddle-node bifurcation lines meet tangentially and terminate at a cusp bifurcation point.

is observed when this committed group constitutes a critical fraction of the total network. Specifically, the transition point separates two dynamical scenarios in the phase space, (n_A, n_B), of uncommitted node densities. Below the critical value, the absorbing state (e.g., $n_A = 1 - p_A$, $n_B = n_{AB} = 0$ when $p_A > 0$, $p_B = 0$) coexists in phase space with a stable mixed steady-state and an unstable fixed ("saddle") point. At or above the critical value, the latter non-absorbing steady-state and the

saddle point cease to exist. Consequently, for a finite system, reaching the (all A) consensus state requires an exponentially long time when p is less than the critical value. Beyond the critical value this time grows only logarithmically with network size. Note that this critical value or threshold is analogous to a spinodal point [96,97] associated with an underlying first-order (or discontinuous) transition in equilibrium systems.

In order to effectively characterize the behavior of the system governed by Eqs. (3.1) for $p_A,\ p_B > 0$, we systematically explore the parameter space $(p_A,\ p_B)$ by dividing it into a grid with a resolution of 0.000125 along each dimension. We then numerically integrate Eqs. (3.1) for each $(p_A,\ p_B)$ pair on this grid, assuming two distinct initial conditions, $n_A = 1 - p_A - p_B$, $n_B = n_{AB} = 0$ and $n_B = 1 - p_A - p_B$, $n_A = n_{AB} = 0$, representing diagonally opposite extremes in phase space. The results of this procedure reveal the picture shown in Fig. 3.1 in different regions of parameter space. As is obvious, with non-zero values for both $p_A,\ p_B$, consensus on a single opinion can never be reached, and therefore all fixed points (steady-states) are non-absorbing. With $(p_A,\ p_B)$ values within the region denoted as I which we refer to as the "beak" (borrowing terminology used in [98]), the phase space contains two stable fixed points, separated by a saddle point, while outside the beak, in region II, only a single stable fixed point exists in phase space. In region I, one fixed point corresponds to a state where opinion A is the majority opinion (A-dominant) while the other fixed point corresponds to a state where opinion B constitutes the majority opinion (B-dominant). Figure 3.1 shows representative trajectories and fixed points in phase space, in different regions of parameter space. Similar phase diagrams have been found in other two-parameter systems in different contexts including chemical reactions [98] and genetic switches [99].

3.2.2 Properties of the Transitions

In order to study the nature of the transitions that occur when we cross the boundaries of the beak, we parametrize the system by denoting $p_B = cp_A$ where c is a real number. Then, we systematically analyze the transitions occurring in two cases: (i) $c = 1$ and (ii) $c \neq 1$. It can be shown that along the diagonal line $c = 1$

the system undergoes a cusp bifurcation at $p_A = p_B = 0.1623$. The movement of the fixed points as p_A and p_B are smoothly varied along the diagonal line is shown in Fig. B.1. Henceforth, we denote the value of p_A and p_B at the cusp as p_c. As is well known, at the cusp bifurcation two branches of a saddle-node (or fold) bifurcation meet tangentially [100]. These two bifurcation curves form the boundary of the beak shown in Fig. 3.1. A detailed analysis demonstrating that $p_A = p_B = p_c$ constitutes a cusp bifurcation, as well as a semi-analytical derivation of the bifurcation curves is provided in Appendix B.1-B.3. The cusp bifurcation point is analogous to a second-order (or continuous) critical point seen in equilibrium systems, while bifurcation curves are analogous to spinodal transition lines.

Next, we study the stochastic evolution of opinions on finite-sized complete graphs through simulations. Here, we systematically vary c from 1 to 0 to obtain the right bifurcation curve, and therefore by virtue of the A-B symmetry in the system, also obtain the left bifurcation curve. In particular for a given value of c we obtain the transition point by varying p_A (with $p_B = cp_A$) and measuring the quantity:

$$m = (n_B - n_A)/(1 - p_A - p_B), \tag{3.2}$$

which we utilize as an order parameter. The above order parameter is analogous to the "magnetization" in a spin system as it captures the degree of dominance of opinion B over opinion A and is conventionally used to characterize the nature of phase transitions exhibited by such a system. Figures. 3.2(a) and (b) show the steady-state magnetization m, for successive $p_A,\ p_B$ pairs along lines of slope $c = 1$ and $c = 0.5$ respectively that pass through the origin. The $c = 1$ line in parameter space passes through the cusp point and gives rise to a second-order phase transition, while the $c = 0.5$ line passes through a point on the (right) bifurcation line giving rise to a first-order phase transition. Here 10 realizations of social influence dynamics were performed for each $p_A,\ p_B$ pair, starting from the initial condition $n_A = 0,\ n_B = 1 - p_A - p_B$, and the magnetization was measured conditioned on the system remaining in the steady state that it initially converged to.

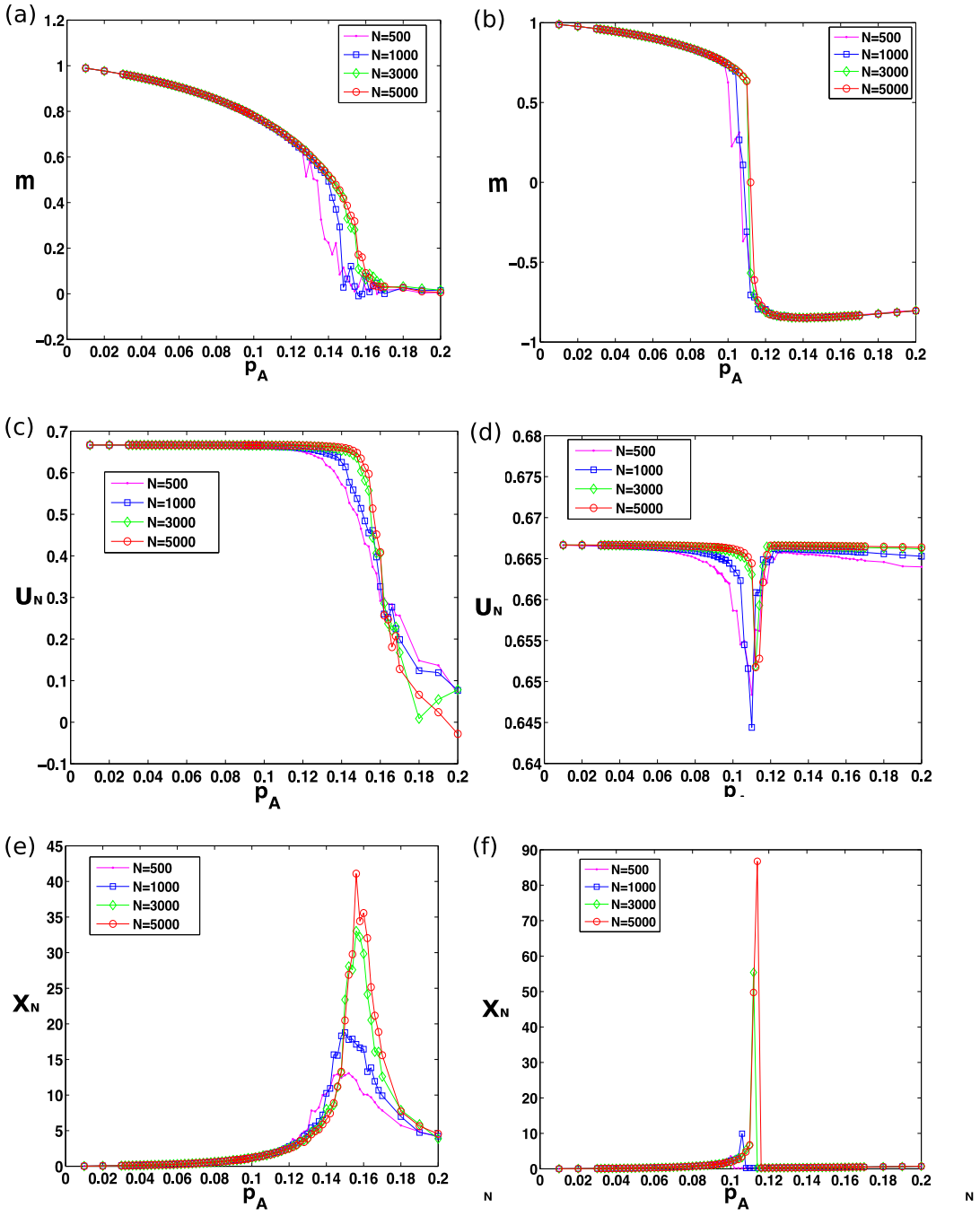

Figure 3.2: Behavior of typical order parameters as a function of linear trajectories of slope c that passes through the origin, in parameter space for a complete graph. (a)-(b) Steady-state magnetization m; (c)-(d) Binder cumulant U_N; (e)-(f) Scaled variance, X_N. Left column is for successive $(p_A,\ p_B)$ pairs along lines of slope $c = 1$, and right column is for $c = 0.5$ that passes through the origin. Data for (c), (d), (e) and (f) were generated from 10 realizations of the social influence dynamics, per $(p_A,\ p_B)$ pair, for each of two initial conditions: $n_A = 1 - p_A - p_B,\ n_B = 0$ and $n_A = 0,\ n_B = 1 - p_A - p_B$.

Another quantity, the Binder cumulant, defined as

$$U_N = 1 - \left[\frac{\langle m^4 \rangle}{3\langle m^2 \rangle^2}\right] \tag{3.3}$$

for a system of size N, is commonly used to distinguish between different types of phase transitions [96]. The utility of the Binder cumulant comes from the markedly different signatures we expect it to produce along a spinodal trajectory (e.g., $c = 0.5$), one that passes through the spinodal line, and one along a trajectory that passes through the critical point (e.g., along the diagonal, $c = 1$). This difference arises from the following distinction in the evolution of the distribution of m, $P(m)$, along these trajectories. Along a spinodal trajectory starting from a point where $p_A = p_B$, an initially symmetric (about $m = 0$), bimodal $P(m)$ becomes asymmetric and unimodal upon crossing the spinodal line, with the single mode eventually becoming a delta function. In contrast, along the diagonal trajectory in parameter space, $P(m)$ is initially a double-delta distribution (for $p_A = p_B \ll p_c$), symmetric about $m = 0$, and it smoothly transitions to a zero-centered Gaussian distribution as the critical point is crossed. The definition of U_N indicates that $U_N = 2/3$ for a delta function distribution (also for a symmetric, double-delta distribution about $m = 0$), while $U_N = 0$ for a zero-centered Gaussian distribution, and thus readily yields the limiting U_N values at both extremes of the spinodal and diagonal trajectory. As illustrated in Figs. 3.2(c) and (d), U_N as a function of p_A shows distinct behaviors for $c = 1$ and $c = 0.5$, indicating the existence of a second-order (or continuous) transition point at $p_A = p_B = p_c(N)$ (Fig. 3.2(c)) and first-order (or discontinuous) phase transition points (Fig. 3.2(d)) along off-diagonal trajectories [96], respectively. The second-order critical point $p_c(N)$ converges to the mean-field value, $p_c \approx 0.1623$, as N becomes larger. The dip observed in U_N along the off-diagonal trajectory serves as an excellent estimator of the location of the first-order (spinodal) transition for a finite network. Thus, to reiterate, for a finite network, the second-order transition point and the first-order transition (spinodal) lines are respective analogues of the cusp bifurcation point and the saddle-node bifurcation curves observed in the mean-field case.

The fluctuations of the quantity m can also be used to identify a transition

point, particularly for the case of the second-order transition. In particular, in formal analogy with methods employed in the study of equilibrium spin systems, the scaled variance:

$$X_N = N\langle(|m| - \langle|m|\rangle)^2\rangle \tag{3.4}$$

serves as an excellent estimate for the second-order transition point p_c for a finite network. As shown in Fig. 3.2(e), X_N peaks at a particular value of p_A, with the size of the peak growing with N (and expected to diverge as $N \to \infty$). In the case of the spinodal transition, one studies fluctuations of m ($X_N = N\langle(m - \langle m\rangle)^2\rangle$) restricted to the metastable state [101, 102] until the spinodal point (Fig. 3.2(f)) at which the metastable state disappears, and fluctuations of m in the unique stable state beyond the spinodal point (Fig. 3.2(f)).

Figure 3.3 shows the bifurcation (spinodal) lines obtained via simulations of finite complete graphs by using the Binder cumulant (Fig. 3.2(d))to identify the location of the spinodal phase transition, and demonstrates that its agreement with the mean-field curves improves as N grows. The cusp points shown here are identified in simulations as the locations where X_N reaches its peak value (Fig. 3.2(e)).

3.2.3 Long Time Scale: the Switching Times

In the region within the beak, the switching time between the coexisting steady-states represents the longest time-scale of relevance in the system. The switching time is defined as the time the system takes to escape to a distinct coexisting steady-state, after having been trapped in one of the steady-states. Figures 3.4(a) and (b) show sample evolutions of the system, demonstrating respectively, the switching between steady-states within the beak, and the fluctuations about the single steady state outside the beak. In stochastic systems exhibiting multistability or metastability, it is well known that switching times increase exponentially with N for large N (the weak-noise limit) [98, 103–105] Furthermore, the exponential growth rate of the switching time in such cases can be determined using the eikonal approximation [98, 106]. The basic idea in the approximation involves (i) assuming an eikonal form for the probability of occupying a state far from the steady-state and (ii) smoothness of transition probabilities in the master equation of the system.

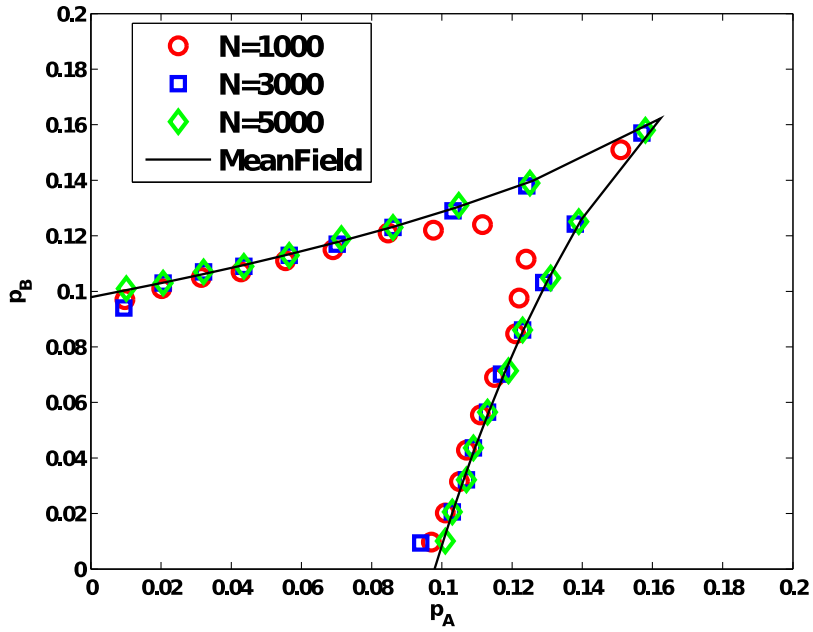

Figure 3.3: Picture in parameter space for a complete graph obtained from analytical and simulation results. The bifurcation lines and the cusp point in parameter space were obtained analytically from the mean field equations and are compared with those found using simulations for finite-sized complete graphs. Analytical and simulation curves show excellent agreement as N increases. The location of the transition occurring across the bifurcation curve was obtained using the Binder cumulant U_N (Fig. 3.2(d)), while the location of the cusp point was obtained by using variance of m (Fig. 3.2(e)). For both analytical and simulation results, the bifurcation curves are obtained by identifying the critical points that lie on linear trajectories in parameter size described by $p_B = cp_A$. This process is carried out for different values of c between 0 and 1 at intervals of 0.1, and for each value of c, p_A is varied at a resolution of 0.002. In simulations, for each such combination of $(p_A,\ p_B)$ obtained, we perform averages over 10 realizations of the social influence dynamics, for each of two initial conditions, $n_A = 1-p_A-p_B$, and $n_A = 0$, with $n_B = 1-n_A-p_A-p_B$ for each case.

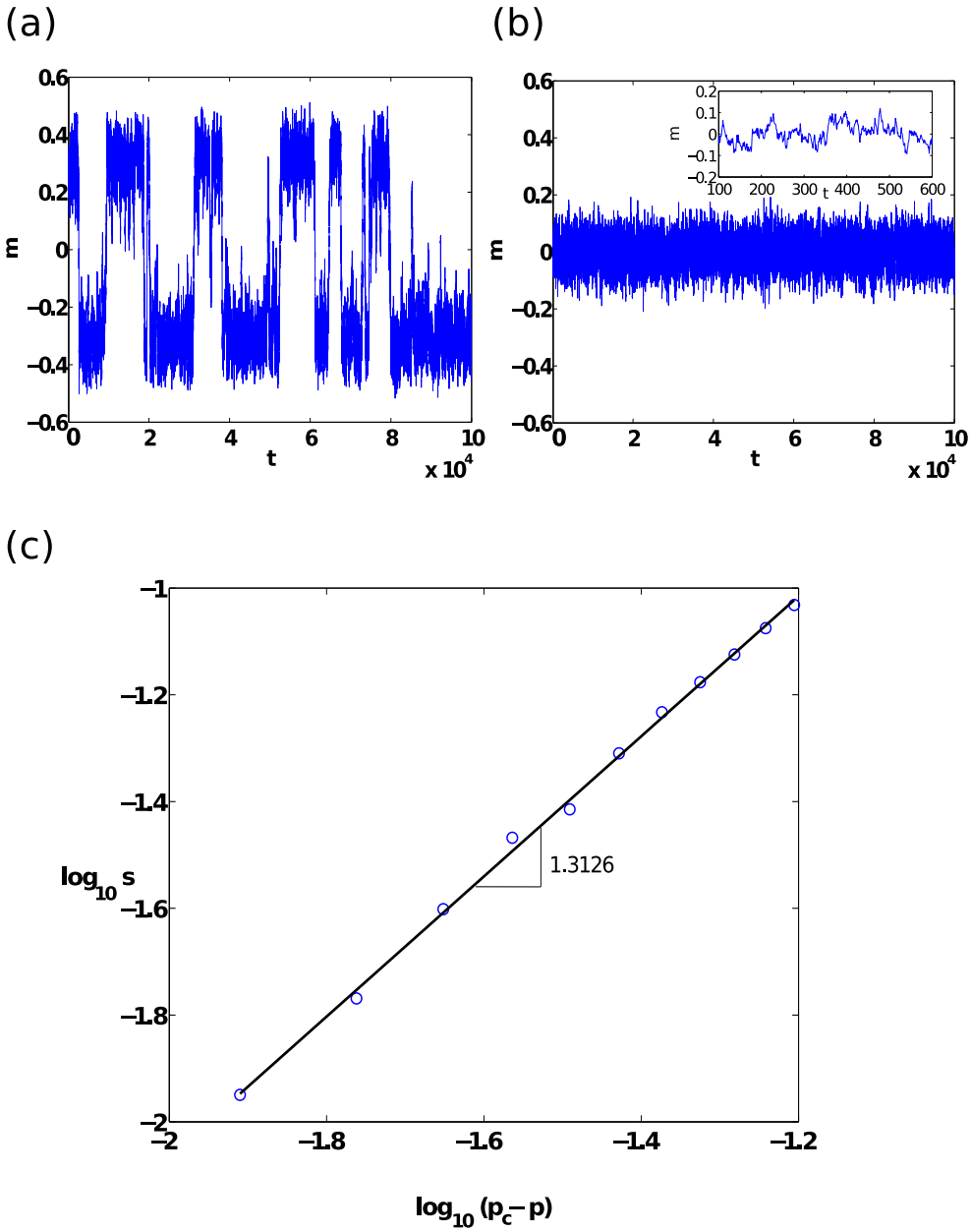

Figure 3.4: Evolution of order parameter m and the exponential growth in switching time as a function of distance from the second-order critical point. (a) Switches in the value of m as a function of time t for a sample evolution (with initial transient removed) of the system when $p_A = p_B = 0.154$ ($< p_c$). This reflects the system repeatedly switching between the A-dominant steady-state ($m > 0$) and the B-dominant steady-state ($m < 0$). (b) Sample evolution of the system (with initial transient removed) for $p_A = p_B = 0.2$ ($> p_c$). The system fluctuates randomly about the only existing steady-state in which densities of A and B nodes are equal. (c) The dependence of s in the exponential scaling $T_{\text{switching}} \sim \exp(sN)$ when $p_A = p_B = p$ ($p < p_c$) as a function of $(p_c - p)$, obtained using the eikonal approximation (see Appendix B.4).

This allows the interpretation of fluctuational trajectories as paths conforming to an auxiliary Hamilton-Jacobi system. This in turn enables us to calculate the probability of escape allowing an *optimal fluctuational path* that takes the system from the vicinity of the steady-state to the vicinity of the saddle point of the deterministic system. The switching time is simply the inverse of the probability of escape along this optimal fluctuational path. We defer details of this procedure to Appendix B.4. Using this approach we find that for the symmetric case, $p_A = p_B = p < p_c$, the exponential growth rate of the switching time $s \sim (p_c - p)^\nu$ with $\nu \approx 1.3$ (Fig. 3.4(c)). Thus, along the portion of the diagonal within the beak:

$$T_{\text{switching}} \sim \exp[(p_c - p)^\nu N]. \tag{3.5}$$

Outside the beak, the time to get arbitrarily close to the sole steady-state value grows logarithmically with N (not shown).

The results presented so far show that there exists a transition in the time needed by a committed minority to influence the entire population to adopt its opinion, even in the presence of a committed opposition (i.e., in the case where both $p_A,\ p_B > 0$), as long as $p_A,\ p_B < p_c$. (Note that the case $p_A > 0,\ p_B = 0$ was considered in Chapter 2 [84]). For example, assume that initially all the uncommitted nodes adopt opinion B, and that $p_A = p_B < p_c$. Then, the steady-state that the system reaches in $\ln(N)$ time is the one in which the majority of nodes hold opinion B. Despite the fact that there exist committed agents in state A continuously proselytizing their state, it takes an exponentially long time before a large (spontaneous) fluctuation switches the system to the A-dominant steady-state. For identical initial conditions, the picture is qualitatively the same if we increase p_A keeping p_B fixed, as long as $(p_A,\ p_B)$ lies within the beak. However, when $(p_A,\ p_B)$ lies on the bifurcation curve or beyond, the B-dominant steady-state vanishes, and with the same initial conditions, where B is the initial majority, it takes the system only $\ln(N)$ time to reach the A-dominant state (the only existing steady-state). Thus, for every value of an existing committed fraction p_B ($< p_c$) of B nodes, there exists a corresponding critical fraction of A nodes beyond which it is guaranteed that the system will reach an A dominant state in $\ln(N)$ time, irrespective of the initial

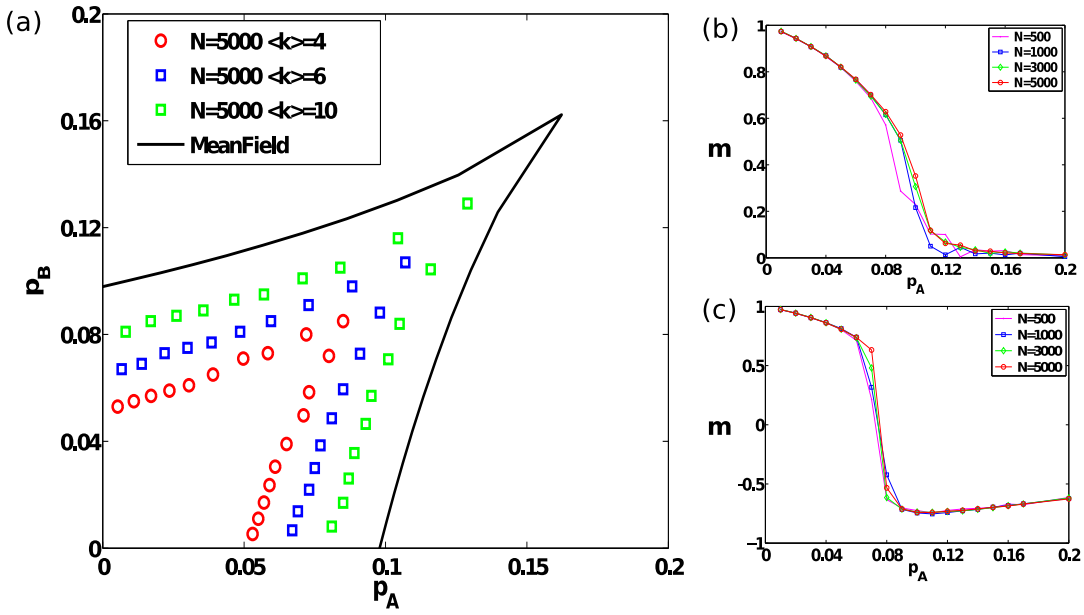

Figure 3.5: Results for Erdős-Rényi random graphs. (a)The bifurcation lines and cusp point in parameter space obtained through simulations of Erdős-Rényi random graphs of size $N = 5000$ with different average degrees. The mean-field analytical curve is shown for comparison. For simulation results, the bifurcation curves are obtained by identifying the critical points that lie on linear trajectories described by $p_B = cp_A$ in parameter space. This process is carried out for different values of c between 0 and 1 at intervals of 0.1, and for each value of c, p_A is varied at a resolution of 0.002. For each such combination of $(p_A,\ p_B)$ obtained, we perform averages for quantities of interest over 10 realizations of networks (with a single realization of the social influence dynamics per network), for each of two initial conditions, $n_A = 1 - p_A - p_B$ and $n_A = 0$ with $n_B = 1 - n_A - p_A - p_B$ in each case. (b)-(c) Steady-state magnetization for ER graphs with $\langle k \rangle = 6$ and different sizes N, as parameter pair values are varied successively along slope $c = 1$ and slope $c = 0.5$ lines in parameter space respectively.

conditions. However, for any trajectory in the parameter space in a region where either p_A or p_B is (or both are) greater than p_c, no abrupt changes in dominance or consensus times are observed. Instead, the dominance of A or B at the single fixed point smoothly varies as the associated committed fractions are varied. Moreover, the system always reaches this single fixed point in $\ln(N)$ time.

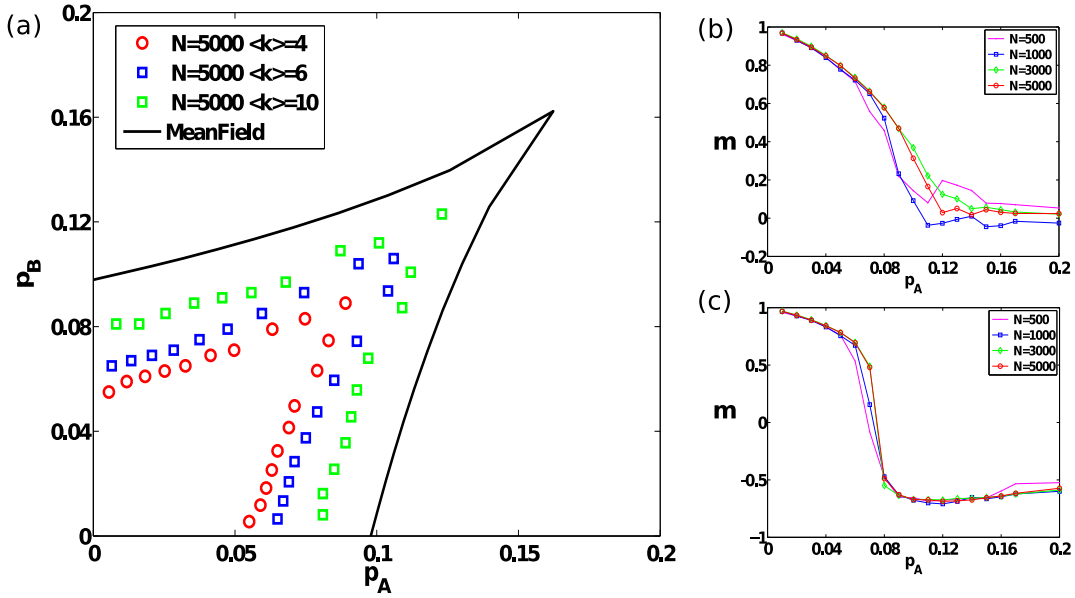

Figure 3.6: Results for Barabási-Albert networks. (a)The bifurcation lines and cusp point in parameter space obtained through simulations of Barabási-Albert networks of size $N = 5000$ with different average degrees. For simulation results, the bifurcation curves are obtained by a similar method as described in the legend of Fig. 3.5(a). (b)-(c) Steady-state magnetization for BA networks with $\langle k \rangle = 6$ and different sizes N, as parameter pair values are varied successively along slope $c = 1$ and slope $c = 0.5$ lines in parameter space respectively.

3.3 The Effect on Sparse Networks

Finally, we study how opinions evolve in the presence of committed groups on sparse graphs, most relevant to social networks. We study Erdős-Rényi (ER) random graphs [107] as well as Barabási-Albert networks [108]. For each of these sparse networks, we find the same qualitative behavior as found for the complete graph. As shown in Figs. 3.5 and 3.6, as the average degree of the sparse networks increases, the bifurcation lines in parameter space tend to approach their mean-field counterparts. Although we do not study sparse networks analytically here, we note that in another instance of a phase transition for a similar model studied in [109], it was demonstrated using heterogeneous mean-field equations that the behavior of sparse networks is qualitatively similar to that of complete graphs. Figure 3.7

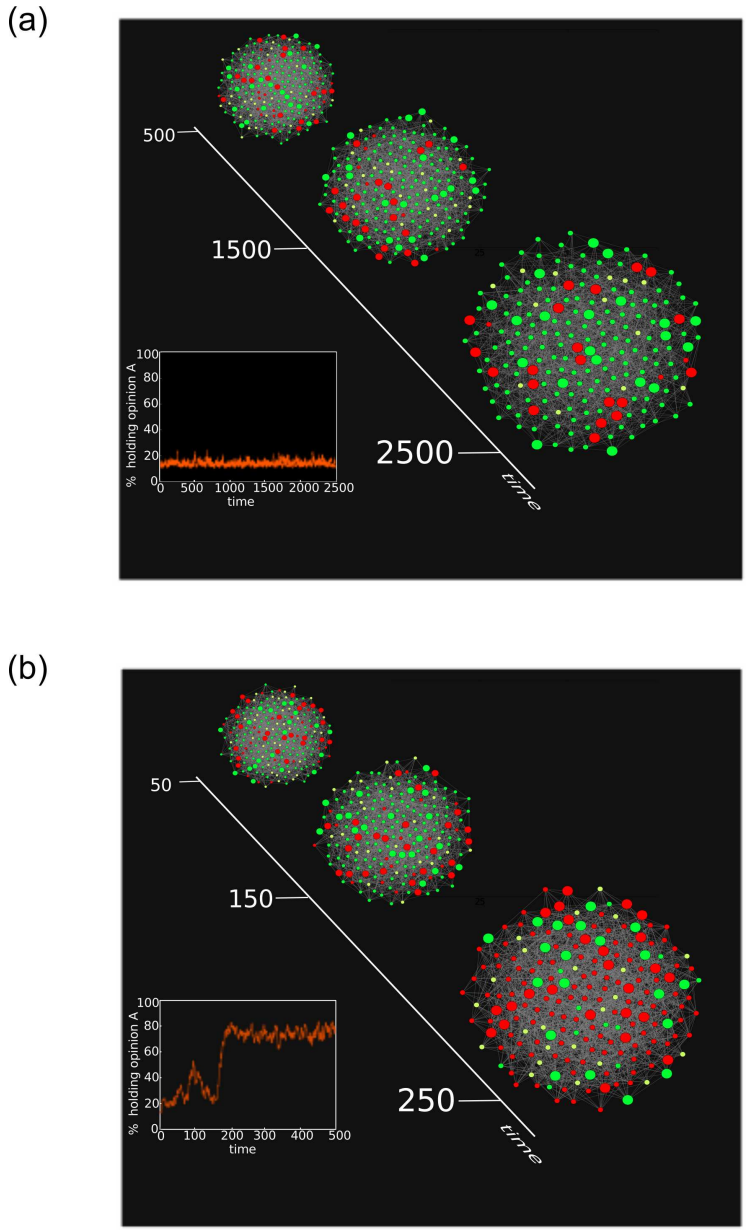

Figure 3.7: Visualization of opinion evolutions. The evolution of opinions on an ER random graph with $N = 200$ and $\langle k \rangle = 6$ for two $(p_A,\ p_B)$ pairs. In each case $n_B = 1 - p_A - p_B$ and $n_A = 0$. Nodes holding opinion A are depicted in red, while nodes holding opinion B are shown in green. Nodes with larger diameters are committed nodes. Top: The case $p_A = p_B = 0.1$ for which the system is in region I in parameter space (following the terminology of Fig. 3.1, and the system is trapped in a B-dominant steady-state. Even after 2500 time steps, the system continues to remain trapped in this state (inset) with $n_A \approxeq 0.05$. Bottom: The case $p_A = 0.125$ and $p_B = 0.1$ for which the system is in the region II, and undergoes an abrupt transition (inset) to the A-dominant state within 250 time steps.

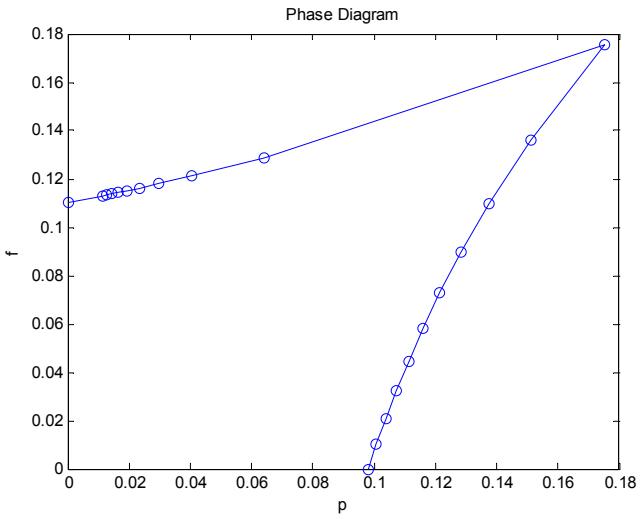

Figure 3.8: The bifurcation boundaries in the parameter space for mean-field with both external media and committed group.

visually depicts typical instances of the evolution of opinions on an ER random graph for $(p_A,\ p_B)$ values within and outside the beak.

3.4 External Media vs. Committed Group

Another factor that affects the evolution of opinions is the existence of external influence like media (TV, radio, online news channel, forum etc). The entire population or the majority of the population may experience propaganda of a certain opinion. Such influence usually exerts itself on top of personal interactions with or without committed agents. In the section, we model such global influence of one opinion against the other competing committed opinion. Since the media persists during the entire evolution, it can be considered as some form of committed group, equivalent to the external field in Ising model. Suppose that the committed opinion is A and the field with strength f_B favors opinion B. The dynamics is now as follows. With probability $1 - f_B$, we apply the usual game rules when an interaction happens. With probability f_B, we apply the special rules that reinforce the influence of the field, specifically,

- when a listener with A is picked, it becomes AB (i.e., $A \to AB$);

- when a listener with AB is picked, it becomes B (i.e., $AB \rightarrow B$);
- when a listener with B is picked, it stays B (i.e., $B \rightarrow B$);

Note that in this model, the field impacts non-committed nodes regardless of the states (i.e., A, B or AB). The mean-field equations that describe the system are given by:

$$\begin{aligned} \frac{dn_A}{dt} &= (1-f_B)\cdot(-n_B n_A + n_{AB}^2 + n_A n_{AB} + 1.5p n_{AB}) - f_B n_A \\ \frac{dn_B}{dt} &= (1-f_B)\cdot(-n_A n_B + n_{AB}^2 + n_B n_{AB} - p n_B) + f_B n_{AB}. \end{aligned} \tag{3.6}$$

As shown in Fig. 3.8, the phase diagram of this model is quantitatively the same as the two committed groups model, wherein two fold-bifurcation curves meet at the cusp point. However, the system is now asymmetric. Without the media, the committed opinion wins when the p is larger than 0.0979 (i.e., reduce to the single committed opinion case). In the case there is no committed opinion, it requires the influence strength f_B to be larger than 0.11 to win the majority. It appears that in this model, even though the media reaches large portion of the population directly, the impact is not as strong as that of the committed group through local interactions.

3.5 Summary

We show for stylized social networks (including Erdős-Rényi random graphs and Barabási-Albert scale-free networks) that the phase diagram of this system in parameter space (p_A, p_B) consists of two regions, one where two stable steady-states coexist, and the remaining where only a single stable steady-state exists. These two regions are separated by two fold-bifurcation (spinodal) lines which meet tangentially and terminate at a cusp. We provide insights to the phase diagram and to the nature of the underlying phase transitions by investigating the model on infinite (mean-field limit), finite complete graphs and finite sparse networks. For the latter case, we derive the scaling exponent associated with the exponential growth of switching times as a function of the distance from the critical point.

CHAPTER 4
SLPA: TOWARDS LINEAR TIME OVERLAPPING COMMUNITY DETECTION

Community or modular structure is considered to be a significant property of real-world social networks. Thus, numerous techniques have been developed for community detection. However, most of the work has been done on *disjoint* community detection. It has been well understood that people in a real social network are naturally characterized by *multiple* community memberships. For example, a person usually has connections to several social groups like family, friends and colleges; a researcher may be active in several areas; in the Internet, a person can simultaneously subscribe to an arbitrary number of groups. For this reason, discovering *overlapping* structures is necessary for realistic social analysis*.

Overlapping community detection algorithms aim to discover a *cover* [51], defined as a set of clusters in which each node belongs to at least one cluster. In this chapter, we propose an efficient algorithm for detecting both individual overlapping nodes and overlapping communities using the underlying network structure alone.

4.1 Related Work

We briefly summarized most relevant work in this section. More comprehensive reviews are presented in our survey paper [40].

Clique Percolation: CPM [38] is based on the assumption that a community consists of fully connected subgraphs and detects overlapping communities by searching for *adjacent* cliques. CPMw [41] extends CPM for weighted networks by introducing a subgraph intensity threshold.

*Portions of this chapter previously appeared as: J. Xie, B. K. Szymanski, and X. Liu, SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, in Proc. ICDM Workshop, 2011, pp. 344-349 [73],
J. Xie and B. K. Szymanski, Towards linear time overlapping community detection in social networks, in Proc. PAKDD Conf., 2012, pp. 25-36 [86], and
J. Xie, S. Kelley, and B. K. Szymanski, Overlapping community detection in networks: the state of the art and comparative study, ACM Comput. Surv. (to be published), 2012 [40].

Local Expansion: The iterative scan algorithm (IS) [49], [50] expands small cluster cores by adding or removing nodes until a local density function cannot be improved. The quality of seeds dictates the quality of discovered communities. LFM [51] expands a community from a random node. The size and quality of the detected communities depends significantly on the resolution parameter of the fitness function. EAGLE [58] and GCE [60] start with all maximal cliques in the network as initial communities. EAGLE uses the agglomerative framework to produce a dendrogram in $O(n^2 s)$ time, where n is the number of nodes, and s is the maximal number of join operations. In GCE communities that are similar within a distance ϵ are removed. The greedy expansion takes $O(mh)$ time, where m is the number of edges, and h is the number of cliques.

Fuzzy Clustering: Zhang [63] used the spectral method to embed the graph into low dimensionality Euclidean space. Nodes are then clustered by the fuzzy c-mean algorithm. Psorakis et al. [70] proposed a model based on Bayesian nonnegative matrix factorization (NMF). These algorithms need to determine the number of communities K and the use of matrix multiplication makes them inefficient. For NMF, the complexity is $O(Kn^2)$.

Link Partitioning: Partitioning links instead of nodes to discover communities has been explored, where the node partition of a link graph leads to an edge partition of the original graph. In [43], single-linkage hierarchical clustering is used to build a link dendrogram. The time complexity is $O(nk_{max}^2)$, where k_{max} is the highest degree of the n nodes.

Dynamical Algorithms: Label propagation algorithms such as [72, 81, 110] use labels to uncover communities. In COPRA [72], each node updates its belonging coefficients by *averaging* the coefficients from all its neighbors in a synchronous fashion. The time complexity is $O(vm \log(vm/n))$ per iteration, where parameter v controls the maximum number of communities with which a node can associate, m and n are the number of edges and number of nodes respectively.

4.2 SLPA: Speaker-listener Label Propagation Algorithm

Our algorithm is an extension of the Label Propagation Algorithm (LPA) [81]. In LPA, each node holds only a single label and iteratively updates it to its neighborhood majority label. Disjoint communities are discovered when the algorithm converges. Like [72], our algorithm accounts for *overlap* by allowing each node to possess multiple labels but it uses different dynamics with more general features.

SLPA mimics human pairwise communication behavior. At each communication step, each node serves as both a speaker (information provider) and a listener (information consumer). Specifically, each node broadcasts a label to neighbors and at the same time receives a label from each neighbor. Unlike other algorithms, each node has a *memory* of the labels received in the past, which is taken into account to make the current decisions. SLPA consists of two loops shown in Algorithm 1.

Algorithm 1 : SLPA(T, r)

T: the user defined maximum iteration
r: post-processing threshold
1) At t=0, the *memory* of each node is initialized with its node id.
2) For t=1:T
   All nodes are marked unvisited.
   While(there is any unvisited node)
   a. One unvisited node is randomly selected as a listener.
   b. Each neighbor (speaker) of the selected node randomly selects a label with probability proportional to the occurrence frequency in the memory and sends the selected label to the listener.
   c. The listener adds the most popular label received to its memory.
   d. Mark the listener visited.
3) Finally, the post-processing based on the labels in the memories and the threshold r is applied to output the communities.

Note that SLPA starts with each node being in its own community represented by its node id (a total of n), the algorithm explores the network and outputs the desired number of communities in the end. As such, the number of communities is not required as an input. Due to the step c, the size of memory increases by one for each node at each step. SLPA reduces to LPA when the size of memory is limited to one and the stop criterion is convergence of all labels. By considering the history

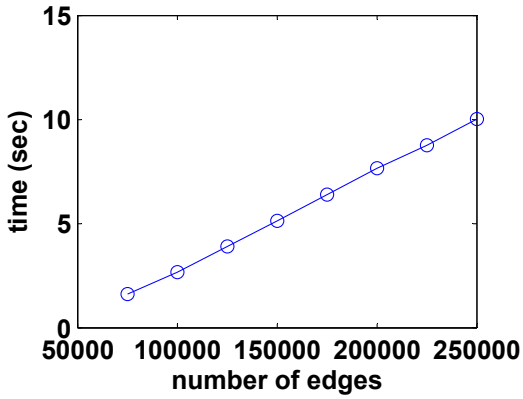

Figure 4.1: The average execution times of SLPA in synthetic networks with $n = 5000$ and average degree $\overline{k}$ varying from 10 to 80.

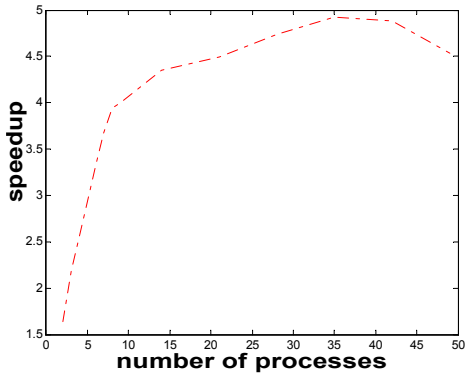

Figure 4.2: The speedup in the parallel version implemented with MPI on the Amazon co-purchasing network.

labels, SLPA has a good estimation of the local neighborhood and avoids producing a number of small sized communities as opposed to other algorithms. Empirically, SLPA produces relatively stable outputs, independent of network structure, when T is greater than 20. Although SLPA is non-deterministic due to the random selection, it performs well on average as shown in later sections.

Post-processing and Community Detection: In SLPA, the determination of communities is performed when the stored information is post-processed (Step 3). Given the memory of a node, SLPA converts it into a probability distribution of labels. Since labels represent community id's, this distribution naturally defines the *strength* of association to communities to which the node belongs. To produce *crisp* communities in which the membership of a node to a given community is *binary*, i.e., either a node is in a community or not, a simple thresholding procedure is performed: if the probability of seeing a particular label during the whole process is less than a given threshold $r \in [0, 0.5]$, this label is deleted. After thresholding, connected nodes having a particular label are grouped together and form a *community*. If a node contains multiple labels, it belongs to more than one community and is called an *overlapping node*. A smaller value of r produces a larger number of communities. However, the effective range is typically narrow in practice. When $r \geq 0.5$, SLPA

outputs disjoint communities, determined by the label with maximum probability in the memory. For two communities with subset relationship, SLPA generally keeps only the superset.

Complexity: The initialization of labels requires $O(n)$, where n is the total number of nodes. The outer loop is controlled by the user defined maximum iteration T, which is a small constant (e.g., in our experiments was set to 100). The inner loop is controlled by n. Each operation of the inner loop executes one speaking rule (i.e., Step (b)) and one listening rule (i.e., Step (c)). The speaking rule requires exactly $O(1)$ operation. The listening rule takes $O(\overline{k})$ on average, where $\overline{k}$ is the average node degree. In the post-processing, the thresholding operation requires $O(Tn)$ operations since each node has a memory of size T. In summary, the time complexity of the entire algorithm is $O(Tn\overline{k})$ or $O(Tm)$, linear with the total number of edges m. The execution times for LFR synthetic networks (see Section 4.3.1 for details) shown in Fig. 4.1 confirm the *linear* scaling of the execution times. On a desktop with 2.80GHz CPU, SLPA takes about six minutes to run over a two million nodes Amazon co-purchasing network[6], which is ten times faster than GCE [60].

SLPA is also easy to parallelize on a cluster or a supercomputer like Blue Gene. In a naive implementation with MPI (using MPICH2[7]) and on the Amazon co-purchasing network, we observed the speedup up to a factor of 5 with 35 processes on a cluster as shown in Fig. 4.2. The results are based on a Master-Slave architecture and random subnetwork partition. As expected, the communication cost becomes the bottleneck as the number of processes increases, and the gain with more processes starts vanishing beyond the bottleneck (i.e., 35 processes). However, when implemented on the Blue Gene/L using non-blocking message exchanges in a non-Master-Slave architecture, the speedup is more than 60, and the bottleneck is not yet observed for up to 1024 processes (in future work, we will continue parallelization of this algorithm).

[6]http://snap.stanford.edu/data.

[7]http://www.mcs.anl.gov/research/projects/mpich2.

Table 4.1: Algorithms included in the experiments.

Algorithm	Reference	Complexity	Imp.
CFinder	[38]	-	C++
LFM	[51]	$O(n^2)$	C++
EAGLE	[58]	$O(n^2 + (h+n)s)$	C++
CIS	[50]	$O(n^2)$	C++
GCE	[60]	$O(mh)$	C++
COPRA	[72]	$O(vm \log(vm/n))$	Java
Game	[74]	$O(m^2)$	C++
NMF	[70]	$O(kn^2)$	Matlab
MOSES	[68]	$O(en^2)$	C++
Link	[43]	$O(nk_{max}^2)$	C++
iLCD	[57]	$O(nk^2)$	Java
UEOC	[54]	$O(ln^2)$	Matlab
OSLOM	[53]	$O(n^2)$	C++
SLPA	[73]	$O(tm)$	C++

4.3 Tests in Synthetic Networks

4.3.1 Methodology

To study the behavior of SLPA, we conducted extensive experiments in both synthetic and real-world networks. For synthetic random networks, we adopted the widely used LFR benchmark[8] [111], which allows heterogeneous distributions of node degrees and community sizes.

We used networks with sizes $n \in \{1000, 5000\}$. The average degree is kept at $\overline{k} = 10$, which is of the same order as most large real-world social networks[9]. The rest of the parameters of LFR generator are set similar to those in [35]: node degrees and community sizes are governed by power law distributions with exponents $\tau_1 = 2$ and $\tau_2 = 1$ respectively, the maximum degree is $k_{max} = 50$, and community sizes vary in both small range $s = (10, 50)$ and large range $b = (20, 100)$. The mixing parameter μ is from $\{0.1, 0.3\}$, which is the expected fraction links through which a node connecting to other nodes in the same community. For each LFR network, we generated 10 instantiations.

The degree of overlap is determined by two parameters. O_n is the number of overlapping nodes, and O_m is the number of communities to which each overlapping node belongs. O_n is set to 10% and 50% of the total number of nodes, indicating

[8]http://sites.google.com/site/andrealancichinetti/files.
[9]http://snap.stanford.edu/data.

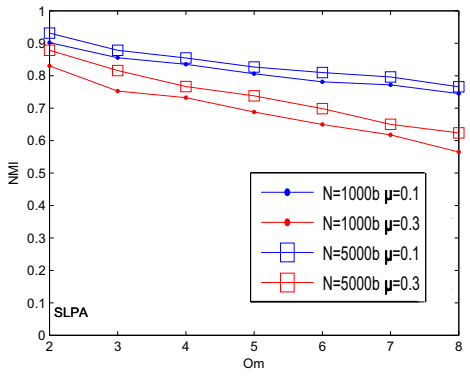

Figure 4.3: The effects of network size n and mixing parameter μ on LFR networks. Plots show NMI's for networks with large community size range and $O_n = 10\%$.

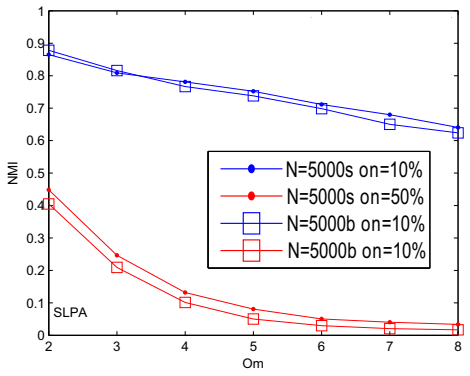

Figure 4.4: The effects of community size range and overlapping density O_n on LFR networks. Plots show NMI's for networks with $n = 5000$ and $\mu = 0.3$.

low and high *overlapping density* respectively. Instead of fixing O_m [35, 72], we also allow it to vary from 2 to 8 indicating the *overlapping diversity* of overlapping nodes. By increasing the value of O_m, we create harder detection tasks. This also allow us to look into more details of the detection accuracy at node level.

We compared SLPA with other thirteen algorithms (Table 4.1) representing different categories discussed in Section 4.1. For algorithms with tunable parameters, the performance with the optimal parameter is reported. For CFinder, k varies from 3 to 10; for COPRA, v varies from 1 to 10; For *Link*, the threshold varies from 0.1 to 0.9 with an interval 0.1. For SLPA, the number of iterations T is set to 100 and r varies from 0.01 to 0.1. The average performance together with error bar over ten repetitions are reported for SLPA and COPRA. For NMF, we applied a threshold varying from 0.05 to 0.5 with an interval 0.05 to convert it to a crisp clustering.

The extended normalized mutual information (NMI) [51] and Omega Index [112] are used to quantify the quality of communities discovered by an algorithm. NMI measures the fraction of nodes in agreement in two covers, while Omega is based on pairs of nodes. NMI yields values between 0 and 1. Omega typically yields values less than or equal to 1. The closer this value is to 1, the better the performance is.

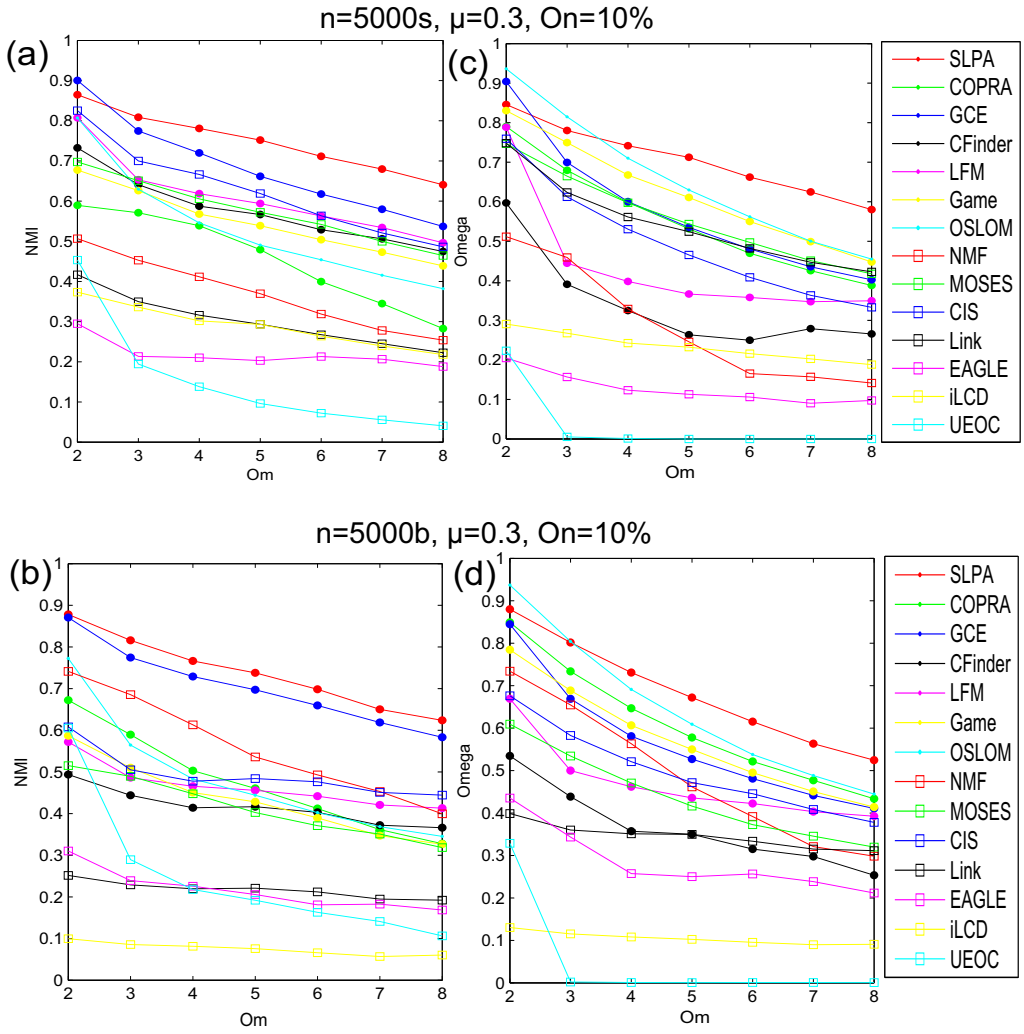

Figure 4.5: Evaluations of overlapping community detection on LFR networks with $O_n = 10\%$. Left column: NMI as a function of the number of memberships O_m; Right column: Omega as a function of the number of memberships O_m. Results for small community size range are shown in top row (i.e., (a) and (c)), and results for large community size range are shown in bottom row (i.e., (b) and (d)). All resutls are from networks with $n = 5000$ and $\mu = 0.3$.

Table 4.2: The community detection ranking for $n = 5000$, $\mu = 0.3$ and $O_n = 10\%$.

Rank	RS^s_{NMI}	RS^s_{Omega}	RS^b_{NMI}	RS^b_{Omega}	$RS^*_{NMI,Omega}$	RS^*_F
1	SLPA	SLPA	SLPA	SLPA	SLPA	SLPA
2	GCE	OSLOM	GCE	OSLOM	GCE	CFinder
3	CIS	Game	NMF	COPRA	OSLOM	Game
4	LFM	GCE	CIS	Game	CIS	OSLOM
5	MOSES	MOSES	COPRA	GCE	Game	MOSES
6	CFinder	COPRA	OSLOM	CIS	COPRA	COPRA
7	Game	Link	LFM	NMF	LFM	iLCD
8	OSLOM	CIS	Game	LFM	MOSES	Link
9	COPRA	LFM	CFinder	MOSES	NMF	LFM
10	NMF	CFinder	MOSES	CFinder	CFinder	UEOC
11	Link	NMF	Link	Link	Link	EAGLE
12	iLCD	iLCD	EAGLE	EAGLE	EAGLE	GCE
13	EAGLE	EAGLE	UEOC	iLCD	iLCD	CIS
14	UEOC	UEOC	iLCD	UEOC	UEOC	NMF

4.3.2 Effects of Changes in the Network

We first examined how the performance, measured by NMI, changes as the number of memberships O_m varies from small to large values (i.e., 2 to 8) for different network parameters.

Results for $O_n = 10\%$ are shown in Fig. 4.3. As expected, the larger the value of μ, the poorer the performance (i.e., red curve < blue curve in Fig. 4.3) due to the fact that the connection inside communities is weak for larger μ. On the other hand, increasing network size from 1000 to 5000 typically results in slightly better performance (i.e., square > dot in Fig. 4.3). Detection performance typically decays with a moderate rate as the diversity of overlapping increases (i.e., O_m getting larger).

As shown in Fig. 4.4, the performance of detection drops in the case there are more overlapping nodes (i.e., red curves ($O_n = 50\%$) < blue curves ($O_n = 10\%$)). Given only marginal difference in performance in two tested community size ranges (curves with the same color in Fig. 4.4), it appears that the well known resolution limit does not play a role here since SLPA is neither based on a modularity nor an extended modularity. Hence, we conclude that the community size range has limited impact on SLPA.

4.3.3 Ranking for Community Detection

Extensive comparisons have been conducted over different overlapping densities ($O_n = 10\%, 50\%$) and community size ranges (small s and large b). Performances measured by both NMI and Omega for $n = 5000$ and $\mu = 0.3$ are shown in Fig. 4.5.

To provide an intuitive way for summarizing the vast volume of experiment results, we proposed $RS_M(i)$, the averaged ranking score for a given algorithm i with respect to some measure M as follows:

$$RS_M(i) = \sum_{j=1} rank(i, O_m^j), \tag{4.1}$$

where O_m^j is the number of memberships (diversity) in $\{2, 3, \cdots, 8\}$, and function *rank* returns the ranking of algorithm i for the given O_m. Sorting RS_M in increasing order gives the final ranking among algorithms. Whenever it is clear from context, we use the term *ranking* to refer to the final rank without the actual score value.

The results in Fig. 4.5 are further summarized as four rankings including RS^s_{NMI}, RS^b_{NMI}, RS^s_{Omega} and RS^b_{Omega} in Table 4.2. Note that $RS^{s/b}_{NMI/Omega}$ denotes the ranking based on NMI (or Omega) for networks with small (or large) community size range. As opposed to CIS, Link, LFM, CFinder and COPRA, which are more sensitive to the measure, SLPA is remarkably stable in NMI and Omega. Based on these four rankings, we further derived the average ranking $RS^*_{NMI,Omega}$ as the overall community detection performance. In this final ranking, SLPA stays the best, followed by three local expansion based algorithms (i.e., GCE, OSLOM and CIS).

4.3.4 Detected Community Size Distribution in LFR

In order to provide insights into the behavior of SLAP (as well as other algorithms) and verify the ranking. We looked closely at the discovered distribution (histograms) of community sizes (CS) and compared it with the known ground truth. Here we only provided analysis for $n = 5000b$, $\mu = 0.3$, $O_n = 10\%$ (the corresponding ranking is RS^b_{NMI} in Table 4.2). In this case, we expect the size distribution

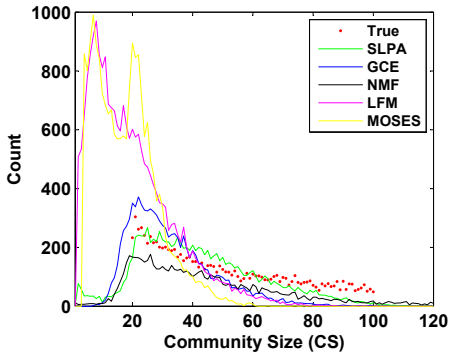

Figure 4.6: Unimodal histogram of the detected community sizes for SLPA, GCE, NMF, LFM and MOSES created from the results for LFR networks with $n = 5000$, $\mu = 0.3$ and $O_n = 10\%$.

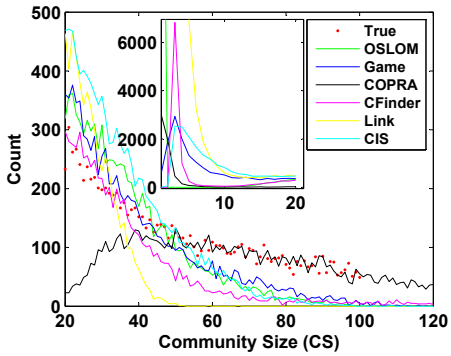

Figure 4.7: Bimodal histogram of the detected community sizes for OSLOM, Game, COPRA, CFinder, Link and CIS created from the results for LFR networks with $n = 5000$, $\mu = 0.3$ and $O_n = 10\%$.

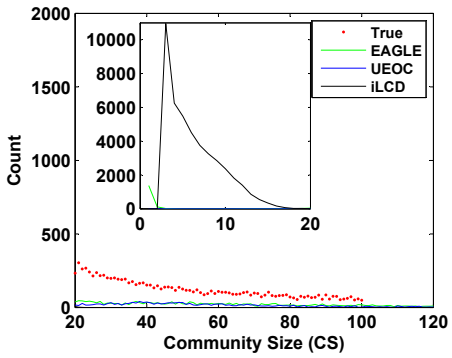

Figure 4.8: Histogram of the detected community sizes for EAGLE, UEOC and iLCD crated from the results for LFR networks with $n = 5000$, $\mu = 0.3$ and $O_n = 10\%$.

to follow the power law with exponent $\tau_2 = 1$ with the minimum size 20 and the maximum size 100. Note that the histograms are created from communities over different $O'_m s$.

As shown in Fig. 4.6, high-ranking (with high NMI) algorithms such as SLPA, GCE and NMF typically yield a *unimodal* distribution with a peak at $CS = 20$ fitting well with the ground truth distribution. In contrast, algorithms in Fig. 4.7 typically produce a *bimodal* distribution. The existence of an extra dominant mode for CS ranging from 1 to 5 results in a significant number of small size communities

in CFinder, COPRA, Link, CIS, and so on. In Figure 4.8, the distribution is shifted mostly outside the predefined range 20∼100. Algorithms with such a distribution create relatively small communities and perform poorly. Here, we conclude that observations on the community size distribution can be used to explain the ranking with respect to NMI.

4.3.5 Identifying Overlapping Nodes in LFR

Identifying nodes belonging to multiple communities is an essential component of measuring the quality of a detection algorithm. However, the node level evaluation was often neglected. Here we first look at the number of detected overlapping nodes O_n^d and detected memberships O_m^d in corresponding to the ground truth O_n and O_m. As an example, we show only results for the case of $n = 5000b$, $\mu = 0.3$, $O_n = 10\%$ (i.e., Fig. 4.5(b)) in Fig. 4.9 and Fig. 4.10 (similar results observed for other cases, not shown). Note that a value close to 1 indicates closeness to the ground truth, and values over 1 are possible when an algorithm detects more nodes or memberships than there are known to exist. As shown, SLPA yields the numbers that are close to the ground truth for O_n. As the O_m increases, all algorithms, including SLPA, fail to identify all the memberships.

Note that O_n^d alone is insufficient to accurately quantify the detection performance, as it contains both true and false positive. To provide precise analysis, we consider the identification of overlapping nodes as a *binary classification* problem. A node is labeled as *overlapping* as long as O_m>1 or O_m^d>1 and labeled as *non-overlapping* otherwise. Within this framework, we can use F-score as a measure of detection accuracy defined as

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}, \tag{4.2}$$

where *recall* is the number of correctly detected overlapping nodes divided by O_n, and *precision* is the number of correctly detected overlapping nodes divided by O_n^d. F-score reaches its best value at 1 and worst score at 0.

As shown in Figs. 4.11(a), (d), SLPA achieves the best F-score for $O_m > 2$. Moreover, it has a positive correlation with O_m while the performances of other

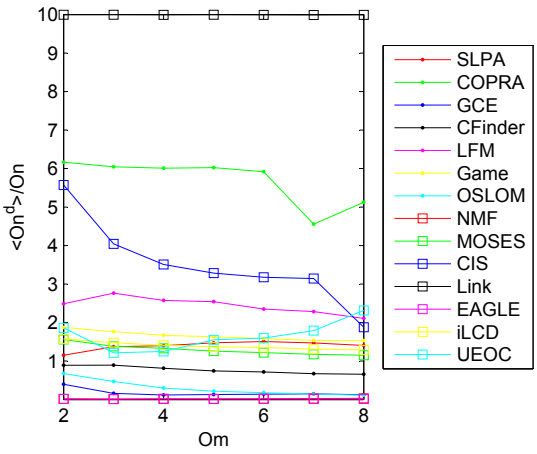

Figure 4.9: The number of detected overlapping nodes relative to the ground truth for $n = 5000b$, $\mu = 0.3$, $O_n = 10\%$.

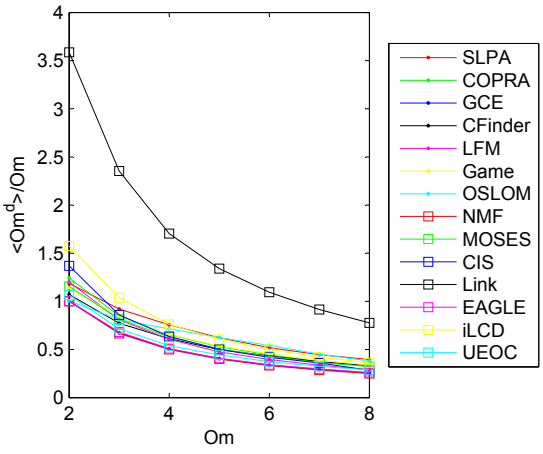

Figure 4.10: The number of memberships of detected overlapping nodes relative to the ground truth for $n = 5000b$, $\mu = 0.3$, $O_n = 10\%$.

algorithms decrease as O_m increases. That is, it is able to correctly uncover a reasonable fraction of overlapping nodes even when those nodes belong to many groups. Algorithms that fail to have a good balance between precision and recall result in low F-score, especially for EAGLE and Link. The high precision of EAGLE (also CFinder and GCE for $O_m = 2$) shows that clique-like assumption of communities may help to identify overlapping nodes. However, they under-detect the number of overlapping nodes. For Link, it has perfect recall score but low precision. This is merely due to the over-detection evidenced in Fig. 4.9. In contrast, SLPA has a good balance between quality and quantity in detecting overlapping nodes as demonstrated by relatively high precision and recall in Fig. 4.11.

4.3.6 Ranking for Overlapping Node Detection

The average ranking with respect to F-scores for different community size ranges is shown as RS^*_F in Table 4.2. It is clearly shown that the community quality ranking $RS^*_{NMI,Omega}$ and node quality ranking RS^*_F might provide quite different pictures of the performance. Algorithms with a low rank in detecting communities could actually have good performances when it comes to identifying overlapping

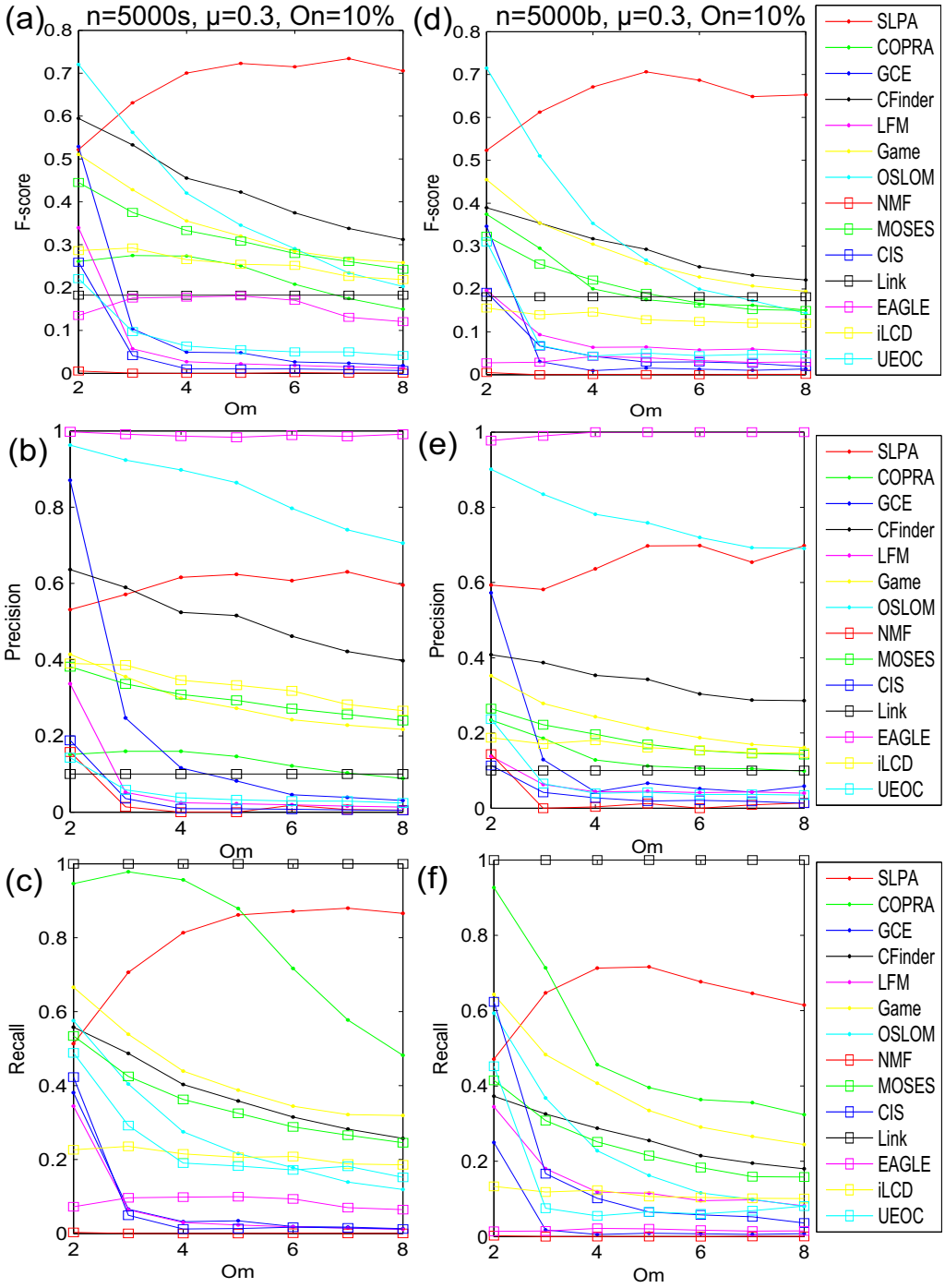

Figure 4.11: Evaluations of overlapping node detection on LFR networks with $O_n = 10\%$. Plots show F-score (together with precision and recall) as a function of the number of memberships for $n = 5000$ and $\mu = 0.3$. Results for small community size range are shown in the left column, and results for large community size range are shown in the righ column.

Table 4.3: Social networks in the tests.

Network	n	$\overline{k}$	Network	n	$\overline{k}$
karate (KR)	34	4.5	Email (EM)	33,696	10.7
football (FB)	115	10.6	P2P	62,561	2.4
lesmis (LS)	77	6.6	Epinions (EP)	75,877	10.6
dolphins (DP)	62	5.1	Amazon (AM)	262,111	6.8
CA-GrQc (CA)	4,730	5.6	HighSchool (HS1)	69	6.3
PGP	10,680	4.5	HighSchool (HS2)	612	8.0

nodes (e.g., CFinder, iLCD and MOSES), while high-ranking algorithms, including GCE and CIS, might perform badly due to under-detection and over-detection respectively. SLPA has very stable and good performance in both rankings.

To sum up, by taking both community level performance (NMI and Omega) and node level performance (F-score) into account, we conclude that SLPA performs well in the LFR benchmarks.

4.4 Tests in Real-world Social Networks

We applied SLPA to a wide range of well-known social networks[10] as listed in Table 4.3. In particular, the high school friendship networks[11] are social networks in high schools self-reported by students together with their grades, races and sexes.

4.4.1 Identifying Overlapping Communities in Social Networks

Without the ground truth on most large networks, we used the overlapping modularity Q_{ov}^{Ni} to quantify the performance, which is an extension of Newman's modularity proposed by Nicosia [113]. A high positive value indicates a significant overlapping community structure relative to the null model. We removed CFinder, EAGLE and NMF from the test because of either their memory or their computation inefficiency on large networks. As an additional reference, we added the disjoint detection results with the Infomap algorithm [114].

[10] http://www-personal.umich.edu/~mejn/netdata and http://snap.stanford.edu/data.

[11] A project funded by the National Institute of Child Health and Human Development.

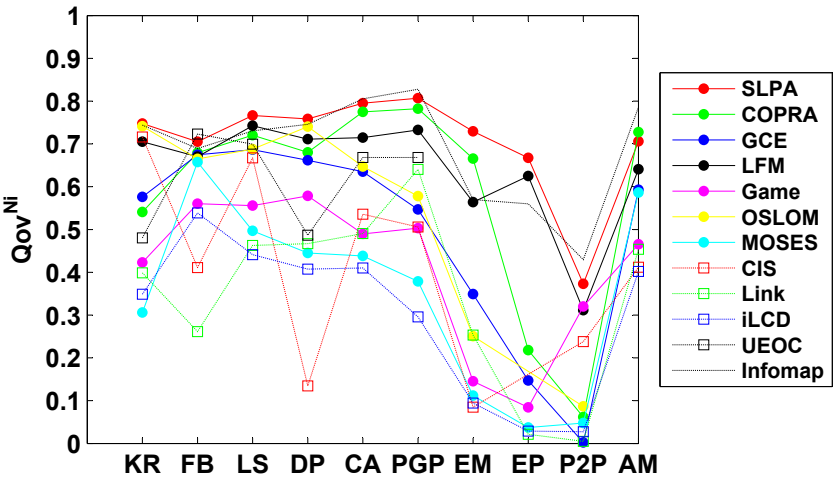

Figure 4.12: Overlapping modularity Q_{ov}^{Ni} for social networks.

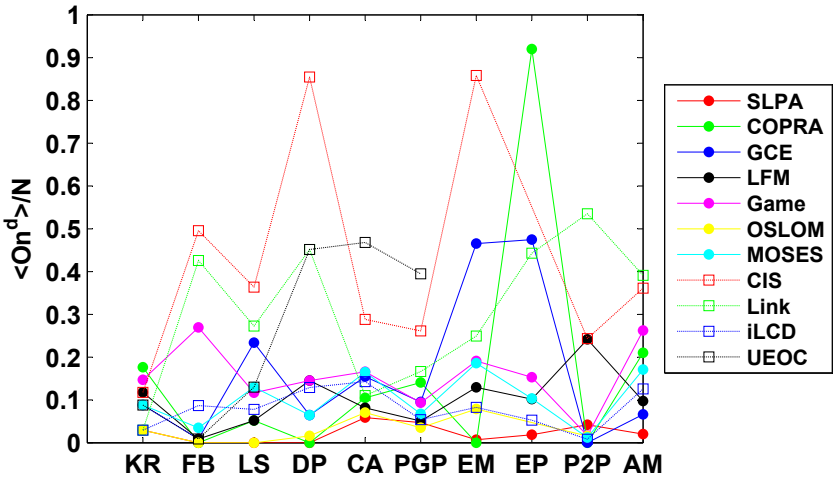

Figure 4.13: The number of detected overlapping nodes (normalized) for social networks.

As shown in Fig. 4.12, in general, SLPA achieves the highest average Q_{ov}^{Ni}, followed by LFM and COPRA. Compared with COPRA, SLPA is applicable to more networks. In contrast, COPRA does not work well on highly sparse networks such as *P2P*, for which COPRA finds merely one single giant community. COPRA also fails on *Epinions* network claiming too many overlapping nodes as shown in Fig. 4.13 (over-detection also applies to CIS and Link, resulting in low Q_{ov}^{Ni} scores for these two algorithms as well). The results in Figs. 4.13 and 4.14 (based on the clustering with the best Q_{ov}^{Ni}) show a common feature in the tested real-world

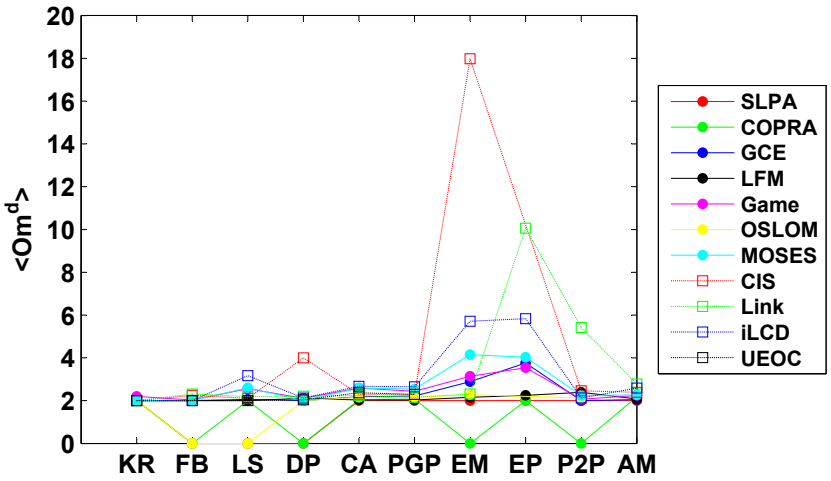

Figure 4.14: The number of detected memberships for social networks.

networks, upon which most algorithms agree. That is, the relatively small overlap in both the fraction of overlapping nodes (typically less than 30%) and the number of communities of which an overlapping node is a member (typically 2 or 3).

As known, a high modularity might not necessarily result in a *true* partitioning as it does in the disjoint community detection. We used the high school network (HS1) with known attributes to verify the output of SLPA. As shown in Fig. 4.15, there is a good agreement between the found and known partitions in term of student's *grades*. In SLPA, the grade 9 community is further divided into two subgroups. The larger group contains only white students, while the majority in the smaller group are black students. These two groups are connected partially via an overlapping node. All overlapping nodes (a total of 4) only exist on the boundaries of communities. Two of them are assigned to three communities, while the other two are assigned to two communities.

4.4.2 Identifying Overlapping Communities in Bipartite Networks

Discovering communities in bipartite networks is important because they provide a natural representation of many social networks. One example is the online tagging system with both users and tags. Unlike the original LPA algorithm, which performs poorly on bipartite networks, SLPA works well on this kind of networks.

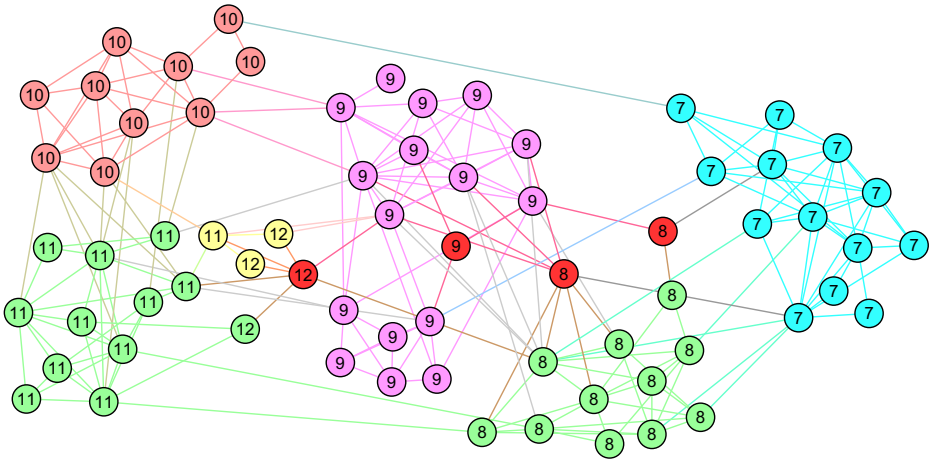

Figure 4.15: High school network ($n = 69$, $\overline{k} = 6.4$). Labels are the known grades ranging from 7 to 12. Colors represent communities discovered by SLPA. The overlapping nodes are highlighted in dark red.

We demonstrate this using two real-world networks[12]. One is a Facebook-like social network. One type of nodes represents users (abbr. FB-M1), while the other represents messages (abbr. FB-M2). The second network is the interlocking directorate. One type of nodes represents affiliations (abbr. IL-M1), while the other individuals (abbr. IL-M2).

We compared SLPA with COPRA in Table 4.4 (together with the standard deviation (std) in the Q_{ov}^{Ni}). One difference between SLPA and COPRA is that SLPA applies to the *entire* bipartite network directly, while COPRA is applied to each type of nodes *alternatively*. Q_{ov}^{Ni} is computed on the *projection* of each type of nodes. Although COPRA is slightly better (by 0.03) than SLPA on the second type of nodes for interlock network, it is much worse (by 0.11) on the first type. Moreover, COPRA fails to detect meaningful communities in the Facebook-like network, while SLPA demonstrates relatively good performance.

4.4.3 Identifying Overlapping Nested Communities

In the above experiments, we applied a post-processing to remove subset communities from the raw output of stages 1 and 2 of SLPA. This may not be necessary

[12]Data are available at http://toreopsahl.com/datasets.

Table 4.4: The Q_{ov}^{Ni} of SLPA and COPRA for two bipartite networks.

Network	n	SLPA (std)	COPRA (std)
FB-M1	899	0.23 (0.10)	0.02 (0.07)
FB-M2	522	0.36 (0.02)	0.02 (0.07)
IL-M1	239	0.59 (0.02)	0.48 (0.02)
IL-M2	923	0.69 (0.01)	0.72 (0.01)

for some applications. Here, we show that rich *nested* structure can be recovered in the high school network (HS2) with $n = 612$. The hierarchy is shown as a treemap[13] shown in Fig. 4.16. To evaluate the degree to which a discovered community matches the known attributes, we define a *matching score* as the *largest* fraction of matched nodes among three attributes (i.e., grade, race and sex). The corresponding attribute is said to best explain the community found by SLPA.

The community name shows the full hierarchy path (connected by a dash) leading to this community. For example, *C1* has id 1 and is located on the first level, while *C1-25* has id 25, and it is the second level sub-community of community *C1*. As shown, SLPA discovers a tree with a height of four. Most of the communities are distributed on the first two levels.

Nested structures are found across different attributes. For example, *C13* is best explained by *race*, while its two sub-communities perfectly account for *grade* and *sex* respectively. In *C1*, sub-communities explained by the same attribute account for *different* attribute *values*. For example, both *C1-25* and *C1-40* are identified by *sex*. However, the former contains only *male* students, while in the latter *female* students are the majority. Although the treemap is not capable of displaying overlaps between communities, overlapping structure presents in the nested communities as well.

4.5 Summary

We introduced a dynamic interaction process, SLPA as a basis for an efficient (linear time) and effective overlapping community detection algorithm. SLPA allows us to analyze different kinds of community structures, such as disjoint com-

[13]Treemap is used for visualization: www.cs.umd.edu/hcil/treemap.

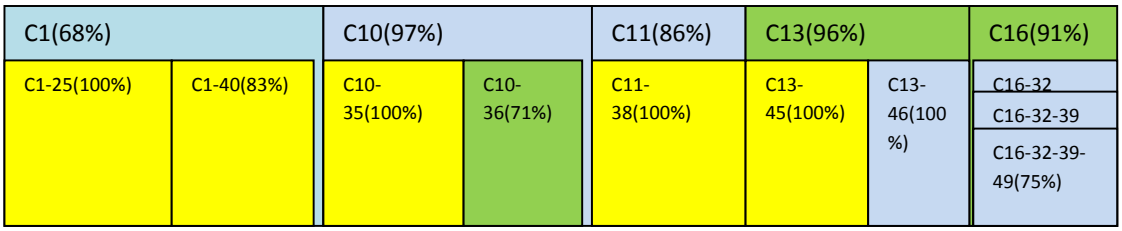

Figure 4.16: The nested structure in the high school network represented as a Treemap. The color represents the best explaining attribute: blue for *grade*; green for *race*; and yellow for *sex*. Numbers in parenthesis are the matching scores defined in the text. The size of shapes is proportional to the community size. Due to the page limit, only part of the entire treemap is shown.

munities, individual overlapping nodes, overlapping communities and overlapping nested hierarchy in both unipartite and bipartite topologies. Although the tests are for undirected unweighted networks, we have implemented the version for directed and weighted networks. The codes are available online[14]. More analyses (e.g., $O_n = 50\%$) are also available in the survey paper [40].

[14]https://sites.google.com/site/communitydetectionslpa.

CHAPTER 5
LABELRANKT: A DECENTRALIZED ONLINE ALGORITHM FOR DETECTION OF EVOLVING COMMUNITIES

In most real-world applications, the network structure is evolving over time, including adding and deleting nodes and edges. The embedded communities are expected to evolve as well. With the rapid emergence of large-scale online social networks, e.g., 900 millions users on Facebook as of 2012, there is a high demand for an efficient community detection algorithm that is able to handle large amount of new data on a daily basis. For instance, more than 250 million photos are uploaded to Facebook every day. Thus, an algorithm that can take advantage of the cutting-edge cloud computing (e.g., EC2 [115]) or the ubiquitous mobile computing [116] (e.g., a human-centralized mobile phone network or an ad hoc wireless sensor network) would be valuable for future applications. This essentially requires that a detection algorithm is parallelizable or distributed*.

Despite the ambiguity in the definition of *community*, numerous techniques have been developed for community detection. However, most community detection algorithms rely on a *global* view of the network, structure information other than local connections is vital for these algorithms. An example is the MCL algorithm [117], which is based on the multiplication of the transition matrix to simulate random walks on the network. The computation is centralized limiting its scalability in large-scale networks. Another challenge is that most of the existing algorithms assume a *static* view of the network, which ignores the temporal correlation between different snapshots over time. Such algorithms are less suitable for a dynamically evolving network, especially in the case where new data come in continuously.

In this chapter, we seek solutions to meet these two needs: online and decen-

*Portions of this chapter previously appeared as: J. Xie and B. K. Szymanski, LabelRankT: A decentralized online algorithm for detection of evolving communities, CIKM (under review), 2012 [87].

tralized detection. Label propagation based community detection algorithms such as LPA [72,81,110] and SLPA [86] have been shown to perform well detecting either disjoint or overlapping communities [40]. Requiring only *local* information, they can serve as good candidates for a parallel or distributed detection. Yet, the extension for dynamic networks has not been fully explored so far. However, the use of random tie breaking strategy makes label propagation based algorithms nondeterministic in nature. They produce different partitions in different runs. Such randomness is particularly undesired when tracking the evolution of communities in a dynamic network.

The contributions of this chapter are two-fold. First, we propose strategies to stabilize the LPA. We introduce a generalized and stable version of LPA, called *LabelRank*, that produces deterministic partitioning and significantly improves the quality of detected communities. Then, we extend LabelRank to online (or incremental) community detection in dynamic networks.

5.1 LabelRank Algorithm

LabelRank is based on the idea of simulating the propagation of labels in the network. Here, we use node ids as labels. LabelRank stores, propagates and ranks labels in each node. Rather than keeping only one label, during the simulation, for each node, LabelRank keeps multiple labels received from its neighbors. This eliminates the need of tie breaking in LPA [81] or COPRA [72] (e.g., between multiple labels with the same maximum size or with the same probability). Nodes whose highest probability labels are the same form a community. Since there is no randomness in the simulation, the output is deterministic. LabelRank relies on four operators applied to the labels including propagation, inflation, cutoff and conditional update.

5.1.1 Label Operators

Propagation: In each node, an entire distribution of labels is maintained and spread to neighbors as in [72]. It is defined as an $1 \times n$ vector P_i (n is the number of nodes), and each element $P_i(c)$ or P_{ic} is interpreted as the probability

of node i observing label $c \in C$ taken from a finite alphabet C. Here, we assume that $C = \{1, 2, \ldots, n\}$ is the same as the set of node ids, so $|C| = n$ for simplicity. This assumption will be lifted later (see Section 5.1.2). In LabelRank, each node broadcasts the distribution to its neighbors at each time step and computes the new distribution $P_i^{'}$ simultaneously using the following equation:

$$P_i^{'}(c) = \sum_{j \in Nb(i)} P_j(c)/k_i, \forall c \in C, \tag{5.1}$$

where $Nb(i)$ is a set of neighbors of node i and $k_i = |Nb(i)|$ is the number of neighbors. Note that $P_i^{'}$ is normalized to represent a probability distribution.

The broadcasting (copying) and computing new distribution is at the heart of this operator, which can be put in a matrix form as

$$A \times P, \tag{5.2}$$

where A is the $n \times n$ adjacency matrix and P is the $n \times n$ (by the above assumption) *label distribution matrix*, which is the assemble of all P_i's. To initialize P, we start with the setting in which every node has equal probability to see each neighbor, so

$$P_i(c) = 1/k_i, \forall c \text{ s.t. } A_{ic} = 1. \tag{5.3}$$

To spread labels, a plausible attempt is to update P in Eq. 5.2 as $P = A \times P$ until convergence. In this way, any label would be observed by any node in the network within at most *dim* steps, where *dim* is the diameter of the network. However, this would not work because in the limit of time, all P_i's *converge* to the same stationary distribution in most networks. This is because the metric space of A is usually compact, which implies the existence and uniqueness of a fixed point by the Banach fixed point theorem [118]. Given such a distribution, there is no good way to group nodes to form communities. We need more operators to trap the process in some local optimum of the quality metric (e.g., modularity Q [119]) without propagating too far.

Inflation: As in MCL [117], we use the inflation operator Γ_{in} on P to contract

the propagation, where $in \in \Re^+$. In contrast to MCL, to decouple inflation from the network structure, we apply it to the label distribution matrix P rather than to a stochastic matrix or adjacency matrix. After operation $\Gamma_{in}P$ (Eq. 5.4), each $P_i(c)$ is proportional to $P_i(c)^{in}$, i.e., $P_i(c)$ rises to the in^{th} power.

$$\Gamma_{in}P_i(c) = P_i(c)^{in} / \sum_{j \in C} P_i(j)^{in}. \tag{5.4}$$

This operator allows a label with higher probability to be even more preferred (in a nonlinear fashion) and increases its chance to be shared around the neighborhood. For example, considering a distribution with two labels a and b, $P_i(a) = 0.6$ and $P_i(b) = 0.4$, $\Gamma_{in=2}$ on P_i gives $P_i(a) = 0.6923$ and $P_i(b) = 0.3077$.

In our tests, this operator helps to form local subgroups. However, it alone does not provide satisfying performance in large networks. Moreover, the memory inefficiency problem implied by Eq. 5.2, i.e., n^2 labels stored in the networks, is not yet fully resolved by the inflation operator.

Cutoff: To further trap the process in some local optimum and alleviate the memory problem, we introduce cutoff operator Φ_r on P to remove labels that are below threshold $r \in (0, 1)$. As expected, Φ_r also serves to constrain the label propagation similarly to the inflation that in a sense resembles a *nonlinear* cutoff. More importantly, Φ_r efficiently reduces the space complexity, from quadratic to linear. For example, with $r = 0.1$, the average number of labels in each node is typically less than 3.0 implying $O(n)$ labels in the network. Note that the case where no labels have probabilities higher than r could happen. In such a case, we keep *all* the labels with the maximum probability.

Conditional Update: Using the above three operators, we have a process that finds communities with high quality. The problem is that the process typically demonstrates a *continuous* decay in performance after hitting a maximum performance (green curves in Fig. 5.1). The algorithm does not know when the best partitioning has been already obtained and therefore when is the best time to stop.

Note that in Fig. 5.1, we compute $Q(t)$ for each step explicitly. This is ONLY for illustrating the problem. Computing Q is not supposed to be part of the algo-

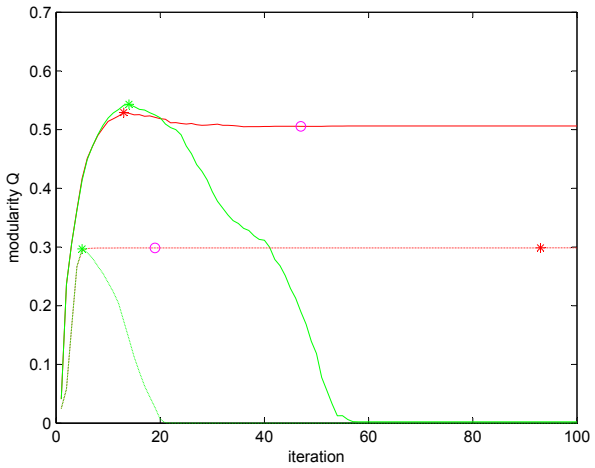

Figure 5.1: The effect of conditional update operator. The plot shows the modularity Q over iterations on the email network [120] with $n = 1,133$ (two curves on the top) and wiki network [121] with $n = 7,066$ (two curves at the bottom). Each Q is computed explicitly for each iteration. Green curves are based on three operators Propagation+Inflation+Cutoff. Red curves are based on four operators Propagation+Inflation+Cutoff+Conditional Update. Asterisk indicates the best performance of $Q(t)$. Purple circle indicates Q when the stop criterion described in the main text is applied.

rithm, and it is undesired because it takes $O(n^2)$ time. Given only the network state, i.e., nodes with label distributions, finding a statistic that indicates the partitioning quality during the simulation is hard, if not impossible (we did try a few with no success). To solve this problem, we propose the conditional update operator Θ_q on both P' and P. Instead of blindly accepting the new distribution P_i', only nodes satisfying the following updating condition computed using P from the *previous* time step accept the change at each iteration:

$$\sum_{j \in Nb(i)} isSubset(C_i^*, C_j^*) \leq qk_i, \tag{5.5}$$

where C_i^* contains the *maximum labels* that are labels with the maximum probability in P_i at the *previous* time step, function $isSubset(s_1, s_2)$ returns 1 if $s_1 \subseteq s_2$, 0 otherwise, k_i is the degree of node i, and q is a parameter in $[0, 1]$. Intuitively, isSubset can be viewed as a measure of *similarity* between two nodes in terms of

labels they have. Equation 5.5 requires a node to update only if it is NOT similar to at least qk_i neighbors. In other words, a node remains unchanged when it is similar to more than a fraction q of neighbors. Note that a node deciding not to change at the current step may continue to update later due to the change in its neighbors. As shown in Fig. 5.1, Θ_q operator successfully traps the process in the modularity space of high quality, indicated by a long-lived plateau in the modularity curve (red curves). Intuitively, Equation 5.5 augments the stability of the label propagation.

Stop criterion: Thanks to the conditional update operator, determining a moment when it is safe to terminate is possible and defined as follows. We track the number of nodes that change, *numChange*, at each iteration and accumulate the number of repetitions *count*(*numChange*) in a table. One obvious stop signal is when *numChange* drops to zero, but it may never happen if the propagation gets into a cycle iterating over the same sequence of configurations. Accordingly, the algorithm stops when the *count* of any *numChange* first exceeds five times (by looking up the table), or no change happened in this iteration (i.e., numChange=0). Although such criterion does not guarantee the best performance, it almost always returns satisfying results. As illustrated on two networks in Fig. 5.1, the difference between the found Q (purple circles) and maximum Q (red asterisks) is small. This criterion does not guarantee the optimal running time neither. The time when our algorithm stops may be either earlier or later than when the maximum performance occurs. However, LabelRank is still very efficient as shown later.

Algorithm 2 : LabelRank

1: add selfloop to adjacency matrix A
2: initialize label distribution P using Eq. 5.3
3: **repeat**
4: $\quad P' = A \times P$
5: $\quad P' = \Gamma_{in} P'$
6: $\quad P' = \Phi_r P'$
7: $\quad P = \Theta_q(P', P)$
8: **until** stop criterion satisfied
9: output communities based on P

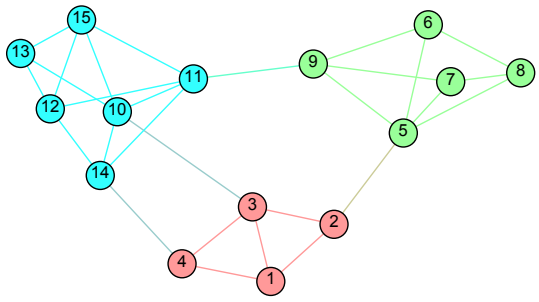

Figure 5.2: The sample network G(0) with $n = 15$. Colors represent communities discovered by LabelRank (see Table 5.1) with cutoff $r = 0.1$, inflation $in = 4$, and conditional update $q = 0.7$. The algorithm stopped at the 7^{th} iteration. The average number of labels dropped from 2.933 to 1.2 during the simulation.

Table 5.1: The resultant P on the sample graph G(0).

0.279	-	**0.721**	-	-	-	-	-	-	-	-	-	-	-	-
-	-	**1**	-	-	-	-	-	-	-	-	-	-	-	-
-	-	**1**	-	-	-	-	-	-	-	-	-	-	-	-
-	-	**1**	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	**1**	-	-	-	-	-	-	-	-	-	-
-	-	-	-	**1**	-	-	-	-	-	-	-	-	-	-
-	-	-	-	**1**	-	-	-	-	-	-	-	-	-	-
-	-	-	-	**1**	-	-	-	-	-	-	-	-	-	-
-	-	-	-	**1**	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	**1**	-	-	-	-
-	-	-	-	-	-	-	-	-	-	**1**	-	-	-	-
-	-	-	-	-	-	-	-	-	-	**1**	-	-	-	-
-	-	-	-	-	-	-	-	-	0.203	**0.797**	-	-	-	-
-	-	-	-	-	-	-	-	-	-	**1**	-	-	-	-
-	-	-	-	-	-	-	-	-	0.126	**0.874**	-	-	-	-

5.1.2 Implementation

The complete Algorithm 2 consists of an iteration over the four operators followed by post-processing that groups nodes whose highest probability labels are the same into a community. A sample network and the output of LabelRank are shown in Fig. 5.2 and Table 5.1. There are only 1.2 labels on average in each node, which results in a very sparse label distribution (Table 5.1). Three communities are identified with red community defined by label 3 (third column), green community defined by label 5 and blue community defined by label 11.

In the previous analysis, we used the matrix form and made the assumption that the length of P_i is set to n resulting in a $n \times n$ P matrix. In the implementation, this is not necessary due to both cutoff and inflation operators. In general, the number of labels in each node monotonically decreases and is much smaller than n after a few steps. The $n \times n$ P matrix is actually replaced by n variable-length vectors (usually short) carried by each node (as illustrated in Table 5.1). Also, the best value of cutoff threshold r is not sensitive and is typically set to 0.1, which removes one dimension of freedom in the parameter space.

For the initialization of P, we tried different configurations, e.g., setting $P_{ij} = 1/k_j$, but it appears not to be very important for large networks. Similarly, replacing A with other matrices, e.g., the stochastic matrix, does not improve the performance.

It turns out that the preprocessing that adds a selfloop to each node (i.e., $A_{ii} = 1$) helps to improve the detection quality. Although we do not have a proof, we believe that the selfloop effect resembles the impact of a lazy walk in a graph that avoids the periodicity problem, because the selfloop can smooth the propagation (update of P_i) by taking its own label distribution into account. Technically, each node will consider itself as a neighbor in Eq. 5.1.

There is a subtle tie we need to deal with. At the end of propagation, there might be more than one label with the maximum probability. However, since the tie is delayed until the last step, chance of it arising is extremely small. In the case it does happen, we select the label with the minimum id to guarantee a deterministic output.

The difference between LabelRank and MCL is worth further elaboration. In both algorithms, there is a matrix multiplication, $A \times P$ for LabelRank and $M \times M$ for MCL (M is the $n \times n$ stochastic matrix). For updating an element, both $P_{ij} \leftarrow A_{i.} \times P_{.j}$ and $M_{ij} \leftarrow M_{i.} \times M_{.j}$ look seemingly equally efficient ($O(n)$ operations), where $i.$ is the i^{th} row and $.j$ is the j^{th} column. However, since A represents the static network structure, there is no need to carry out any operation associated with zero value elements for LabelRank. More specifically, the number of effective operations for each node is known and defined by k_i, the number of neighbors. Thus, the time for computing the P_{ij} is $O(k_i)$. Supposed there are

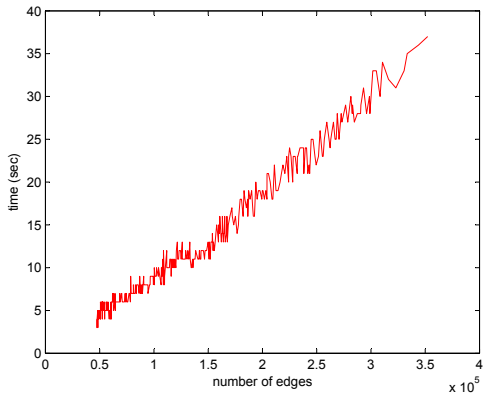

Figure 5.3: The execution times on a set of Arxiv high energy physics theory citation graphs (see Section 5.4.1) with n ranging from 12,917 to 27,769 and m from 47,454 to 352,285. Tested on a desktop with Intel@2.80GHz.

x labels (typically less than 3) in each node on average. Updating one row P_i requires $O(xk_i)$ steps. As a result, the time for updating the entire P in LabelRank is $O(x\overline{k}n) = O(xm) = O(m)$, where $\overline{k}$ is the average degree, and m is the total number of edges. In contrast, during the expansion (before convergence), M_{ij} of M that rises to power larger than 1 is changing by the definition of transition matrix of a random walk, i.e., a zero or nonzero values in M_{ij} no longer reflects the network connections in one hop. Therefore, the computation of M_{ij} may require non-local information and the time is $O(n)$, which leads to $O(n^3)$ for the entire $M \times M$ operator in worst case (although it is usually much faster than this in sparse networks). In conclusion, the propagation scheme in LabelRank is highly parallel and allows the computation to be distributed to each individual node. Such property will be particularly useful for highly distributed and self-organizable applications like wireless sensor networks [122] and mobile ad hoc networks (MANET) [123], where each node in the network corresponds to a physical platform.

The running time of LabelRank is $O(m)$, linear with the number of edges m because adding selfloop takes $O(n)$, the initialization of P takes $O(m)$, each of the four operators takes $O(m)$ on average and the number of iterations is usually around 10. Note that although sorting the label distribution is required in conditional update, it takes effectively linear time because usually there are no more than three

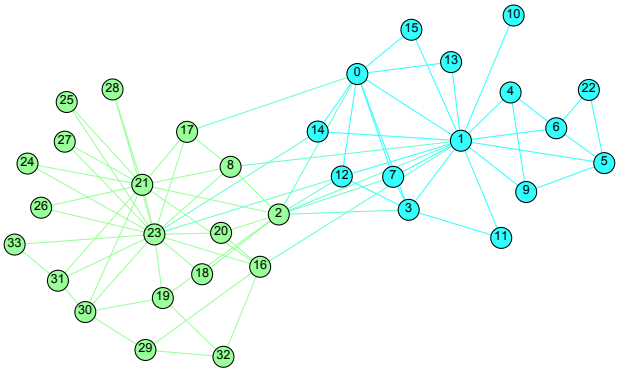

Figure 5.4: Communities detected on the Zachary's karate club network. The ground truth of this network is two clusters centered on node 1 and 23 respectively. The communities found by LabelRank are highlighted with colors, which show a perfect matching with the ground truth.

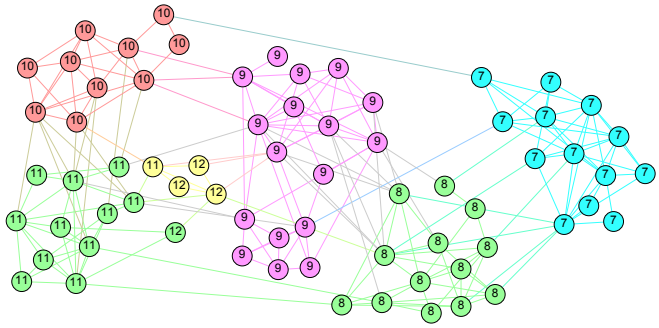

Figure 5.5: Communities detected on a High school friendship network ($n = 69$, $\overline{k} = 6.4$). Labels are the known grades ranging from 7 to 12. Colors represent communities discovered by LabelRank.

labels to sort. The execution times on a set of citation networks that verify the linear scaling are shown in Fig. 5.3. The test is on a single computer, but we expect further improvement on a parallel platform.

5.2 Evaluation on Static Networks

Despite the existence of synthetic network generators like LFR benchmark [111], in our tests, we used solely real-world networks. This is because in our experience algorithms may perform well on well-defined synthetic networks but not on real-world networks.

We first verified the quality of communities reported by our algorithm on

Table 5.2: The modularity Q's of different community detection algorithms.

Network	n	LPA	LabelRank	MCL	Infomap
Football	115	0.60	0.60	0.60	0.60
HighSchool	1,127	0.66	0.66	0.60	0.58
Eva	4,475	-	0.89	0.89	0.89
PGP	10,680	0.63	0.81	0.80	0.81
Enron Email	33,696	0.31	0.58	0.48	0.53
Epinions	75,877	-	0.34	0.27	0.38
Amazon	262,111	0.73	0.76	0.76	0.77

networks for which we know the true grouping. Fig. 5.4 shows the result on the classical Zachary's karate club network [124] with $n = 34$. It is known that this network has two clusters centered on the teacher and manager (node 1 and 23) respectively. LabelRank discovers exactly these two communities with $Q = 0.37$.

We also applied LabelRank to a set of high school friendship networks [15]. In addition to the connections, each student self-reported his grade, race and sex, which provide us with rich information for verification. The results on these networks are similar. Hence, we presented only one instance with $n = 69$ in Fig. 5.5. LabelRank reports six groups of students. As shown, there is a good agreement between the found and known partitions in terms of *grades* ranging from 7 to 12. The only exception is that one node from grade 12 is grouped into grade 11, and one node from grade 11 is put into grade 12. This is due to the fact that the group of grade 12 itself is loosely defined as compared to the others. With this partitioning, the modularity is found to be $Q = 0.59$ indicating a strong community structure.

Second, we tested LabelRank on a wider range of large social networks[16] and compared with other known algorithms. Since in LabelRank all nodes update simultaneously, the baseline is the LPA with synchronous update [81]. The other algorithm which we included in comparison is MCL that uses a similar inflation operator. We also included Infomap algorithm [114] that is reported to be highly efficient in the review paper [33] and represents the state of the art in our opinion. Since the output of LPA is nondeterministic, we repeated the algorithm 10 times

[15] A project funded by the National Institute of Child Health and Human Development.

[16] snap.stanford.edu/data and www-personal.umich.edu/~mejn/netdata.

and reported the best performance. For MCL, the best performance with inflations in the range of [1.5, 5] is shown. For LabelRank, q is 0.5 or 0.6, while in is from the set $\{1, 1.5, 2\}$. Due to the lack of knowledge of true partitioning in most networks, we used modularity [120] as the quality measure. Discussions of this measure can be found in [125, 126]. The detection results are shown in Table 5.2.

As shown, LPA works well on only two networks with relatively dense average connections ($\overline{k} \approx 10$) including football and HighSchool networks. In general, it performs worse than the other three algorithms. However, with the stabilization strategies introduced in this chapter, LabelRank, a generalized and stable version of LPA, boosts the performance significantly, e.g., with an increase of 28.57% on PGP and 87.1% on Enron Email. More importantly, LPA suffers from the fact that it might easily lead to a trivial output (i.e., a single giant community). For instance, it completely fails on Eva and Epinions. The conditional update in LabelRank appears to provide a way to prevent such undesired output. As a result, LabelRank allows label propagation algorithms to work on a wider range of network structures, including both Eva and Epinions.

LabelRank outperforms MCL significantly on HighSchool, Epinions and Enron Email by 10%, 20.83% and 25.93% respectively. This provides an evidence, at least to some degree, that there is an advantage of simulating label propagation from every node over a simple random walk as in MCL. LabelRank and Infomap have close performance. LabelRank outperforms Infomap on HighSchool and Email by 10.34% and 9.43% respectively, while Infomap outperforms LabelRank on Epinions by 11.76%.

5.3 LabelRankT: An Extension for Dynamic Networks

In a dynamic network, two consecutive snapshots may be different from each other due to the changes in the network structure, as nodes and edges can be added or removed. Here, we simplify by considering only changes to a node, because a change of an edge always imposes a change on two end-nodes. The extended algorithm, called *LabelRankT*, is described in Algorithm 3, which is based on the LabelRank introduced in the previous section.

Algorithm 3 : LabelRankT

input: snapshots $G([0, 1, \cdots, T])$
run LabelRank on $G(0)$ to produce P^0
for t=1:T **do**
(a) Identify the *changed* nodes in $G(t)$ due to the changes in edges to which they are attached since $t-1$.
(b) Initialize P^t. For node i that does not change since $t-1$, copy its label distributions, i.e., set $P_i^t = P_i^{t-1}$. For *changed* nodes, reinitialize their label distributions as in LabelRank.
(c) Iteratively update only *changed* nodes' label distributions and assign them to the corresponding communities as in LabelRank.
end for

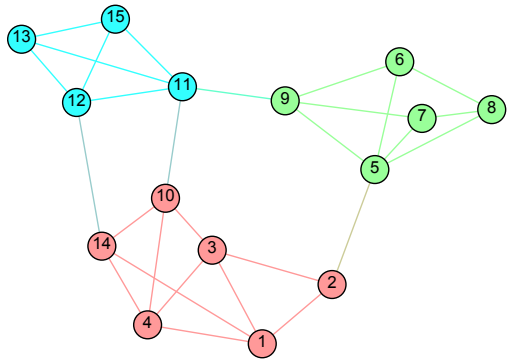

Figure 5.6: The sample network G(1) with n = 15. Since G(0), two evolutionary events happened simultaneously: splitting and merging. Two nodes (10 and 14) separated from blue group in G(0) and now merged into the red group. Three edges were deleted and three new edges were added to the network. Colors represent communities discovered by LabelRank after these events.

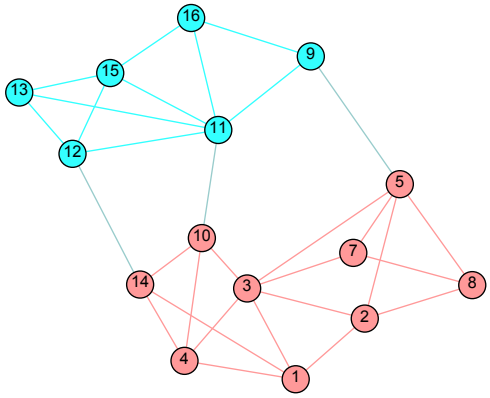

Figure 5.7: The sample network G(2) with n = 15. Since G(1), four evolutionary events happened simultaneously: birth, death, merging and dissolving. Node 6 was removed and node 16 was born. The green group dissolved and its members were merged into blue and red groups resulting in two larger communities. Colors represent communities discovered by LabelRank.

As in other work [127–129], the only assumption we make is that the evolution between two snapshots is *smooth*. This implies that some parts of the network stay unchanged between steps, and the information (e.g., our label distributions) we obtained in the previous snapshot could be still helpful in the current time step.

In our extension, the smoothing can be implemented easily by copying the label distributions for nodes that appear in both snapshots.

By our assumption, we only need to update nodes that are changed between two consecutive snapshots (called *changed nodes*). This includes cases that an existing node adds links or deletes links, or a node is removed from or newly added to the network. In our algorithm, all these cases are handled by simply comparing neighbors of a node at snapshots $t-1$ and t, i.e., $Nb^t(i)$ and $Nb^{t-1}(i)$. For *changed* nodes, we reinitialize and then update their label distributions (i.e., P^t) until the simulation stops as in LabelRank.

LabelRankT can be view as a LabelRank with one extra condition update rule in which only changed nodes accept the new distribution. However, since only changed nodes and their neighbors are involved (some neighbors only propagate labels but do not update them), LabelRankT is more efficient than LabelRank. To see this, supposed there are l changed nodes which have a total of h unique neighbors (including changed nodes themselves), then the *effective* size of A is $l \times h$ and *effective* size of P^t is $h \times n$.

LabelRankT also provides a natural way to trace a community over time. This is because due to the stabilization, LabelRankT results in deterministic outcome for a given graph. Consequently, regions that are unchanged will possess the same labels, provided that only new nodes, or nodes that changed connections can be potentially relabeled. Thus, nodes that are not relabeled provide the cores of communities that evolved from communities in the previous snapshot. Specifically, if a community is labeled with label X, its *successor* must also be labeled with label X if the evolution is smooth.

An example that shows different evolutionary events [130] in three consecutive snapshots, G(0), G(1) and G(2) is shown in Figs. 5.2, 5.6 and 5.7. During the evolution, nodes (edges) are added or removed. Communities split, merge and dissolve, which are captured by LabelRankT. The same parameters as in Fig. 5.2 are used through all snapshots.

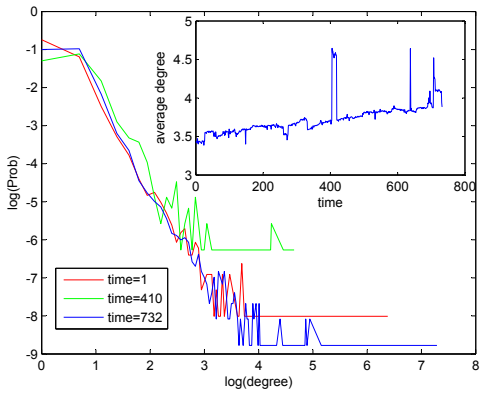

Figure 5.8: AS-Internet Routers Graph. Degree distributions at the beginning, middle and the end of evolution (main plot). Average degree over time (inset).

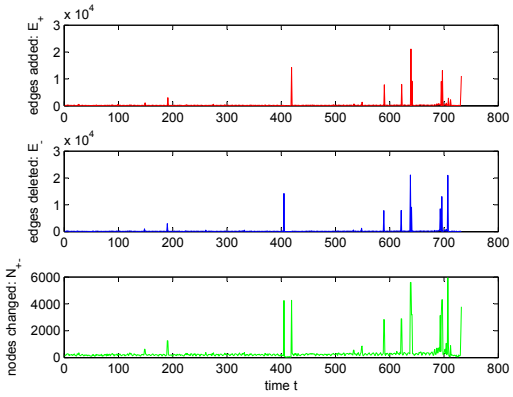

Figure 5.9: The structure changes in AS-Internet Routers Graph, including the numbers of edges added (E_+), edges deleted (E_-) and nodes involved in changes (N_{+-}).

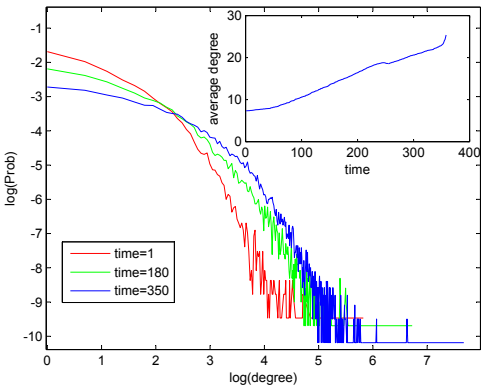

Figure 5.10: Arxiv HEP-TH. Degree distributions at the beginning, middle and the end of evolution (main). Average degree over time (inset).

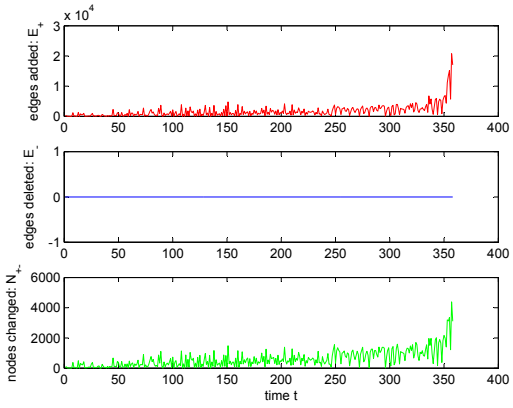

Figure 5.11: The structure changes in Arxiv HEP-TH, including the numbers of edges added (E_+), edges deleted (E_-) and nodes involved in changes (N_{+-}).

5.4 Evaluation on Dynamic Networks

We tested the detection quality of LabelRankT in terms of modularity and efficiency in two real-world datasets.

5.4.1 Datasets

AS-Internet Routers Graph [131]. This is a communication network of who-talks-to-whom from the Border Gateway Protocol logs of routers in the Internet. The data were collected by the University of Oregon Route Views Project. The dataset contains 733 daily snapshots which span a period of 785 days from November 8, 1997 to January 2, 2000. The number of nodes in the largest snapshot is 6,477 (with 13,233 edges). The nodes and edges are added or removed over time. The structure at each snapshot could change dramatically indicated by fluctuations in the average degree in Fig. 5.10 and changed statistics in Fig. 5.9.

Arxiv HEP-TH. High energy physics theory citation graph is from the e-print arXiv and covers all the citations within a dataset of 27,769 papers with 352,285 edges. If a paper i cites paper j, the graph contains a directed edge from i to j. The data cover papers in the period from January 1993 to April 2003 representing essentially the complete history of the HEP-TH section [131]. The data were originally released as a part of 2003 KDD Cup [132]. Different from AS networks, it grows monotonically over time (i.e., edges and nodes are only added, never deleted).

In our experiment, we made edges unidirectional. For a paper that does not have a time stamp, we inferred its published date as the middle between the publication time of the latest paper it cites and the publication time of the first paper that cites it. The dataset is separated into snapshots by week (a total of 359 snapshots). The first snapshot consists of papers published before 1993. The number of nodes in the t^{th} snapshot, $n(t)$, ranges from 12,917 to 27,769, and the number of edges, $m(t)$, ranges from 47,454 to 352,285. The structures over time are similar with a monotonical increase in average degree from 8 to 26 (see Fig. 5.10). The changed statistics over time are shown in Fig. 5.11.

5.4.2 Analysis

We first compared the performance of LabelRankT with static algorithms MCL[17] and Infomap, which run through each snapshot independently. Since a

[17]We used the parameter that results in the best performance for the first snapshot and applied it to the rest of snapshots.

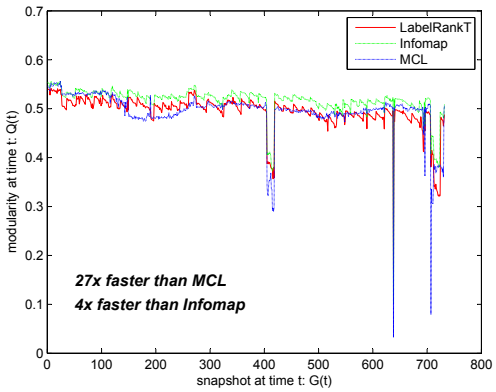

Figure 5.12: Comparison of modularity over time, $Q(t)$, with static detection algorithms on AS-Internet Routers Graph.

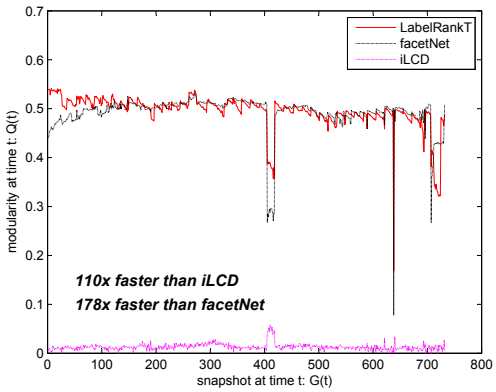

Figure 5.13: Comparison of modularity over time, $Q(t)$, with dynamic detection algorithms on AS-Internet Routers Graph.

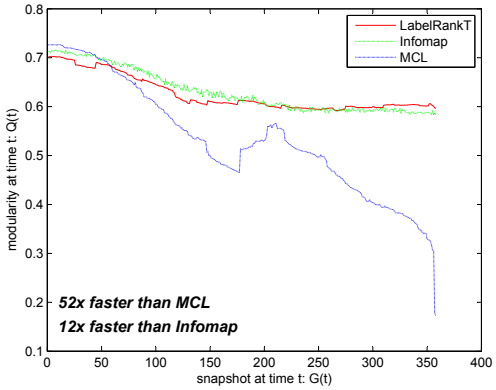

Figure 5.14: Comparison of modularity over time, $Q(t)$, with static detection algorithms on Arxiv HEP-TH.

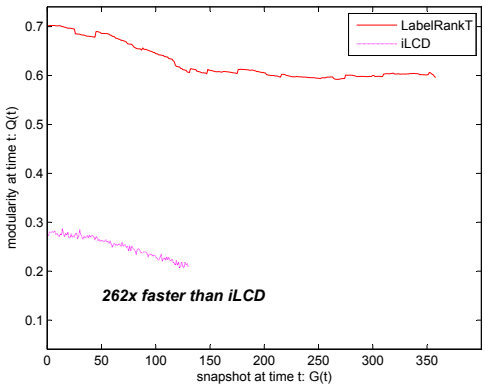

Figure 5.15: Comparison of modularity over time, $Q(t)$, with dynamic detection algorithms on Arxiv HEP-TH. We ran iLCD on only the first 130 snapshots due to the time complexity.

dynamic (especially incremental) algorithm like LabelRankT does not recompute the entire network, static algorithms might perform better. In fact, on AS Graph, three algorithms actually have close performance. Infomap slightly outperforms LabelRankT by about 5.03% in modularity on average, while LabelRankT and MCL are as close as 0.43% in difference. On Arxiv HEP-TH (with much larger size than AS Graph), Infomap and LabelRankT are very close with 0.88% in difference.

However, LabelRankT outperforms MCL significantly by 15.37%[18]. On the other hand, LabelRankT has benifit of effeciency. It runs 4 and 12 times faster than Infomap on AS Graph and Arxiv HEP-TH respectively. It is faster than MCL by 27 and 52 times on AS Graph and Arxiv HEP-TH respectively.

We also compared LabelRankT with two publicly available dynamic algorithms that employ incremental detection, namely facetNet[19] [127] and iLCD [20] [57]. On AS Graph, facetNet achieves performance similar to LabelRankT (the difference is just 0.07%), while iLCD does not find strong community structure at all. On Arxiv HEP-TH, facetNet does not work due to the overflow in memory, while LabelRankT performs at least twice better than iLCD. Moreover, LabelRankT is more than 100 times faster than either facetNet or iLCD on the two tested datasets.

The number of communities and community size (relative to n) distributions, i.e., the probability of seeing a community with certain size, produced by LabelRankT are tracked. As shown in Fig. 5.16, AS Graph does not evolve smoothly all the time. The abrupt drop in the number of communities at time 410 signals a dramatic change in structure, which is evidenced by a completely different distribution compared to the one detected at the beginning (time 1). Note that even though this violates our smoothness assumption, LabelRankT works well in this case because it is consistent with static algorithm Infomap. In contrast, Arxiv HEP-TH exhibits a fairly smooth pattern shown in Fig. 5.17, which follows a smooth performance in LabelRankT. The community distributions at time 1 and 350 (near the end of evolution) obey power laws with essentially identical exponents. However, small sized communities grow as more and more papers were published as indicated by the shift downward in distribution. Some communities grow relatively faster than the others and the largest communities expanded as indicated by the shift to the right (see the inset).

[18]Note that the behavior of MCL is partially due to its sensitivity to the parameter.

[19]Since facetNet requires the number of communities as input, we assigned it with the value produced by LabelRankT.

[20]After detection, if a node belongs to more than one community, we assigned it to the single community with maximum size to output disjoint partitioning.

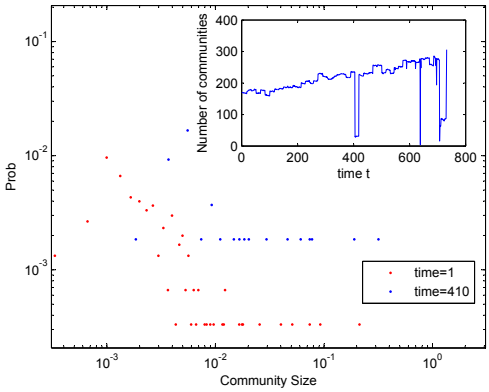

Figure 5.16: The community size distribution of AS-Internet Routers Graph tracked by LabelRankT (log-log plot). Results at time 1 and 410 (dramatic changes occur) are shown in the main plot. The inset shows the number of communities over time.

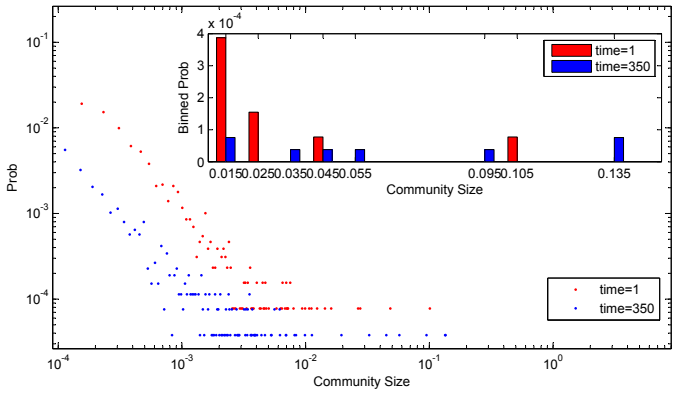

Figure 5.17: The community size distribution of Arxiv HEP-TH tracked by LabelRankT (log-log plot). Results at time 1 and 350 (near the end of evolution) are shown. In the inset, the tails of the distributions in main plot are binned with a width 0.01 to show the shift in large size communities.

5.5 Related Work

Label propagation based and random walk based algorithms are most relevant to our work (for review of other algorithms, refer to [33, 40]). LPA [81] starts from a configuration where each node has a distinct label. Each node holds only a single label and iteratively updates it to its neighborhood majority label. Disjoint communities are discovered by grouping nodes with the same label in a community when

the algorithm converges. COPRA [72] and SLPA[21] [86] extend LPA to detection of overlapping communities by allowing multiple labels. However, none of these algorithm resolves the randomness issue of LPA, where different solutions are produced in different runs.

Markov Cluster Algorithm (MCL) proposed in [133] is based on simulations of flow (random walk). MCL executes the alternation of matrix multiplication and inflation operator. The matrix multiplication corresponds to taking the k^{th} (usually 2nd) power of a stochastic matrix M. The latter corresponds to a parametrized operator $\Gamma_r : \Re^{k\times l} \rightarrow \Re^{k\times l}$ defined as $(\Gamma_r M)_{pq} = (M_{pq})^r / \sum_{i=1}^{k} (M_{iq})^r$. LabelRank is different from MCL in at least two aspects. One is that LabelRank applies the inflation to the label distributions rather than to the matrix M. The other is that the update of label distributions on each node in LabelRank requires only local information. Thus, it can be computed in a decentralized way.

For dynamic networks, an online or incremental detection algorithm [134] considers the stream of changes between snapshots explicitly as opposed to applying static algorithms to each static snapshot [129, 130, 135–139]. Ning et al. [140] proposed an incremental spectral clustering that continuously updates the eigenvalues and eigenvectors by carrying an approximation of the generalized eigenvalue system of the normalized cut. facetNet [127] extends the nonnegative matrix factorization algorithm [141]. The partitioning matrix in the previous snapshot can be used as the seed for the next one to handle the smooth change. The weakness of this algorithm is that the number of communities has to be provided as an input. GraphScope [142] is a parameter-free algorithm where the minimum description length principle is used to extract communities and to detect the changes. It does not consider the deletion of nodes. Bansal et al. [143] extended CNM algorithm [144], where two communities are merged at each step to optimize the increase in modularity. By reducing redundant computations that do not involve the changed edges, the algorithm shows better efficiency than CNM. However, their evaluations are limited to very small changes between snapshots. Similarly, Gorke et al. [145] modified both global and local modularity optimization based algorithms CNM and Louvain [146]

[21] https://sites.google.com/site/communitydetectionslpa.

to handle small changes. iLCD [57] updates the existing communities by adding a new node if its number of second neighbors and number of robust second neighbors are greater than expected values. A limitation is that it can not add two new nodes and a link in-between at the same time.

5.6 Summary

Detecting and tracking temporally evolving communities in a dynamic network where changes arrive as a stream is challenging. The proposed algorithm LabelRankT is based on a stabilized label propagation algorithm proposed in this chapter. At each step, it maintains the previous partitioning and dynamically updates only nodes involved in changes. Its effectiveness and efficiency are tested on real-world networks. As compared to other static algorithms including MCL and Infomap, LabelRankT achieves similar performance but with much lower computational cost. Furthermore, it significantly outperforms and is more than 100 times faster than dynamic detection algorithms such as facetNet and iLCD. The propagation scheme in LabelRankT is highly parallel and allows the computation to be distributed to each individual node. Such property will be particularly useful for applications like wireless sensor networks and mobile ad hoc networks, where each node in the network corresponds to a physical platform.

CHAPTER 6
DISCUSSION

6.1 Discussion on Opinion Dynamics

In Chapter 2, we have demonstrated the existence of a *tipping point* at which the initial majority opinion of a network switches quickly to that of a consistent and inflexible minority. There are several historical precedents for such events, for example, the suffragette movement in the early 20th century, and the rise of the American civil-rights movement that started shortly after the size of the African-American population crossed the 10% mark. Such processes have received some attention in sociological literature under the term *minority influence* [30, 147]. Our motivation here has been to study this process in more detail through semi-analytical methods and simulations for finite-sized and sparse networks, within the realm of a particular social influence model - the binary agreement model. There are several open questions and extensions of this work that are worth studying, in our opinion: for example, given a network with non-trivial community structure, what is the optimal scheme for selecting committed agents (for a given committed fraction) that would minimize consensus times, and reduce p_c? Secondly, extensions of the model to include utility-driven opinion switching by agents may be useful in designing optimal incentive schemes for opinion spreading.

In Chapter 3, by using a simple model, we have explored and quantified possible outcomes for the evolution of opinions on a social network in the presence of groups committed to competing opinions. Broadly speaking, our results indicate that as long as the fraction, p_B, of nodes committed to a given opinion B is held fixed at a value less than a critical value p_c, it is possible to induce the network to quickly tip over to a state where it widely adopts a competing opinion A, by introducing a fraction of nodes committed to opinion A. The value of the competing committed fraction, p_A, at which this tipping point arises depends on the value of p_B, and is determined by the bifurcation curve (see Fig. 2.1). Importantly, for a given value of $p_B < p_c$, the excess commitment $p_A - p_B$ required for the network

to tip over to A is a decreasing function of p_B that reaches zero when $p_B = p_c$. While the critical value p_c itself may depend on the network structure and its size, the feature described above holds for the three different classes of networks studied here. A corollary to this feature is that if the committed fraction p_B is held fixed at a value greater than p_c, increasing the competing committed fraction p_A only yields continuous incremental gains in the adoption of A (i.e., no tipping point or discontinuous changes in opinions exist).

Our results could be of utility in situations where public opinion is deadlocked due to the influence of competing committed groups. Perhaps one example of such a situation is the observed lack of consensus in the U.S. on the existence of human-induced climate change. Indeed, there is evidence in this particular case that the commitment of individuals to particular political ideologies may have an effect on their opinions [148]. Another scenario to which our model could bear some relevance is the adoption of competing industrial standards. A classic example of this scenario is the case of the Sellers' screw manufacturing standard that proliferated despite competition from the Whitworth standard [149]. A more recent example of such a scenario is the competition between Flash and HTML5 in web-development. A potential competition between DC fast charging standards is also expected as electric vehicles become increasingly popular with consumers.

To conclude, this thesis has presented results from a simple, abstract model for understanding how opinions on a social network evolve through social influence when there are groups within the network dedicated to competing opinions. Despite the simplicity of our model, we believe the insights provided here form a useful theoretical complement to data-driven studies [150] and randomized evaluations [151] aimed at understanding the spread of opinions.

However, our model still may be improved. There are many factors that can be considered to allow a more realistic model. One possibility is to consider that the committed opinions are spread subject to some constraints, for example, budget constraints. Given a constraint, the consensus state may or may not be reached, and the strategy for selecting the listener (as opposed to our random selection) could play a very important role. In the current model, we assume that the committed

agents have 100 percent faith in an opinion. As a relaxation, one can consider aging in the commitment, which is most likely application dependent. Also, we assume a perfect communication channel, and the third extension would be to incorporate the delay or noise in the communication.

6.2 Discussion on Community Detection

In the area of community detection, the work on directed weighted networks receives less attention compared to undirected unweighted ones. In Chapter 4, this thesis contributes a fast algorithm SLPA for overlapping community detection in large-scale networks. The results are also mainly on undirected unweighted networks. However, for many real-world applications, the weight and direction associated with each edge bear very important information and should be taken into account. After the codes were posted online[22], we actually received several inquiries of detection in weighted networks using SLPA for different applications. SLPA can be (and already has a simple implementation) for directed weighted networks. However, it needs more thorough tests and may benefit from a careful adaptation to the weighted networks.

To parallelize SLPA, the network partitioning is non-trivial. In Chapter 4, we have presented a preliminary version based on random subnetwork partition, where we simply distribute nodes randomly to processes. We only guarantee the total number of edges in each partition is close to the average number of edges per partition, creating a balance partitioning in that respect. However, in this case, the communication load between processes is not necessary balanced because it is defined by the number of edges crossing partition boundaries. Ideally, a pre-processing that can *efficiently* explore the network structure and minimize the community boundaries (edges between them) will help to offer better parallel performance.

In Chapter 5, aiming at a highly parallelizable online detection algorithm, this thesis first proposed operators to stabilize and boost the LPA, and then demonstrated how extended LPA can be generalized for dynamic networks. We believe the stabilization is important and can provide insights to an entire family of label

[22]https://sites.google.com/site/communitydetectionslpa.

propagation algorithms, including SLPA and COPRA. It is also possible to extend LabelRankT to overlapping community detection in a dynamic network as well. Combining ideas from both SLPA and LabelRankT helps achieve such goal.

As noted in [152], a distributed detection algorithm is valuable for highly distributed and self-organizable applications, such as ad hoc mobile networks (MANET), where each node in the network corresponds to a physical platform. Static group information has been shown to be useful in improving the routing in MANET [153]. However, work taking into the temporal (and even spatio) correlation is not yet fully studied. LabelRankT will be particularly suitable for such applications and is worthy of a further study. We plan to further improve its efficiency and apply it to real-life applications.

REFERENCES

[1] G. Tarde, *On Communication and Social Influence: Selected Papers.* Chicago, IL: University of Chicago Press, 1969.

[2] T. C. Schelling, *Micromotives and Macrobehavior.* New York, NY: Norton, 1978.

[3] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," *Rev. Mod. Phys.*, vol. 81, pp. 591–646, May 2009.

[4] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. SIGKDD Conf.*, 2003, pp. 137–146.

[5] S. Galam, "Application of statistical physics to politics," *Physica A*, vol. 274, no. 1-2, p. 132, Dec. 1999.

[6] ——, "Sociophysics: A review of galam models," *Int. J. Mod. Phys. C*, vol. 19, pp. 409–440, Jan. 2008.

[7] F. Harary, "A criterion for unanimity in french's theory of social power," in *Studies in Social Power.* Ann Arbor, MI: University of Michigan Press, 1959, pp. 168–182.

[8] N. E. Friedkin and E. C. Johnson, "Social influence and opinions," *J. Math. Sociol.*, vol. 15, no. 1, p. 193, Jan. 1990.

[9] E. M. Rogers, *Diffusion of Innovations.* Glencoe, IL: Free Press, 1962.

[10] M. Granovetter, "Threshold models of diffusion and collective behavior," *Am. J. Sociol.*, vol. 83, no. 6, p. 1420, May 1978.

[11] F. M. Bass, "A new product growth for model consumer durables," *Manage. Sci.*, vol. 15, no. 5, p. 215, Dec. 1969.

[12] B. Uzzi, S. Soderstrom, and D. Diermeier, "Buzz and the contagion of cultural products: The case of hollywood movies," 2011, unpublished.

[13] N. A. Christakis and J. H. Fowler, "The spread of obesity in a large social network over 32 years," *New Engl. J. Med.*, vol. 357, no. 4, p. 370, Jul. 2007.

[14] X. Castelló, A. Baronchelli, and V. Loreto, "Consensus and ordering in language dynamics," *Eur. Phys. J. B*, vol. 71, no. 4, p. 557, Aug. 2009.

[15] L. Steels, "A self-organizing spatial vocabulary," *Artif. Life*, vol. 2, pp. 319–332, Jan. 1995.

[16] A. Baronchelli, M. Felici, E. Caglioti, V. Loreto, and L. Steels, "Sharp transition towards shared vocabularies in multi-agent systems," *J. Stat. Mech.*, vol. 2006, no. 06, p. P06014, Jun. 2006.

[17] L. Dall'Asta, A. Baronchelli, A. Barrat, and V. Loreto, "Nonequilibrium dynamics of language games on complex networks," *Phys. Rev. E*, vol. 74, no. 3, p. 036105, Sep. 2006.

[18] Q. Lu, G. Korniss, and B. K. Szymanski, "Naming games in two-dimensional and small-world-connected random geometric networks," *Phys. Rev. E*, vol. 77, no. 1, p. 016111, Jan. 2008.

[19] A. Baronchelli, "Role of feedback and broadcasting in the naming game," *Phys. Rev. E*, vol. 83, no. 4, p. 046103, Apr. 2011.

[20] D. Stauffer, "Sociophysics: the sznajd model and its applications," *Comput. Phys. Comm.*, vol. 146, no. 1, pp. 93–98, Jun. 2002.

[21] P. L. Krapivsky and S. Redner, "Dynamics of majority rule in two-state interacting spin systems," *Phys. Rev. Lett.*, vol. 90, no. 23, p. 238701, Jun. 2003.

[22] V. Sood and S. Redner, "Voter model on heterogeneous graphs," *Phys. Rev. Lett.*, vol. 94, p. 178701, May 2005.

[23] P. Krapivsky, "Kinetics of monomer-monomer surface catalytic reactions," *Phys. Rev. A*, vol. 45, pp. 1067–1072, Jan. 1992.

[24] D. J. Daley and J. Gani, *Epidemic Modeling: An Introduction.* Cambridge, UK: Cambridge University Press, 2005.

[25] D. J. Watts and P. S. Dodds, "Influentials, networks, and public opinion formation," *J. Cons. Res.*, vol. 34, no. 1, pp. 441–458, Dec. 2007.

[26] Q. Lu, G. Korniss, and B. K. Szymanski, "The naming game in social networks: community formation and consensus engineering," *J. Econ. Interact. Coord.*, vol. 4, p. 221, Jul. 2009.

[27] S. Biswas and P. Sen, "Model of binary opinion dynamics: Coarsening and effect of disorder," *Phys. Rev. E*, vol. 80, no. 2, p. 027101, Aug. 2009.

[28] M. Mobilia, A. Petersen, and S. Redner, "On the role of zealotry in the voter model," *J. Stat. Mech.*, vol. 2007, no. 8, p. P08029, Aug. 2007.

[29] E. Yildiz, D. Acemoglu, A. E. Ozdaglar, A. Saberi, and A. Scaglione, "Discrete opinion dynamics with stubborn agents," Jan. 2011, unpublished.

[30] S. Galam and F. Jacobs, "The role of inflexible minorities in the breaking of democratic opinion dynamics," *Physica A*, vol. 381, pp. 366–376, Jul. 2007.

[31] S. Galam, "Public debates driven by incomplete scientific data: The cases of evolution theory, global warming and h1n1 pandemic influenza," *Physica A*, vol. 389, no. 17, pp. 3619–3631, Apr. 2010.

[32] N. Morris, "Bridging the gap: An examination of diffusion and participatory approaches in development communication," The Manoff Group, Inc., Tech. Rep., 2000.

[33] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, pp. 75–174, Feb. 2010.

[34] L. Danon, J. Duch, A. Arenas, and A. Diaz-guilera, "Comparing community structure identification," *J. Stat. Mech.*, vol. 2005, no. 09, p. P09008, Sep. 2005.

[35] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Phys. Rev. E*, vol. 80, p. 056117, Nov. 2009.

[36] J. Leskovec, K. J. Lang, and M. Mahoney, "Empirical comparison of algorithms for network community detection," in *Proc. WWW Conf.*, 2010, pp. 631–640.

[37] S. Kelley, M. Goldberg, M. Magdon-Ismail, K. Mertsalov, and A. Wallace, *Handbook of Optimization in Complex Networks.* New York, NY: Springer, 2011, ch. 6.

[38] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, no. 1, pp. 814–818, Jun. 2005.

[39] J. Baumes, M. Goldberg, M. Krishnamoorthy, M. Magdon-Ismail, and N. Preston, "Finding communities by clustering a graph into overlapping subgraphs," in *Proc. IADIS Conf.*, 2005, pp. 97–104.

[40] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Comput. Surv. (to be published)*, 2012.

[41] I. Farkas, D. Ábel, G. Palla, and T. Vicsek, "Weighted network modules," *New J. Phys.*, vol. 9, no. 6, p. 180, Jun. 2007.

[42] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki, "Sequential algorithm for fast clique percolation," *Phys. Rev. E*, vol. 78, no. 2, p. 026109, Aug. 2008.

[43] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," *Nature*, vol. 466, no. 1, pp. 761–764, Aug. 2010.

[44] T. S. Evans and R. Lambiotte, "Line graphs, link partitions and overlapping communities," *Phys. Rev. E*, vol. 80, p. 016105, Jul. 2009.

[45] T. Evans and R. Lambiotte, "Line graphs of weighted networks for overlapping communities," *Eur. Phys. J. B*, vol. 77, pp. 265–272, Sep. 2010.

[46] Z. Wu, Y. Lin, H. Wan, and S. Tian, "A fast and reasonable method for community detection with adjustable extent of overlapping," in *Proc. ISKE Conf.*, 2010, pp. 376–379.

[47] Y. Kim and H. Jeong, "Map equation for link communities," *Phys. Rev. E*, vol. 84, p. 026110, Aug. 2011.

[48] T. Evans, "Clique graphs and overlapping communities," *J. Stat. Mech.*, vol. 2010, no. 12, p. P12037, Dec. 2010.

[49] J. Baumes, M. Goldberg, and M. Magdon-Ismail, "Efficient identification of overlapping communities."

[50] S. Kelley, "The existence and discovery of overlapping communities in large-scale networks," Ph.D. dissertation, Rensselaer Polytechnic Inst., Troy, NY, 2009.

[51] A. Lancichinetti, S. Fortunato, and J. Kertesz, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, p. 033015, Mar. 2009.

[52] F. Havemann, M. Heinz, A. Struck, and J. Glaser, "Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels," *J. Stat. Mech.*, vol. 2011, no. 01, p. P01023, Jan. 2011.

[53] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS ONE*, vol. 6, no. 4, p. e18961, Apr. 2011.

[54] D. Jin, B. Yang, C. Baquero, D. Liu, D. He, and J. Liu, "A markov random walk under constraint for discovering overlapping communities in complex networks," *J. Stat. Mech.*, vol. 2011, no. 05, p. P05031, May 2011.

[55] A. Padrol-Sureda, G. Perarnau-Llobet, J. Pfeifle, and V. Munts-Mulero, "Overlapping community search for social networks," in *Proc. ICDE*, 2010, pp. 992–995.

[56] D. Chen, M. Shang, Z. Lv, and Y. Fu, "Detecting overlapping communities of weighted networks via a local algorithm," *Physica A*, vol. 389, no. 19, pp. 4177–4187, Oct. 2010.

[57] R. Cazabet, F. Amblard, and C. Hanachi, "Detection of overlapping communities in dynamical social networks," in *Proc. SocialCom Conf.*, 2010, pp. 309–314.

[58] H. Shen, X. Cheng, K. Cai, and M.-B. Hu, "Detect overlapping and hierarchical community structure," *Physica A*, vol. 388, pp. 1706–1712, Apr. 2009.

[59] H. Shen, X. Cheng, and J. Guo, "Quantifying and identifying the overlapping community structure in networks," *J. Stat. Mech.*, vol. 2009, no. 7, p. 9, Jul. 2009.

[60] C. Lee, F. Reid, A. McDaid, and N. Hurley, "Detecting highly overlapping community structure by greedy clique expansion," in *Proc. SNAKDD Workshop*, 2010, pp. 33–42.

[61] N. Du, B. Wang, and B. Wu, "Overlapping community structure detection in networks," in *Proc. SIGKDD Conf.*, 2008, pp. 1371–1372.

[62] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó, "Fuzzy communities and the concept of bridgeness in complex networks," *Phys. Rev. E*, vol. 77, p. 016107, Jan. 2008.

[63] S. Zhang, R.-S. Wangb, and X.-S. Zhang, "Identification of overlapping community structure in complex networks using fuzzy c-means clustering," *Physica A*, vol. 374, pp. 483–490, Jan. 2007.

[64] W. Ren, G. Yan, X. Liao, and L. Xiao, "Simple probabilistic algorithm for detecting community structure," *Phys. Rev. E*, vol. 79, no. 3, p. 036111, Mar. 2009.

[65] Q. Fu and A. Banerjee, "Multiplicative mixture models for overlapping clustering," in *Proc. ICDM*, 2008, pp. 791–796.

[66] M. Magdon-ismail and J. Purnell, "Fast overlapping clustering of networks using sampled spectral distance embedding and gmms," Rensselaer Polytechnic Inst., Tech. Rep., 2011.

[67] P. Latouche, E. Birmele, and C. Ambroise, "Overlapping stochastic block models with application to the french political blogosphere," *Ann. Appl. Stat.*, vol. 5, no. 1, pp. 309–336, Jan. 2011.

[68] A. McDaid and N. Hurley, "Detecting highly overlapping communities with model-based overlapping seed expansion," in *Proc. ASONAM Conf.*, 2010, pp. 112–119.

[69] S. Zhang, R.-S. Wang, and X.-S. Zhang, "Uncovering fuzzy community structure in complex networks," *Phys. Rev. E*, vol. 76, no. 4, p. 046103, Oct. 2007.

[70] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon, "Overlapping community detection using bayesian non-negative matrix factorization," *Phys. Rev. E*, vol. 83, no. 6, p. 066114, Jun. 2011.

[71] F. Ding, Z. Luo, J. Shi, and X. Fang, "Overlapping community detection by kernel-based fuzzy affinity propagation," in *Proc. ISA Workshop*, 2010, pp. 1–4.

[72] S. Gregory, "Finding overlapping communities in networks by label propagation," *New J. Phys.*, vol. 12, no. 10, p. 103018, Oct. 2010.

[73] J. Xie, B. K. Szymanski, and X. Liu, "SLPA: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process," in *Proc. ICDM Workshop*, 2011, pp. 344–349.

[74] W. Chen, Z. Liu, X. Sun, and Y. Wang, "A game-theoretic framework to identify overlapping communities in social networks," *Data Min. Knowl. Disc.*, vol. 21, pp. 224–240, Sep. 2010.

[75] F. Breve, L. Zhao, and M. Quiles, "Uncovering overlap community structure in complex networks using particle competition," in *Proc. ICAI*, 2009, pp. 619–628.

[76] J. Reichardt and S. Bornholdt, "Detecting fuzzy community structures in complex networks with a potts model," *Phys. Rev. Lett.*, vol. 93, p. 218701, Nov. 2004.

[77] M. Blatt, S. Wiseman, and E. Domany, "Superparamagnetic clustering of data," *Phys. Rev. Lett.*, vol. 76, pp. 3251–3254, Apr. 1996.

[78] D. Li, I. Leyva, J. Almendral, I. Sendina-Nadal, J. Buldu, S. Havlin, and S. Boccaletti, "Synchronization interfaces and overlapping communities in complex networks," *Phys. Rev. Lett.*, vol. 101, p. 168701, Oct. 2008.

[79] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente, "Synchronization reveals topological scales in complex networks," *Phys. Rev. Lett.*, vol. 96, p. 114102, Mar. 2006.

[80] J. Reichardt. and S. Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E*, vol. 74, no. 1, p. 016110, Jul 2006.

[81] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E*, vol. 76, p. 036106, Sep. 2007.

[82] G. Tibély and J. Kertész, "On the equivalence of the label propagation method of community detection and a potts model approach," *Physica A*, vol. 387, pp. 4982–4984, Aug. 2008.

[83] I. X. Y. Leung, P. Hui, P. Li, and J. Crowcroft, "Towards real-time community detection in large networks," *Phys. Rev. E*, vol. 79, p. 066107, Jun. 2009.

[84] J. Xie, S. Sreenivasan, G. Korniss, W. Zhang, C. Lim, and B. K. Szymanski, "Social consensus through the influence of committed minorities," *Phys. Rev. E*, vol. 84, no. 1, p. 011130, Jul. 2011.

[85] J. Xie, J. Emenheiser, M. Kirby, S. Sreenivasan, B. K. Szymanski, and G. Korniss, "Evolution of opinions on social networks in the presence of competing committed groups," *PLoS ONE*, vol. 7, no. 3, p. e33215, Mar. 2012.

[86] J. Xie and B. K. Szymanski, "Towards linear time overlapping community detection in social networks," in *Proc. PAKDD Conf.*, 2012, pp. 25–36.

[87] ——, "LabelRankT: A decentralized online algorithm for detection of evolving communities," *CIKM (under review)*, 2012.

[88] R. Dickman and R. Vidigal, "Quasi-stationary distributions for stochastic processes with an absorbing state," *J. Phys. A*, vol. 35, no. 5, p. 1147, Feb. 2002.

[89] R. Dickman, "Numerical analysis of the master equation," *Phys. Rev. E*, vol. 65, no. 4, p. 047701, Mar. 2002.

[90] L. Dall'Asta and A. Baronchelli, "Microscopic activity patterns in the naming game," *J Phys. A*, vol. 39, no. 48, p. 14851, Dec. 2006.

[91] M. M. de Oliveira and R. Dickman, "Quasi-stationary distributions for models of heterogeneous catalysis," *Physica A*, vol. 343, no. 1, pp. 525–542, Nov. 2004.

[92] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, "Modularity from fluctuations in random graphs and complex networks," *Phys. Rev. E*, vol. 70, p. 025101, Aug. 2004.

[93] T. Opsahl and P. Panzarasa, "Clustering in weighted networks," *Soc. Networks*, vol. 31, no. 2, pp. 155–163, May 2009.

[94] M. E. J. Newman, "The structure of scientific collaboration networks," *Proc. Natl. Acad. Sci.*, vol. 98, no. 2, pp. 404–409, Jan. 2001.

[95] W. Zhang, C. Lim, and B. Szymanski, "Tipping points of diehards in social consensus on large random networks," in *Proc. ComplexNet Workshop*, 2012.

[96] D. Landau and K. Binder, *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge, UK: Cambridge University Press, 2000.

[97] T. Mori, S. Miyashita, and P. A. Rikvold, "Asymptotic forms and scaling properties of the relaxation time near threshold points in spinodal-type dynamical phase transitions," *Phys. Rev. E*, vol. 81, p. 011135, Jan. 2010.

[98] M. I. Dykman, E. Mori, J. Ross, and P. M. Hunt, "Large fluctuations and optimal paths in chemical kinetics," *J. Chem. Phys.*, vol. 100, no. 8, pp. 5735–5750, Apr. 1994.

[99] T. S. Gardner, C. R. Cantor, and J. J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, no. 6767, pp. 339–342, Jan. 2000.

[100] V. I. Arnold, *Geometrical Methods in the Theory of Ordinary Differential Equations*. New York, NY: Springer, 1988.

[101] D. W. Herrmann, W. Klein, and D. Stauffer, "Spinodals in a long-range interaction system," *Phys. Rev. Lett.*, vol. 49, pp. 1262–1264, Oct. 1982.

[102] T. S. Ray, "Evidence for spinodal singularities in high-dimensional nearest-neighbor ising models," *J. Stat. Phys.*, vol. 62, pp. 463–472, Oct. 1991.

[103] R. S. Maier, "Large fluctuations in stochastically perturbed nonlinear systems," in *Proc. Applicat. Comput. Conf.*, 1992, pp. 1–21.

[104] R. Graham and T. Tél, "On the weak-noise limit of fokker-planck models," *J. Stat. Phys.*, vol. 35, no. 5, pp. 729–748, Feb. 1984.

[105] H. Gang, "Stationary solution of master equations in the large-system-size limit," *Phys. Rev. A*, vol. 36, pp. 5782–5790, Dec. 1987.

[106] P. V. E. M. D. G. Luchinsky and M. I. Dykman, "Analogue studies of nonlinear systems," *Rep. Prog. Phys.*, vol. 61, no. 8, p. 889, Aug. 1998.

[107] B. Bollobas, *Modern Graph Theory*. New York, NY: Springer, 2001.

[108] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[109] A. Baronchelli, L. Dall'Asta, A. Barrat, and V. Loreto, "Nonequilibrium phase transition in negotiation dynamics," *Phys. Rev. E*, vol. 76, p. 051102, Nov. 2007.

[110] J. Xie and B. K. Szymanski, "Community detection using a neighborhood strength driven label propagation algorithm," in *Proc. NSW*, 2011, pp. 188–195.

[111] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E*, vol. 78, p. 046110, Oct. 2008.

[112] L. M. Collins and C. W. Dent, "Omega: A general formulation of the rand index of cluster recovery suitable for non-disjoint solutions," *Multivar. Behav. Res.*, vol. 23, no. 2, pp. 231–242, Feb. 1988.

[113] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, "Extending the definition of modularity to directed graphs with overlapping communities," *J. Stat. Mech.*, vol. 2009, no. 3, p. P03024, Mar. 2009.

[114] C. B. M. Rosvall, "Maps of random walks on complex networks reveal community structure," *Proc. Natl. Acad. Sci.*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008.

[115] M. LaMonica. (2008, Aug.) Amazon web services adds resiliency to EC2 compute service. [Online]. Available: http://en.wikipedia.org/wiki/Amazon_EC2, [Accessed Apr. 5, 2012].

[116] M. Srivastava, T. Abdelzaher, and B. Szymanski, "Human-centric sensing," *Phil. Trans. Roy. Soc.*, vol. 370, no. 1958, pp. 176–197, Jan. 2012.

[117] S. van Dongen, "A cluster algorithm for graphs," Nat. Res. Inst. Math. Comput. Sci., Tech. Rep. INS-R0010, 2000.

[118] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fund. Math.*, vol. 3, no. 1, pp. 133–181, Jan. 1922.

[119] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, p. 066133, Jun. 2004.

[120] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Phys. Rev. E*, vol. 68, p. 065103, Dec. 2003.

[121] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proc. WWW Conf.*, 2010, pp. 641–650.

[122] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks: Technology, Protocols, and Applications*. Hoboken, NJ: Wiley, 2007.

[123] I. Chlamtac, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, pp. 13–64, Jul. 2003.

[124] W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthr. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.

[125] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proc. Natl. Acad. Sci.*, vol. 104, no. 1, pp. 36–41, Jan. 2007.

[126] B. H. Good, Y.-A. de Montjoye, and A. Clauset, "The performance of modularity maximization in practical contexts," *Phys. Rev. E*, vol. 81, p. 046106, Apr. 2010.

[127] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Analyzing communities and their evolutions in dynamic social networks," *Trans. Knowl. Disc. Data*, vol. 3, no. 2, pp. 8:1–8:31, Apr. 2009.

[128] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, "Community structure in time-dependent, multiscale, and multiplex networks," *Science*, vol. 328, no. 5980, pp. 876–878, May 2010.

[129] V. Kawadia and S. Sreenivasan. (2012) Online detection of temporal communities in evolving networks by estrangement confinement. [Online]. Available: http://arxiv.org/abs/1203.5126, [Accessed Apr. 5, 2012].

[130] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs," *Trans. Knowl. Disc. Data*, vol. 3, no. 4, pp. 16:1–16:36, Mar. 2009.

[131] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proc. SIGKDD Conf.*, 2005, pp. 177–187.

[132] J. Gehrke, P. Ginsparg, and J. M. Kleinberg, "Overview of the 2003 KDD cup," *SIGKDD Explor.*, vol. 5, no. 2, pp. 149–151, Dec. 2003.

[133] S. van Dongen, "Graph clustering by flow simulation," Ph.D. dissertation, Univ. Utrecht, Netherlands, 2000.

[134] T. Aynaud, J.-L. Guillaume, Q. Wang, and E. Fleury. (2011) Communities in evolving networks: definitions, detections and analysis techniques. [Online]. Available: http://www-rp.lip6.fr/~magnien/DynGraph/Docs/art_communities.pdf, [Accessed Apr. 5, 2012].

[135] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," *Proc. Natl. Acad. Sci.*, vol. 101, pp. 5249–5253, Apr. 2004.

[136] T. Y. Berger-Wolf and J. Saia, "A framework for analysis of dynamic social networks," in *Proc. SIGKDD Conf.*, 2006, pp. 523–528.

[137] G. Palla, A.-L. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, Apr. 2007.

[138] M. Goldberg, M. Magdon-Ismail, S. Nambirajan, and J. Thompson, "Tracking and predicting evolution of social communities," in *Proc. SocialCom Conf.*, 2011, pp. 780–783.

[139] S. Pandit, Y. Yang, V. Kawadia, S. Sreenivasan, and N. V. Chawla, "Detecting communities in time-evolving proximity networks," in *Proc. NSW*, 2011, pp. 173–179.

[140] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. Huang, "Incremental spectral clustering with application to monitoring of evolving blog communities," in *Proc. SIAM Conf.*, 2007.

[141] S. Yu, K. Yu, and V. Tresp, "Soft clustering on graphs," in *Proc. NIPS Conf.*, 2005.

[142] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "Graphscope: parameter-free mining of large time-evolving graphs," in *Proc. SIGKDD Conf.*, 2007, pp. 687–696.

[143] S. Bansal, S. Bhowmick, and P. Paymal, "Fast community detection for dynamic complex networks," in *Proc. CompleNet Workshop*, 2010.

[144] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, p. 066111, Dec. 2004.

[145] R. Görke, P. Maillard, C. Staudt, and D. Wagner, "Modularity-Driven Clustering of Dynamic Graphs," in *Proc. SEA Symp.*, 2010, pp. 436–448.

[146] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, vol. 2008, no. 10, p. P10008, Oct. 2008.

[147] S. Moscovici, E. Lage, and M. Naffrechoux, "Influence of a consistent minority on the responses of a majority in a color perception task," *Sociometry*, vol. 32, no. 4, pp. 365–380, Dec. 1969.

[148] L. A., M. E., R.-R. C., and H. J. D., "Politics and global warming: Democrats, republicans, independents, and the tea party," Sep. 2011, unpublished.

[149] B. Sinclair, "At the turn of a screw: William sellers, the franklin institute, and a standard american thread," *Technol. Cult.*, vol. 10, no. 1, p. 1, Jan. 1969.

[150] A. Madan, K. Farrahi, D. Gatica-Perez, and A. Pentland, "Pervasive sensing to model political opinions in face-to-face networks," in *Proc. Pervasive Conf.*, 2011, pp. 214–231.

[151] E. D. A. Banerjee, A. Chandrasekhar and M. Jackson, "The diffusion of microfinance," MIT working paper.

[152] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proc. MobiArch Workshop*, 2007, pp. 7:1–7:8.

[153] T. N. Dinh, Y. Xuan, and M. T. Thai, "Towards social-aware routing in dynamic communication networks," in *Proc. IPCCC*, 2009, pp. 161–168.

[154] J. Carr, *Applications of Centre Manifold Theory.* New York, NY: Springer, 1981.

[155] N. V. Kampen, *Stochastic processes in physics and chemistry.* Oxford, UK: Elsevier, 2007.

[156] M. I. Freidlin and A. D. Wentzell, *Random Perturbations of Dynamical Systems.* New York, NY: Springer, 1984.

[157] M. I. Dykman and M. A. Krivoglaz, "Theory of fluctuational transitions between stable states of a nonlinear oscillator," *Sov. J. Exp. Theor. Phys.*, vol. 50, no. 1, p. 30, Jul. 1979.

APPENDIX A
APPENDIX OF CHAPTER 2

A.1 Fixed Points of the Mean-field Equations

Here, we analyze the mean-field equations for the existence of fixed points. To simplify notation we use the notation $x = n_A$ and $y = n_B$. Thus for a fixed point of the evolution given by Eq. (3.1):

$$\begin{aligned} -xy &+ (1-x-y-p)^2 + x(1-x-y-p) + 1.5p(1-x-y-p) = 0 \\ -xy &+ (1-x-y-p)^2 + y(1-x-y-p) - yp = 0, \end{aligned} \tag{A.1}$$

which can be reduced further to:

$$\begin{aligned} x &= [(1-y-p/4)^2 - 9p^2/16]/(p/2+1) \\ y &= (1-x-p)^2. \end{aligned} \tag{A.2}$$

Substituting the expression of x into the expression for y, and denoting $z^2 = y$ we get

$$z(z^3 - (2-p/2)z + p/2 + 1) = 0. \tag{A.3}$$

For any value of p, $z = z_0 = 0$ is a solution to the above equation. In other words, for any value of p, the mean-field equations admit a stable fixed point, $n_A = x_1 = 1-p$, $n_B = y_1 = 0$ which represents the network having reached a consensus state where all nodes have adopted the opinion of the committed agents.

The remaining fixed points are roots of

$$f(z) = z^3 - (2-p/2)z + p/2 + 1 = 0. \tag{A.4}$$

In order to find the criterion which has to be satisfied for valid roots (i.e. $0 \leq z \leq \sqrt{(1-p)}$) of the above equation to exist, we analyze the extrema of the function $f(z)$ which are given by:

$$f'(z) = 3z^2 - 2 + p/2 = 0. \tag{A.5}$$

Thus, the extrema occur at :

$$z_{1,2} = \pm\sqrt{2/3 - p/6}. \tag{A.6}$$

It can be seen from Eq. (A.5) that $f(z)$ is increasing, decreasing and increasing again in the intervals $(-\infty, z_1)$, (z_1, z_2), $(z_2, +\infty)$ respectively. Consequently, $f(z)$ achieves a maximum at $-1 < z_1 = -\sqrt{2/3 - p/6} < 0$ and a minimum at $0 < z_2 = \sqrt{2/3 - p/6} < 1$. Furthermore, since $f(-2) = -p/2 - 3 < 0$ and $f(-1) = 2 > 0$, one root of $f(z) = 0$ occurs in the interval $-2 < z < -1$. Since $f(z)$ is positive at z_1, decreasing from z_1 to z_2 where it achieves a minimum, and increasing thereafter,

it follows that a necessary and sufficient condition for more roots of $f(z) = 0$ to exist, is that $f(z_2)$ be less than zero:

$$f(z_2) = z_2^3 - (2 - p/2)z_2 + p/2 + 1 < 0.$$

Denoting $z_2 = q$ and $p = 4 - 6q^2$ (from Eq. (A.6)) yields the following inequality for q as a condition for more roots of $f(z) = 0$ to exist:

$$f(q) = q^3 + 1.5q^2 - 1.5 > 0. \tag{A.7}$$

Note that z_2 is itself a function of p. Analyzing the derivative of $f(q)$ enables us to glean that the inequality Eq. (A.7) is satisfied for $q > q_0$ where q_0 is the solution of the cubic equation $f(q) = 0$ and is given by:

$$q_0 = \left[\sqrt[3]{5 + \sqrt{24}} + \sqrt[3]{5 - \sqrt{24}} - 1\right]/2.$$

Thus, the original fixed point equation Eq. (A.1) has at least one valid root besides $z = 0$, so long as p is less than or equal to:

$$p_c = \frac{5}{2} - \frac{3}{2}\left(\sqrt[3]{5 + \sqrt{24}} - 1\right)^2 - \frac{3}{2}\left(\sqrt[3]{5 - \sqrt{24}} - 1\right)^2, \tag{A.8}$$

which using standard computer algebra software is evaluated to be $p_c = 0.09789$. Using, $z^2 = y = q_0$ and Eq. (A.2), we obtain the state of the system at p_c to be $\{n_A, n_B\} = \{0.0957, 0.6504\}$. It also follows from the expression for $f(z)$, that $f(0) > 0$ and therefore if $f(z_2)$ is negative, Eq. (A.4) has *two* roots on the positive line when $p < p_c$. Thus there are two fixed points of Eq. (A.1) when $p < p_c$.

The exact expressions for these fixed points (that can also be obtained numerically), obscure their dependence on p. We therefore adopt an approximation which exhibits a much clearer dependence of the fixed point values on p, and numerically yield values close to those obtained from the exact expressions. Substituting $z = t\sqrt{2 - \frac{p}{2}}$ in Eq. (A.4) reduces it to:

$$t^3 - t + r = 0, \tag{A.9}$$

where $r = \frac{1+p/2}{(\sqrt{2-p/2})^3}$. Clearly, r is a monotonically increasing function of p, and therefore $\frac{1}{2\sqrt{2}} \leq r < \frac{1+p_c/2}{(\sqrt{2-p_c/2})^3} = \frac{2}{3\sqrt{3}}$ for $0 \leq p < p_c$, our range of interest. Function $g(t) = t^3 - t$ is monotonically decreasing for $t < -1$ and $g(-1) = 0$, while $g(-2) < -1 < -r$. Hence, there is a real root $t_1 \in (-2, -1)$ to Eq. (A.9) which is a monotonically decreasing function of r, but which clearly does not yield a valid fixed point. This root can be expressed as $t_1(r) = -\frac{2}{\sqrt{3}} + \epsilon(r)$, where $\epsilon(r)$ is monotonically decreasing from less than 0.0106 to 0 over the range of our interest for r. Substituting this expression back into Eq. A.9 and neglecting powers of $\epsilon(r)$

higher than unity, we get an approximation of ϵ in terms of r, and consequently an approximation for t_1:

$$t_1(r) \approx -\frac{16}{9\sqrt{3}} - \frac{r}{3} \tag{A.10}$$

with relative error of less than 0.01%. Now, we can factorize the LHS of Eq. (A.9) and write it as $(t^2+bt+c)(t-t_1)$. Equating this factorized expression with t^3-t+r, gives us b and c in terms of r. Thus, two more roots of Eq. (A.9) are obtained in terms of r by solving the quadratic equation $t^2 + bt + c = 0$ which yields:

$$t_{2,3} = \frac{8}{9\sqrt{3}} + \frac{r}{6} \pm \sqrt{\frac{17}{81} - \frac{8r}{9\sqrt{3}} - \frac{r^2}{12}}. \tag{A.11}$$

Finally, we can obtain the values of z associated with the above roots, and therefore the values of x and y written in terms of these roots are derived as:

$$\begin{aligned} y_{2,3} &= t_{2,3}^2 \frac{4-p}{2} \\ x_{2,3} &= \frac{(4-4y_{2,3}-p)^2 - 9p^2}{8p+16}. \end{aligned} \tag{A.12}$$

The stability of these fixed points can be checked via linear stability analysis. Following the standard procedure, the stability matrix is given by:

$$S = \begin{bmatrix} -1-\frac{p}{2} & -2+2y^*+\frac{p}{2} \\ -2+2x^*+2p & -1 \end{bmatrix},$$

where (x^*, y^*) is the fixed point under consideration. The eigenvalues of the stability matrix are given by:

$$\lambda = \frac{1}{4}\Big(-4-p \pm (17p^2 + 64(x^*-1)(y^*-1) + 16p(x^*+4y^*-5))^{\frac{1}{2}} \Big) \tag{A.13}$$

From the expression for the eigenvalues we numerically determine that the real part of both the eigenvalues is negative for (x_2, y_2) over the range $0 \leq p < p_c$ indicating that (x_2, y_2) is a stable fixed point. This is however, not the case for (x_3, y_3), making it unstable. Similarly, the consensus fixed point (x_1, y_1) is found to be stable for $0 \leq p \leq 1$. Finally, we note that as $p \to 0$, the stable fixed point converges to $n_A = 0, n_B = 1$, while the unstable fixed point converges to $n_A = n_B \approx 0.38$.

APPENDIX B
APPENDIX OF CHAPTER 3

B.1 Analysis of Steady States and a Critical Value for $p_A = p_B$

For notational simplicity we replace n_A by x and n_B by y. The mean field equations describing the system with $p_A = p_B = p$, $0 < p \leq 0.5$ then are:

$$\begin{aligned}\frac{dx}{dt} &= -xy + (1-x-y-2p)^2 + x(1-x-y-2p) + \frac{3}{2}p(1-x-y-2p) - px \\ \frac{dx}{dt} &= -xy + (1-x-y-2p)^2 + y(1-x-y-2p) + \frac{3}{2}p(1-x-y-2p) - py,\end{aligned} \tag{B.1}$$

where $n_{AB} = 1 - x - y - 2p$. In the steady state, $dx/dt = dy/dt = 0$, and the resulting equations can be solved to yield four solutions for (x, y). Out of these one solution lies outside the valid range for all feasible values of p, i.e., $0 < p \leq 0.5$. The valid fixed points for Eqs. B.1 are:

$$\begin{aligned}x_1 &= \frac{3}{2} - \frac{1}{2}\sqrt{5-2p} - p \\ y_1 &= \frac{3}{2} - \frac{1}{2}\sqrt{5-2p} - p\end{aligned}$$

$$\begin{aligned}x_2 &= \frac{1}{2} + \frac{1}{2}\sqrt{1-p^2-6p} - \frac{3}{2}p \\ y_2 &= \frac{1}{2} - \frac{1}{2}\sqrt{1-p^2-6p} - \frac{3}{2}p\end{aligned}$$

$$\begin{aligned}x_3 &= \frac{1}{2} - \frac{1}{2}\sqrt{1-p^2-6p} - \frac{3}{2}p \\ y_3 &= \frac{1}{2} + \frac{1}{2}\sqrt{1-p^2-6p} - \frac{3}{2}p.\end{aligned}$$

Since the solutions are symmetric in x and y, in order to investigate the range of p over which these solutions are valid, we restrict our analysis to y. The solution y_1 is valid for all values of p. For y_2, y_3 to be valid solutions, we require $U(p) = 1 - p^2 - 6p \geq 0$. $U(p)$ is a monotonically decreasing function for $p > 0$, and the value of p at which $U(p)$ first crosses zero is the critical point.

$$p_c = \sqrt{10} - 3 \approxeq 0.1623. \tag{B.2}$$

Thus, there exist three fixed points in the range $[0, p_c]$. In the range $(p_c, 0.5]$ only one valid fixed point exists, viz. (x_1, y_1).

We can further examine the stability of the obtained fixed points. Linear stability analysis yields the following stability matrix:

$$Q = \begin{bmatrix} -1 - \frac{p}{2} & -2 + 2y^* + \frac{5}{2}p \\ -2 + 2x^* + \frac{5}{2}p & -1 - \frac{p}{2} \end{bmatrix}, \tag{B.3}$$

where (x^*, y^*) is the fixed point under consideration. The eigenvalues of the stability matrix at the fixed point are given by

$$\lambda = -(2 + p) \pm \sqrt{26p^2 + (20(x^* + y^*) - 36) + 16(1 - x^* - y^* + x^*y^*)},$$

and examination of the real part of these eigenvalues indicates that (x_2, y_2) and (x_3, y_3) are stable fixed points, and (x_1, y_1) is an unstable fixed point (saddle point) for $p \leq p_c = 0.1623$. For $p > p_c$, (x_1, y_1), the only valid fixed point, is a stable fixed point. Figure B.1 shows the movement of the fixed points in the phase space as a function of p.

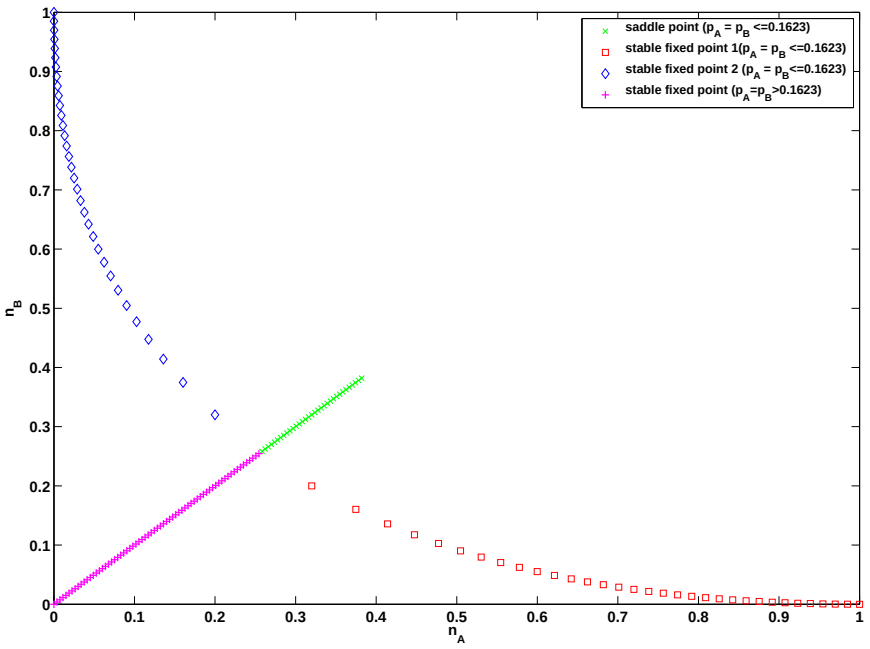

Figure B.1: Movement of fixed points as p_A and p_B are smoothly varied along the diagonal line $p_A = p_B$. For $p_A = p_B, p_c \approxeq 0.1623$ three fixed points exist, two of which are stable, and the third is unstable. For $p_A = p_B > p_c$, only a single stable fixed point exists.

B.2 Existence of a Cusp Point

Suppose that a *one*-dimensional parameter(α) dependent system

$$\frac{dx}{dt} = f(x;\alpha), x \in \Re^1, \alpha \in \Re^m \tag{B.4}$$

with smooth function f, has an equilibrium at $x = 0$ for $\alpha = 0$, and let $f_x(0;0) = 0$ and $f_{xx}(0;0) = 0$ hold. Further, assume that the non-degeneracy conditions (e.g., $f_{xxx}(0;0) \neq 0$) are satisfied. Then the system undergoes a *cusp* bifurcation at $x = 0$ [100].

We prove that such a cusp bifurcation is encountered in our system (i.e., Eq. B.1) at $p_A = p_B = p_c$ as we move along the diagonal in parameter space ($p_A = p_B$). Note that our system is two-dimensional. To be able to use the above theory, we first need to reduce the dimensionality of our system. The Center Manifold Theorem [154] guarantees the existence of a one-dimensional center manifold to which we can restrict our system, and such a system preserves the same behavior as the original system in the vicinity of the steady-state under consideration. Once we get the restricted system, we can perform the usual bifurcation analysis in one-dimensional system. Following this idea, we first shift the coordinates such that the origin is located at the critical point we found from the $p_B = p_A$ case (for simplicity, we denote p_A by p and p_B by r), i.e., $(x_0, y_0; p_0, r_0) = (0.2565, 0.2565; \sqrt{10}-3, \sqrt{10}-3)$. In the shifted coordinates, the eigenvalues and eigenvectors are given by $\Lambda = [0; -2.1623]$ and $V = [-0.7071, 0.7071; 0.7071, 0.7071]$. Using transformation $[\tilde{x}\ \tilde{y}]^T = V[x\ y]^T$ and after some algebraic manipulations, we obtain in the new co-ordinate system:

$$\begin{aligned}
\frac{d\tilde{x}}{dt} &= 0.7071(1.5p + 0.2434)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad -0.7071(1.5r + 0.2434)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad -0.7071(0.7071\tilde{x} + 0.7071\tilde{y} + 0.2566)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad -0.7071(p + 0.1623)(0.7071\tilde{x} + 0.7071\tilde{y} + 0.2566) \\
&\quad +0.7071(0.7071\tilde{y} - 0.7071\tilde{x} + 0.2566)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad +0.7071(r + 0.1623)(0.7071\tilde{y} - 0.7071\tilde{x} + 0.2566) \\
\frac{d\tilde{y}}{dt} &= -0.7071(1.5r + 0.2434)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad -0.7071(1.5p + 0.2434)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad -0.7071(0.7071\tilde{x} + 0.7071\tilde{y} + 0.2566)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad -0.7071(p + 0.1623)(0.7071\tilde{x} + 0.7071\tilde{y} + 0.2566) \\
&\quad -0.7071(0.7071\tilde{y} - 0.7071\tilde{x} + 0.2566)(p + r + 1.414\tilde{y} - 0.1623) \\
&\quad -1.414(0.7071\tilde{x} + 0.7071\tilde{y} + 0.2566)(0.7071\tilde{y} - 0.7071\tilde{x} + 0.2566) \\
&\quad -0.7071(r + 0.1623)(0.7071\tilde{y} - 0.7071\tilde{x} + 0.2566) \\
&\quad +1.414(p + r + 1.414\tilde{y} - 0.1623)^2.
\end{aligned} \tag{B.5}$$

Next, we use a quadratic approximation for the center manifold of the above system [154] i.e., we assume $\tilde{y} = h(\tilde{x}) = \frac{1}{2}w\tilde{x}^2$. We can find w by comparing two expressions obtained for $\frac{d\tilde{y}}{dt}$; the first is obtained by using $\frac{d\tilde{y}}{dt} = \frac{d\tilde{y}}{d\tilde{x}}\frac{d\tilde{x}}{dt}$ and then using the first equation in Eq. B.5 and the quadratic approximation for $\tilde{y}$; the second is obtained by direct substitution of the quadratic approximation into the second equation in Eq. B.5. Doing this yields:

$$\tilde{y} = h(\tilde{x}) = -0.7071\tilde{x}^2/(4p + 4r - 2.1620).$$

Hence we obtain the following one dimensional system restricted to the one-dimensional center manifold:

$$\begin{aligned}
\frac{\partial\tilde{x}}{\partial t} &= 0.1814r - 0.1814p - \tilde{x}(1.5p + 1.5r) \\
&\quad +0.7071(1.5p + 0.2434)(p + r - 0.1623) \\
&\quad -0.7071(1.5r + 0.2434)(p + r - 0.1623) \\
&\quad -\tilde{x}^2(0.7071(1.5p + 0.2434)/(4p + 4r - 2.1623) \\
&\quad -0.7071(1.5r + 0.2434)/(4p + 4r - 2.1623) \\
&\quad -0.7071(p + 0.1623)/(8p + 8r - 4.3246) \\
&\quad +0.7071(r + 0.1623)/(8p + 8r - 4.3246)) \\
&\quad +\tilde{x}^3/(4p + 4r - 2.1623).
\end{aligned} \tag{B.6}$$

It is easy to check that the origin in this transformed system satisfies the necessary conditions for a cusp bifurcation. The origin of this transformed system corresponds to the point $p_A = p_B = p_c$ in our original system Eq. B.1. Thus, the system undergoes a cusp bifurcation at $p_A = p_B = p_c$ where $p_c = \sqrt{10} - 3 \approx 0.1623$.

B.3 Mapping out the Bifurcation Curves

In order to map out the first-order transition line (bifurcation curve) we adopt a semi-analytical approach. We assume $p_B = cp_A$ with $c < 1$ to obtain the lower bifurcation curve (symmetry of the system allows us to obtain the upper bifurcation curve, given the lower one). Using Eqs. B.1, the fixed point condition becomes (for simplicity, we denote p_A by p):

$$\begin{aligned}
f(x, y, p) &\equiv -xy + (1 - x - y - (1 + c)p)^2 + x(1 - x - y - (1 + c)p) \\
&+ \frac{3}{2}p(1 - x - y - (1 + c)p) - cpx = 0 \\
g(x, y, p) &\equiv -xy + (1 - x - y - (1 + c)p)^2 + y(1 - x - y - (1 + c)p) \\
&+ \frac{3}{2}cp(1 - x - y - (1 + c)p) - py = 0.
\end{aligned}$$

In addition, for a fold bifurcation, we also require that the stability matrix has an eigenvalue with zero real part. Since, the valid solutions in our case are always real, this is equivalent to requiring the determinant of the stability matrix to be zero. Thus the condition $|Q| = 0$ (with Q given by Eq. B.3) along with Eqs. B.7 enable us to determine for a given c, the location (p_A, cp_A) at which the bifurcation occurs. By numerically solving these equations for different values of c, $0 < c \leq 1$ at intervals of 0.1, we obtain the lower bifurcation curve shown in Fig. 3.1 of the main text.

B.4 Optimal Fluctuational Paths, the Eikonal Approximation and Switching Times between Co-existing Stable States

The master equation for our system takes the general form [155]:

$$\frac{\partial P(\mathbf{X},t)}{\partial t} = \sum_{\mathbf{r}} \Big[W(\mathbf{X}-\mathbf{r},\mathbf{r})P(\mathbf{X}-\mathbf{r},t) - W(\mathbf{X},\mathbf{r})P(\mathbf{X},t) \Big].$$

where $\mathbf{X} = [N_A \ N_B]^T$ denotes the (macro) state of the system as vector whose elements are the numbers of uncommitted nodes in state A and B respectively, $W(\mathbf{X},\mathbf{r})$ is the probability of the transition from $\mathbf{X}$ to $\mathbf{X}+\mathbf{r}$, and $\mathbf{r}$ runs over the allowed set of displacement vectors in the space of macro-states. For our system, $\mathbf{r}$ runs over $[1\ 0]^T, [0\ 1]^T, [2\ 0]^T, [0\ 2]^T, [-1\ 0]^T, [0\ -1]^T$. The deterministic equations can be derived from this master equation and yield:

$$\frac{d\mathbf{X}_{\rm det}}{dt} = \sum_{\mathbf{r}} \mathbf{r} W(\mathbf{X}_{\rm det},\mathbf{r}).$$

The Wentzell-Friedlin theory [103, 156] assumes that for any path $[\mathbf{X}]$ in configuration space:

$$\mathcal{P}([\mathbf{X}]) \sim \exp(-\mathcal{S}([\mathbf{X}]))$$

with $\mathcal{S}([\mathbf{X}^*]) = 0$ for the deterministic path $[\mathbf{X}^*]$. It follows that the dominant contribution to the probability of a fluctuation that brings the system to state $\mathbf{X}$ starting from a stable state $\mathbf{X}_m$ can be written as:

$$\mathcal{P}(\mathbf{X}|\mathbf{X}_m, t=0) = \exp(-S(\mathbf{X})), \tag{B.7}$$

where

$$S(\mathbf{X}) = min_{[\mathbf{X}]:\mathbf{X_m}\rightarrow\mathbf{X}} \mathcal{S}([\mathbf{x}]), \tag{B.8}$$

where the minimization is over all paths $[\mathbf{X}]$ starting at $\mathbf{X}_m$ and ending at $\mathbf{X}$. For $\mathbf{X}$ far away from the steady state, the probability of occupation $P(\mathbf{X})$ is equivalent to logarithmic accuracy to the probability of the most likely fluctuation, $\mathcal{P}(\mathbf{X}|\mathbf{X}_m, t = 0)$ that brings the system to $\mathbf{X}$. The assumption of the form given by Eq. B.7 for

the occupation probability is known as the eikonal approximation.

Using a smoothness assumption for $W(\mathbf{X}, \mathbf{r})$, and since the changes in numbers of A and B nodes are $O(1)$, we can neglect the difference between $W(\mathbf{X} - \mathbf{r}, \mathbf{r})$ and $W(\mathbf{X}, \mathbf{r})$. With this approximation, the eikonal form for the occupation probabilities in the master equation yields the following equation for $S(\mathbf{X})$ [98]:

$$H\left(\mathbf{x}, \frac{\partial s}{\partial \mathbf{x}}\right) = 0, \tag{B.9}$$

where

$$H(\mathbf{x}, \mathbf{p}) = \sum_{\mathbf{r}} w(\mathbf{x}, \mathbf{r})(\exp(\mathbf{rp}) - 1) \tag{B.10}$$

and

$$\mathbf{x} = \mathbf{X}/N, \quad w(\mathbf{x}, \mathbf{r}) = W(\mathbf{X}, \mathbf{r})/N, \quad s(\mathbf{x}) = S(\mathbf{X})/N.$$

Eq. B.9, is analogous to a Hamilton-Jacobi equation for the action of a system with Hamiltonian given by Eq. B.10. The corresponding Hamilton equations of motion for components of position $\mathbf{x}$ and momentum $\mathbf{p}$ are:

$$\dot{x_i} = \frac{\partial H}{\partial p_i}, \quad \dot{p_i} = -\frac{\partial H}{x_i} \tag{B.11}$$

with $s(\mathbf{x})$ playing the role of the classical action:

$$s([\mathbf{x}]) = \int_{[\mathbf{x}]} L(\mathbf{x}, \mathbf{p}) d\mathbf{x} = \int_{[\mathbf{x}]} \mathbf{p}\dot{\mathbf{x}} d\mathbf{x},$$

where $[\mathbf{x}]$ denotes a particular path obeying the equations of motion (Eqs. B.11).

Following this Hamiltonian formulation to characterize the fluctuational paths of the system, our goal is to find the path with minimum action that reaches the separatrix in phase space of the deterministic motion, starting from the vicinity of the stable state under consideration [98, 103]. Arguments in [157] show that the fluctuational path reaching the separatrix with the minimal value of the action, is the path that passes through the saddle point. This is the *optimal escape path*, i.e., the path whose probability of occurrence dominates the probability of escape and we denote it by $[\mathbf{x}_{\text{opt}}]$. This path can be found by integrating the equations of motion Eq. B.11, and finding the required path that starts from the vicinity $\mathbf{x}_m$ to the saddle point $\mathbf{x}_{\text{saddle}}$. Thus following Eqs. B.7, B.8 we have for the probability of escape from the current stable point in which the system is trapped:

$$P_{\text{escape}} = P(\mathbf{x}_{\text{saddle}}) \sim \exp[-Ns(\mathbf{x}_{\text{saddle}})], \tag{B.12}$$

where

$$s(\mathbf{x}_{\text{saddle}}) = \int_{[\mathbf{x}_{\text{opt}}]} L(\mathbf{x}, \mathbf{p}) d\mathbf{x}$$

and the transition time (or time to escape from the steady state) follows:

$$T_{\text{switching}} \sim \exp[Ns(\mathbf{x}_{\text{saddle}})]. \tag{B.13}$$

In practice we start from some point $\mathbf{x}$ in the vicinity of the stable state, and to obtain the corresponding momenta $\mathbf{p}$ and action $s(\mathbf{x})$, we employ a Gaussian approximation [98]:

$$S(\mathbf{x}) = \sum Z_{ij}(x_i - x_i^m)(x_j - x_j^m),$$

where Z satisfies an algebraic Ricatti equation:

$$\mathbf{QZ}^{-1} + \mathbf{Z}^{-1}\mathbf{Q}^T + \mathbf{K} = 0,$$

where Q is the linear stability matrix (Eq. B.3) evaluated at $\mathbf{x}_m$ and

$$K_{ij} = \sum_{\mathbf{r}} w(\mathbf{x}_m, \mathbf{r}) r_i r_j.$$

Solving this Ricatti equation yields Z which in turn yields $S(\mathbf{x})$ and $p(\mathbf{x})$.

In order to find the optimal fluctuational path of escape from a given steady state, we numerically generate fluctuational paths from various points close to the steady state (we explore points at intervals of 10^{-5} along the x_1 dimension and 10^{-2} along the x_2 dimension around the steady state) and find one that passes close enough (no greater than a distance of 10^{-5}) to the saddle point. The equations of motion, Eqs. B.11, are integrated using a trapezoidal rule to generate these paths starting with initial conditions obtained using the Gaussian approximation described above and subsequent numerical solution of the Ricatti equation (we use a Matlab Ricatti equation solver for the latter). The scaling behavior of switching times obtained using this approach for various committed fraction values as a function of distance from second-order transition (or cusp) point are shown in Fig. 3.4 of the main text.