Analyzing Information Flow in Social Networks for Knowledge Discovery

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Nishith Pathak

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Prof. Jaideep Srivastava & Prof. Arindam Banerjee

February, 2013

Acknowledgements

The pursuit of my doctorate has been a long and eventful journey. Most of all, it has been a growing experience, along many different facets, influenced by all the wonderful people who have been a part of my life over the last few years.

First and foremost I want to acknowledge the role of my advisor Prof Jaideep Srivastava. Prof. Srivastava has always been a strong force of support, guidance and inspiration. His depth of experience and knowledge, along with the ability to step back and look at the "larger picture" has proved invaluable for my research as well as shaping my own thought process. His guidance as an academic advisor and a mentor always ensured continued progress and helped deal with the more difficult moments. Apart from being one of the technology leaders in his field, Prof. Srivastava is also a wonderful advisor and teacher.

I would also like to thank my co-advisor Prof. Arindam Banerjee. Prof. Banerjee's depth of knowledge and technical expertise has always inspired me to push the limits of my own technical knowledge. His persistent, thorough and rigorous approach to problem solving has had a strong influence in shaping me as a researcher. It is easy to see his enthusiasm and passion towards his research, which is reflected in the outstanding quality of his work.

A special mention of Prof. Vipin Kumar, who worked with me during my internship at the Army Center in the summer of 2004, which ultimately culminated in the opportunity for me to pursue graduate school at the University of Minnesota. Prof. Kumar was also a member of my examination committee and has always provided valuable feedback on my research. I want to thank my committee member Prof. Xiatong Shen, who provided important feedback on various aspects of my thesis. I also thank Prof. Shashi Shekhar, who during his brief interaction with me, provided very useful advice

on being a better researcher. A special thanks to Georgane Tolaas who has always been very supportive and approachable regarding every administrative aspect of graduate school.

I would like to thank all the wonderful colleagues I have met over the course of graduate school: Prasanna Desikan, Sandeep Mane, Colin DeLong, Jim Kang, Gaurav Pandey, Shyam Boriah, Jaya Kawale, Kyong Shim, Muhammad Ahmad Aurangzeb, Senthil Krishnamurthly, Zoheb Borbora, Nisheeth Srivastava, Atanu Roy, Chandrima, Karthik S, Komal Kapoor and other members of the DMR lab. The many discussions and feedback have had an immeasurable contribution towards my work.

Over the course of my doctorate, I have had the pleasure of working with collaborators from other universities and fields from the social science domain. I would like to thank my collaborators from the Virtual Worlds Expoloratorium project, partucularly, Prof. Noshir Contractor, Prof. Dmitri Williams, Prof. Scott Poole, Robbie Ratan, Tracy Kennedy, Yun Huang, Annie Wang and Cindy.

Finally, I want to acknowledge my family's support, love and encouragement. They have always been there to motivate me to push my limits and pick myself up when I am down. The many hours my parents persisted with tutoring me throughout my childhood helped develop a strong academic foundation that has been very instrumental in my professional success.

Dedication

This thesis is dedicated to my wonderful parents, Mr. Nagesh Pathak and Mrs. Neerja Pathak, as well as my brother, Ksitij Pathak, who have always been a force of encouragement and support in my life.

Abstract

In the last few years the online world has seen a surge in users' social behavior. No longer is the image of a lone user surfing the web relevant anymore and with social sites such as Facebook, Twitter, etc. online users can now actively interact with other users. It is now quite common for web businesses to offer support for friends lists, forums, private message systems, community maintenance tools etc. As as result, not only are users finding more social satisfication online, but the businesses themselves are now able to interact with and monitor the communities around them. Consequently, large amounts of data are being collected from such "social systems", which capture users' participation in the community. The data can include user-user interactions as well as their activities with time stamps. The data is also unique in that it captures complex social phenomenon in a much more comprehensive manner and at a much more finer granularity, than any other traditional source of communication data. This presents rich opportunities for the development of knowledge discovery algorithms which will find immense value in revealing trends, latent structures or interesting behaviors in these social systems.

In any social system, communication exposes people to information, opinions as well as behavior of other users. According to a well studied phenomenon in social science, summarized in the theory of contagion, users in such networks tend to develop beliefs, attitudes and assumptions that are similar to those of others around them. By "word-of-mouth" rumors, ideas, opinions, information, etc. can propagate to different regions in the network. The research presented in this thesis explores the analysis of such information flow in social networks from a variety of perspectives, including the network topology, actors' interests, actors' cognition and actors' influence. It is shown that the proposed analyses techniques can discover valuable knowledge regarding community structure, user interests and sentiments, as well as prominent users in the community. Such knowledge is of immense value to online business owners, as it allows them to monitor and identify factors for improving the overall experience of their users.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Online Social Networks

Recently the online industry has seen a steep increase in demand for social satisfaction from users. Popularity of websites such as Facebook, Twitter, MMO games such as World of Warcraft etc. have demonstrated that being online can be a social experience. Users are no longer a bunch of solo web surfers littered across the internet. Instead, they can now interact with each other in a variety of ways. Message boards, friends lists, likes/dislikes, community maintenance tools etc. have all become staple features of any online business. Consequently, large volumes of data on users in these social ecosystems became available for researchers to study. This data is unique in that it captures a comprehensive picture of every users' behavior at a fine granularity (time stamped activity logs). Availability of such data has prompted quite a lot of interest in research on social networks.

A *social network* is a social structure of individuals, who are related (directly or indirectly to each other) based on a common relation of interest (e.g. friendship, trust, etc.). In any social network, it is not possible for everyone to be connected to everyone else, nor is it desirable ([1]). Thus, *social network analysis* is the study of social networks to understand their structure and behavior. It can involve various analyses such as identifying undesirable structures in a social network. Examples of such social structures ([2]) include imploded relationships (more frequent inter-group communication than intra-group communication), holes (missing communication between people

who are expected to communicate) and bow-ties (high dependency on a single person for communication within a group, hence a source of bottleneck).

Social network analysis is the field which studies social networks, as well as develops algorithmic techniques to do so. It has been widely used in different application frameworks from product marketing to search engines to understanding organizational dynamics. An example application is the evolving field of study called viral marketing. It is a widely accepted fact that word-of-mouth promotions play a significant role in marketing ([3]). Social network analysis enables us to understand which individuals play an important role in such word-of-mouth marketing and thus helps to target those individuals when marketing new products. Domingos and Richardson ([4]) introduced the concept of network value of a customer, which measures the expected profit from sales to customers (and so on recursively), influenced by that particular customer. Another example of application of social network analysis is the Google (`http://www.google.com`) search engine. The main philosophy behind the rankings of web pages generated by this search engine is that a webpage (created by an individual) has links to other web pages (created by other individuals) which are related to the former webpage. Thus, web pages form a social network based on relevance relationship. Social network analysis is an active field of research in sociology, anthropology and recently in computer science.

There are two different schools of thought in the analysis of social networks, viz, *sociocentric (whole) network analysis* and *egocentric (individual) network analysis*. Sociocentric analysis has its roots in sociology, where the focus is to study the whole social network and to find global social patterns. This analysis usually is carried out by first well-defining the set of individuals of interest and then quantifying the relationships between them. Sociocentric analysis thus focuses on the structural relationships in the network. Two techniques are mostly used for this analysis – graph based and statistics based. Network visualization may also be used (using either a graph-based or a statistics-based approach) but it is usually easy to do so only with small networks. Most of the social network analysis studies till now have focused on sociocentric analysis. Egocentric analysis comes from anthropology, where the focus is more on the individual rather than the whole groups of individuals. This analysis begins by quantifying the relations between the individual of interest (called *ego*) and all other individuals (called *alters*) related (directly or indirectly) to the ego. Each such social network considered

from an egos perspective is termed here as a *personal social network* or *ego-centered networks*. Usually several such personal social networks are analyzed and then generalizations of patterns in all these networks are made. Note that ego-centered networks is also sometimes used to refer to socio-cognitive networks i.e. what does an ego think about the social ties between him/herself and alters as well as the ties among the alters. However, it is a broader concept as it is also used in the context of analyzing the actual social ties among egos and alters. Usually several such personal social networks are analyzed and then generalizations of patterns in all these networks are made. Graph-based techniques are usually used for personal social network analysis. The difficulty as well as high costs to collect data for several such personal social networks have traditionally caused such studies to be infrequent.

Social network analysis has a theoretical as well as a methodological perspective. In terms of theory, social network analysis complements and extends current social science methods by focusing on causes and implications of relationships between individuals and among groups of individuals. From methodological perspective, social network analysis quantifies the relationships between people, which allow the use of models and techniques from natural and social sciences.

1.2 Constructing Social Networks

An important question in social network analysis that needs to be explained is how does one construct the social network. We first discuss the classical techniques for gathering social interactions data, and then discuss how new kinds of data collected by computer networks will lead to new methods for constructing and analyzing social networks.

1.2.1 Classical approaches for constructing social networks

In classical social network approaches, data are collected for different interactions from individual actors, pairs of actors (or dyads), triples of actors (or triads), a group (subset) of actors or the network as a whole. Data about individuals are usually collected by observing, questioning and/or interviewing the actors. For dyads, actors are questioned about their relationships to other actors. For larger groups (subsets) of actors,

social data are collected by observing actors affiliations to and behavior at certain social events (e.g. meetings). Thus, social network data are gathered in a variety of ways like questionnaires, interviews, observations, archival records, experiments and other techniques like ego-centered, small world or diaries (see [5] for detailed description on each of these techniques). For sociocentric analysis, a common method of constructing whole social network is to collect such data from several actors and then assimilating the information to obtain the whole network. This is illustrated in Figure 1.1 as synthesis of actors surveys to construct the whole social network. To obtain a *cognitive social structure* ([6]), information is collected about actors perceptions of other actors network ties and then assimilated to obtain such a social structure. In addition to static information about social ties, a researcher may also be interested to study how a social network changes over time. Two such questions of interest are, (i) how has the social (or socio-cognitive) network changed over time; and (ii) how well can the past predict the future. Longitudinal social network data for such analysis are collected using same techniques (like questionnaire and interviews) as discussed before.

However, data from such surveys are usually affected by two inherent problems (i) Error in the perceptions of the actors about the (social) relationships between actors, and (ii) Bias in information about social relationships reported by the actors (e.g. actors may lie about their relationships). Thus, the *true structure* of a social network is defined as the relatively prolonged and stable pattern of interpersonal relations, whereas the *observed structure* is obtained from the measured social network data. A lot of discrepancies may be present between the two due to issues of validity, reliability and measurement errors in social network data. *Accuracy* of the data collected is of prime concern; since in many studies, the social network data are collected using interviews and questionnaires. Considerable research effort is directed to reduce inaccuracy in data by developing/applying better data collection designs for questionnaires. Another problem occurs when actors in networks are organizations whereas data are collected from individuals, who are assumed to have complete knowledge of the network. To address this, techniques that address *validity* are used, which measure the extent to which a concept actually measures what it is intended to measure. For example, for measuring friendship between actors, instead of just asking who is a friend of whom, the researcher

may have to come up with several theoretical situations which measure different degrees of friendship between the actors. *Reliability* of a variable (or a concept) measures whether repeated measurements give same estimate for the variable. *Measurement error* occurs when the true value of a relationship differs from its observed (measured) value. [7] reviews different methods for addressing problems in collection of social network data.

1.2.2 Computer Networks as Social Networks

The emergence of computer networks allows collection of data on social interactions with less measurement errors. An important motivation for computer networks (and Internet) to come into existence was to foster collaborative work between geographically dispersed researchers. These computer networks have now turned into an infrastructure that supports social networks - connecting people, organizations as well as knowledge ([8]). The widespread use of internet and the growing online community of users have enabled the formation of social networks based on different relations of interest, e.g., Usenet, a widely used online newsgroup, had more than 80,000 topic-oriented discussion groups (or social networks) in 2000. Such an infrastructure allows individuals to form geographically dispersed social networks, but these social networks are usually loosely bounded. On the other hand, computer networks facilitate an actor to participate in multiple social communities (networks), thus enabling the actor to know many more actors and increase his/her social capital.

With development and widespread use of network-based software for applications like instant messaging (e.g. AOL Instant Messenger `http://www.aim.com`), e-mail communication and online social communities (e.g. Facebook `http://www.facebook.com/` and Twitter `http://www.twitter.com`), gigabytes of data about online social communities are now being logged in data warehouses. Such data provides an unbiased view of communication between actors, while bringing in new challenges to develop efficient computation techniques for such analyses. Information gained from such social communities will be used for different applications, from viral marketing to user recommendations.

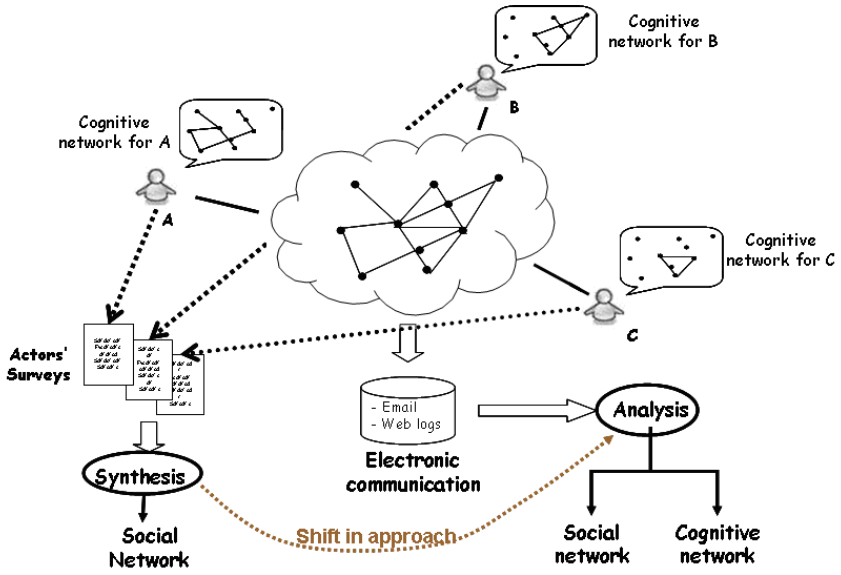


Figure 1.1: Constructing Social Networks

1.2.3 A new Paradigm for Constructing Social Networks

Figure 1.1 illustrates the classical approach as well as proposed approach based on computer-networks data for constructing social networks. Most classical research in social networks focused on gathering social relationship data (e.g. using surveys) from three actors A, B and C. These data provided the actors perspectives regarding of the social network, and were combined (*synthesis* step) by researchers to build the social network for analysis. However, as discussed previously, these data suffer from problems like inaccuracy and bias in actors opinions. Computer networks provide a new mechanism to capture more accurate and less biased data about social behavior of actors. In addition, computer networks provide social networking freedom to actors, allowing the actors social behavior to be not solely based only on other actors who physically surround him/her. For example, if actor A is interested in sports (e.g. soccer) but none of his/her friends (or social contacts) living in his/her city are interested in it, the computer network (e.g. internet) allows A to form/join a social community with similar soccer interests. Analogously, in an organization, computer networks allow employees in different offices to collaborate and/or socialize with each other. Such data can be accurately captured using computer networks.

1.3 Information Flow in Social Networks

In this day and age, it is not enough to simply provide a service and any web business must also have a platform for users to discuss as well as build communities around the commodities of interest. This provides a state of persistency to the user base as well as relevance to these commodities, which in turn is highly beneficial to both, online businesses as well as the users. Users can now maintain friends lists, share their opinions on forums with votes or likes/dislikes, connect with other users via Facebook, Twitter or social games etc. Usually these relationships can be extracted and expressed in the form of social networks.

These social networks expose people to information, opinions as well as behavior of other users and according to the theory of contagion (ref Monge and Contractor), from social science, users in these networks develop beliefs, attitudes and assumption that are similar to those of others around them. By "word-of-mouth" rumors, ideas, opinions, information, etc. can propagate to different regions in the network. In the scientific literature such a process is formally referred to as a cascading process and the commodities being propagated are called cascades. Network topology plays an important role in dictating the process of cascade flow. The more well connected nodes are, the easier it is for a cascade to spread through them. Understanding and analyzing these "word of mouth" processes is very important because it allows one to project, and in some cases even control, the extent and directions of the flow of information, rumors, opinions etc. For example, one can help predict which portions of the market can be captured in viral marketing scenarios, or whom to pick for seeding word of mouth campaigns etc. The research presented in this thesis explores the analysis of such information flow in social networks from a variety of perspectives, including the network topology, actors' interests, actors' cognition and actors' influence.

- **Simulating Multiple Cascades on Networks.** The scenario involving a single cascade propagating in the network has been well studied and this research advances the state of the art by presenting a technique for propagating multiple cascades in a network. The proposed approach is a generalization of the traditional linear threshold model and allows for multiple cascades as well as the ability for nodes to switch between the cascades based upon their neighborhood.

The algorithm infers the different local regions of the network in which different cascades can be endemic and partitions it accordingly. Results show that these partitions are optimal according to the normalized cut measure and the proposed approach finds significant application for discovering information flow based community structure in networks.

- **Social Topic Models.** This analysis involves extracting latent communities from the social network based upon senders, recipients as well as topics of conversation in the emails. Thus, one can learn the different topics of interest and the corresponding information being exchanged among the different communities of users. Bayesian models from the topic modeling domain, from text analysis, are extended to account for social connections in order to infer the desired social-topic based communities.

- **Modeling Socio-cognitive Networks.** Only the sender and recipient information, from communication logs, is used to analyze users' awareness of each others' social networks. The idea is that, the more isolated users/small communities of users are, the less they are aware of social relations outside of their local region of the network. On the other hand, the more knowledgeable users are about others' social connections, the more well connected they are and the more conducive the network is to spreading knowledge, rumors etc. The Enron email data is used for this purpose as it exhibits two distinct behaviors - common organizational communication during times of normal functioning of the firm as well as different communication patterns from the Enron crisis period.

- **Modeling a Cognitive Knowledge Network.** The communication text associated with the emails is analyzed along with sender recipient information. Logical propositions are used to represent knowledge and actor beliefs are represented as probability values in a Dempster-Shafer thery framework. Emails in users' inboxes are analyzed and each user's cognitive state is estimated using evidence combination.

- **Data Driven Influence Value Computation.** Network value of a given user is a quantification of how much influence a user commands in the network. The key

idea is that any cascade seeded from a high network value user will spread much further and capture more important nodes than one seeding from a lower network value user. Traditional approaches for computing network value use theoretical probabilistic models which are suitable for a general setting where the only information available is the network itself. However, there is often a lot of context as well as corresponding data associated with the users and the relationships among them, and in order to account for these the traditional network value computation algorithms are extended to more suitable data driven network value computation counterparts.

With the internet continuing to become an integral part of our lives, it is inevitable that strong social platforms become a must for most online entities that desire to create a persistent user base around them. Knowledge discovery on data generated by these online social environments allows the identification as well as monitoring of factors for improving the overall experience of their users. Tracking and predicting the flow of information, rumors, opinions etc. can also be used to project potential population/sub-population behaviors allowing the interested parties to better prepare for or leverage these expected behaviors. Moreover, identifying influential users from within the community and engaging them helps develop important communication channels to the user base.

Chapter 2

Simulating Multiple Cascades on a Network

2.1 Introduction

One of the more convenient ways for information to travel in the network is by "word-of-mouth" processes. Actors communicate something to their neighbors, who can then do the same with their neighbors and thus, the original message porpagates in the network. Such processes are formally reffered to as cascading processes and the commodity being propagated is called a cascade. Cascading processes are typically used to map out the flow of information along the network topology. Such analysis can provide interesting insights regarding the network topology as higher spreads typically mean that the network topology encourages information flow. This knowledge is valuable for any viral marketing efforts. Cascades need not be limited to information itself. They can generally mean anything from opinions, ideas, awareness, viruses etc. Generally, any commodity than can propagate through network edges falls under the phenomenon of cascading effects.

Existing literature has primarily focused on binary state cascades i.e. either a node is infected or not. In most real world scenarios it is incorrect to assume the existence of a single cascade progressing through the network e.g. in viral marketing, a cascade corresponds to the "word of mouth" awareness of a product and a unary cascade assumes just the existence of this product alone. In a real world scenario it is highly likely that

competing products will respond with their own strategies and vye for a share of the nodes in the network. This particular scenario also raises other interesting questions - with which products are the individual nodes expected to align with? What will be the market share of each product?, what strategies must be used to capture certain segments of the network? or is there a limit on the number of cascades surviving in the network? All the above questions would require models more sophisticated than the current state of the art.

With the above motivation in mind, this research works towards extending the state of the art in cascade modeling. A generalized version of the linear threshold model ([9],[10]) for simulating multiple cascades on a network is presented. The proposed model is a natural extension to the *Liner Threshold model* for handling multiple competing cascades on a network, while allowing nodes to switch back and forth between them. Such scenarios can be used to simulate conditions of actors choosing between mutiple cascade commodities, which could be competing products, ideas etc. Allowing nodes to change their choices will also simulate conditions of actors being able to rearrange their choices based upon the choices of their neighbors. Thus, resulting in a dynamic process which converges to a steady state distribution. The steady state can be analyzed to extract knowledge regarding which cascades are more likely to be endemic in what regions of the network.

A stochastic graph coloring process is used to simulate the process and estimate the highly likely states of cascades' spreads in the newtork. The process is shown to be a rapidly mixing Markov chain, which allows polynomial time approximation algorithms for deducing the steady state distribution.

2.2 Related Work

Broadly two theoretical models of diffusion have been widely studied - the linear threshold model ([9],[10]) and the independent cascade model ([11],[12]). Others consist of Markov random field ([4],[13]) and game theory based models [14]. Existing literature on network diffusion processes and applications for the same is quite rich and a comprehensive review is presented in [15].

Different variations of cascading processes include progressive (once a node is infected it cannot go back to the non-infected state) and non-progressive models (nodes are allowed to switch between infected and non-infected states) with simulation algorithms existing for the progressive as well as finite time non-progressive cases [16]. Traditional independent cascade models allow an infected node only one chance to infect its neighbor, however, recent extensions get around this assumption and provide methods for computing probabilities of nodes being infected at different points in time [17]. Most existing research has focussed on modeling a single cascade and all of these models are limited in that respect.

Recent extensions to the independent cascade model consist of progressive models allowing for multiple cascades. In [18] the authors extend the progressive independent cascade model for scenarios where a second cascade is to be introduced in a network already harboring another competing cascade. The influence maximization problem in this case is one of selecting the optimal seed nodes such that the cascade being introduced captures as many nodes as possible. It is shown to be NP-hard and greedy approximation algorithms are provided for solving the problem. In [19] Budak at al. discuss a similar problem in the context of limiting the spread of misinformation in a network.

Bharati et al. also extend the progressive independent cascade model to handle multiple (≥ 2) cascades [20]. The extended model is based on Hotelling's model of competition [21], which in turn is based on location theory [22] and Voronoi games [23]. Approximation algorithms for computing best response strategies are presented along with a discussion of first mover strategies for maximizing spread against perfect competition. An FPTAS for maximizing influence of a single player for a tree structured graph is also presented.

Kosta et al. use game theory and lcoation theory to model selection of influential nodes for competing cascades as a strategic game [24]. The authors show that computing optimal strategies for the first and second player are NP-complete even in a restricted model. They also show that computing an approximate solution for the first player is NP-complete. Several heuristics are analyzed to conclude that networks in which the second player is at an advanatge can exist. Overall, existing literature regarding competing cascades involves models based on progressive independent cascades and the

influence maximization problem is analyzed as a game among them.

2.3 Methodology

Graph coloring is the problem of labeling vertices, with k colors, so that no two adjacent vertices have the same color. In the context of this thesis, a general definition of graph coloring is considered which includes problems of labeling vertices with k colors subject to a given set of constraints, not necessarily requiring adjacent vertices to have different colors.

2.3.1 Simulating cascades using graph coloring

Consider an undirected graph $G(V, E)$, with non-negative edge-weights $w_{ij} > 0$ for each edge $(i, j) \in E$ and no self-loops $w_{ii} = 0, \forall v_i \in V$. Weight w_{ij} represents the similarity and/or affinity between nodes i and j. The nodes of this graph are colored using $|C|$ number of colors from the set $C = \{c_1, c_2, \ldots, c_{|C|}\}$. The state space S of all possible colorings is of size $|C|^{|V|}$ and each colored state $G^C \in S$ induces a corresponding partitioning P on G.
$P = \{V_1, V_2, \ldots, V_k\}$ such that $V_i \subset V \ \forall V_i \in V$, $V_i \cap V_j = \phi \ \forall i \neq j$ and $\cup_{i=1}^{k} V_i = V$. Nodes in each set V_i are from a maximal subgraph in G^C, such that all nodes in V_i are connected and have the same color. Semantically, each color corresponds to a cascade and P is the set of subgraphs corresponding to the regions in G, in which different cascades are endemic. This is illustrated in figure 2.1 where the sub-figures show, (a) G with three partitions, (b) coloring the graph with cascades induces the same partitioning as (a), note that endemic regions appear as smooth regions of colors, (c) endemic regions with a coloring that merges p_2 and p_3 with the third cascade dying out, (d) another coloring that splits p_1 into two regions, allowing a fourth cascade in the network No two adjacent partitions can have the same color as otherwise they are both considered to be a single partition. G^C to P is a many to one mapping and the state space S can be mapped to the space of all possible partitionings on G that can be expressed using $|C|$ colors. The key issues are defining an optimal solution state and computing it, both of which are resolved using a stochastic graph coloring process.

The coloring process is designed to favor states in which (i) nodes in close knit-regions

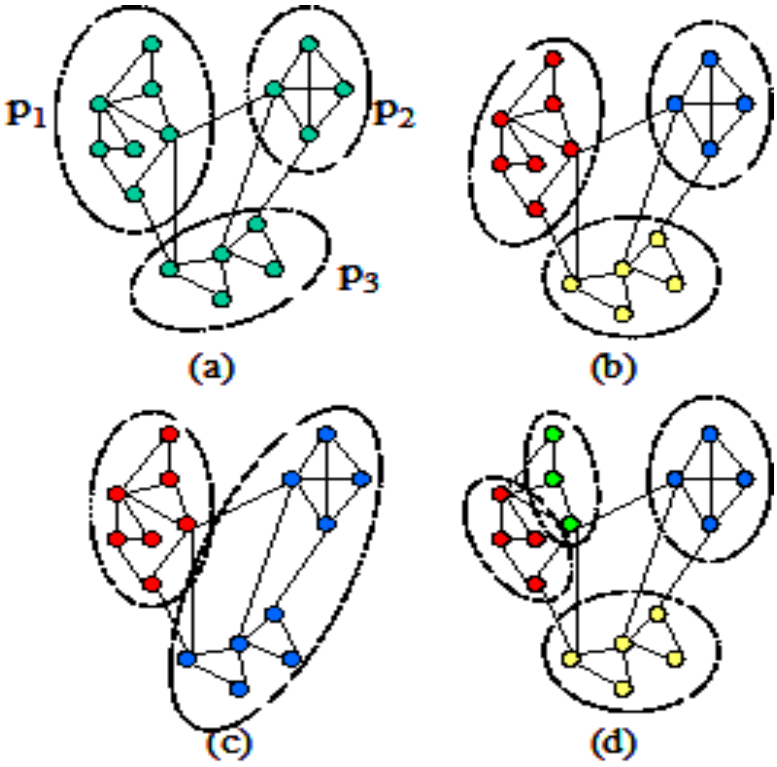


Figure 2.1: Partitions induced by coloring a graph

of the graph have the same color, (ii) no two such close-knit regions with sufficiently low connectivity between them have the same color. Consider the following process: In each step, a node $v_i \in V$ is sampled uniformly and for v_i, a color c_p is sampled. If G^C is the current coloring of the graph and $n^i_j \in N(v_i)$ is the jth neighbor of v_i, then probability of assigning color c_p to node v_i, given colors for all other nodes is,

$$p_{v_i}(c_p|G^C_{-v_i}) = \frac{\sum_{n^i_j \in N(v_i)} \delta_{n^i_j}(c_p) w_{in^i_j}}{\sum_{n^i_j \in N(v_i)} w_{in^i_j}} \tag{2.3.1}$$

In (1) $\delta_{n^i_j}(c_p) = 1$ if node n^i_j is colored with c_p and 0 otherwise.

The process described is a Markov chain and the state space is the set of all possible colorings of G using $|C|$ colors. Note that this process corresponds to the traditional linear threshold model, where probabilities from neighbors having a certain color c_p are summed up and if this sum exceeds a stochastic threshold from the range $(0, 1)$, the node under consideration is colored with c_p. While this method favors assigning the same color to nodes close together, three problems make it unsuitable for handling the multiple cascade scenario: (i) The trivial state of coloring all nodes with the same color is favored, (ii) if a state where some color c_p is not assigned to any node is reached, then the chain can never transition to a state where any of the nodes have color c_p. Thus, the chain can have transient states and is not ergodic, and (iii) given the search space size $|C|^{|V|}$, it is difficult to say how many iterations are required to reach the steady state. The next sub-section presents a small modification that addresses all of these problems.

2.3.2 A generalized linear threshold model for multiple cascades

Consider the graph $G(V, E)$ and set of colors C. In each step, node v_i is sampled according to a fixed distribution J and a color c_p is sampled for v_i. Probability of sampling color c_p for v_i, given colors for all other nodes is,

$$p_{v_i}(c_p|G^C_{-v_i}) = \frac{\beta}{|C|} + (1-\beta)\frac{\sum_{n^i_j \in N(v_i)} \delta_{n^i_j}(c_p) w_{in^i_j}}{\sum_{n^i_j \in N(v_i)} w_{in^i_j}} \tag{2.3.2}$$

In the above equation, $\beta \in (0, 1)$. This is also a Markov chain with state space S. Semantically, this change can be interpreted as there being a chance β, of node v_i ignoring its neighbors and randomly picking a color from a uniform distribution.

The number of colored states corresponding to p par- titions, is greater than the number corresponding to every $p' < p$ partitions. Consequently, the first term in the RHS of 2.3.2 favors colorings with more partitions, thus, acting as a regularizer against the second terms bias to merge them.

Note that (i) there are a finite number of states ($|S| = |C|^{|V|}$), (ii) every state is reachable from every other state, and (iii) the chain has aperiodic states. This makes the process in Eqn 2.3.2 irreducible with aperiodic states and hence, ergodic in nature. Therefore, it will converge to a stationary state distribution for any undirected graph with non-negative edge-weights and no self-loops.

The most important feature is that it is rapidly mixing, which allows us to place an upper bound on the number of steps required for the chain to reach steady state [25].

Lemma 1. For a given undirected graph $G(V, E)$ with non-negative edge-weights and no self-loops, if the Markov chain proposed in Eqn. 2.3.2 takes $t(\epsilon)$ number of steps to reach the steady state distribution then,

$$t(\epsilon) \leq (\frac{W}{w_{\min}\beta} \log \frac{|V|}{\epsilon})$$

In the above equation, W is the sum of weighted degrees of all nodes, $W = \sum_{\forall v_i \in V} w_{v_i}$ with $w_{v_i} = \sum_{j \in N(v_i)} w_{ij}$, $w_{\min}$ is the minimum weighted degree in the graph excluding isolated nodes, $w_{\min} = \min_{v_i \in V, w_{v_i} > 0}(w_{v_i})$ and ϵ is the error between the estimated steady state distribution (i.e. distribution followed by (2) after $t(\epsilon)$ steps) and the true steady state distribution measured in terms of statistical variation.

For any two distributions D_1 and D_2, on the same state space Ω, statistical variation is denoted by $||D_1 - D_2||$ and is given by $\frac{1}{2} \sum_{\forall i \in \Omega} |p_1(i) - p_2(i)|$, where $p_1(i)$ and $p_2(i)$ are the probabilities for the ith state in D_1 and D_2 respectively. For a given number of colors $|C|$, state space S is the set of all possible colorings on G.

Proof. Consider two Markov chains M_X and M_Y, both coloring the same graph G. In each step both chains pick the same node $v_i \in V$ according to a fixed distribution J and each chain samples a new color for it. M_X picks a new color $c^x_{v_i}$ for v_i according to distribution D_{X,v_i} and M_Y uses distribution D_{Y,v_i} to sample color $c^y_{v_i}$ for the same node. Let $\kappa_{s,v}$ denote the distribution, according to Eqn 2.3.2, for sampling a color for node v given the current state of G, $s \in S$.

Define distribution $D_{X_t,v_i} = \kappa_{X_t,v_i}$ and if c is the color picked by D_{X_t,v_i}, distribution D_{Y_t,v_i} picks color c' such that with probability $\min\{1, \kappa_{Y_t,v_i}(c)/\kappa_{X_t,v_i}(c)\}$ $c' = c$, otherwise, sample c' according to the distribution,
$D(c_s) = \frac{\max\{0,\kappa_{Y_t,v_i}(c_s)-\kappa_{X_t,v_i}(c_s)\}}{||\kappa_{Y_t,v_i}-\kappa_{X_t,v_i}||}$.

Note that if M_Y is observed by itself, independent of M_X, then it appears to be following Eqn 2.3.2. Thus, a coupling between M_X and M_Y is defined.

Assume M_Y is following the true steady state distribution and states X_t and Y_t (both at some time t), from M_X and M_Y respectively, differ only in the color of a single vertex $v_q \in V$. If Δ_t is the number of nodes having different colors in X_t and Y_t, then $\Delta_t = 1$. According to the path-coupling lemma [25], in order to prove that Eqn 2.3.2 is rapidly mixing it is sufficient to show that the maximum possible value for $E[\Delta_{t+1}]$ is less than one i.e. $\gamma = \max_{X_t,Y_t \in S, v_q \in V} E[\Delta_{t+1}] < 1$. Moreover, if the above condition holds then then the mixing time will be $t(\epsilon) \leq log(|V|\epsilon^{-1})/(1-\gamma)$.

We have, $E[\Delta_{t+1}] = 1 - P(\Delta_{t+1} = 0|X_t, Y_t) + P(\Delta_{t+1} = 2|X_t, Y_t)$
which gives us,

$$\gamma = \max_{X_t,Y_t \in S, v_q \in V} \{1 - J(v_q) + \sum_{v_j \in V} J(v_j)||D_{X_t,v_j} - D_{Y_t,v_j}||\}$$

Since states X_t and Y_t differ only on one vertex v_q, $||D_{X_t,v_i} - D_{Y_t,v_i}|| = 0$ for all vertices $v_i \in V$ except neighbors of node v_q. $||D_{X_t,v_l} - D_{Y_t,v_l}|| = \frac{(1-\beta)w_{lq}}{w_l}$ for all nodes $v_l \in N(v_q)$. If we fix distribution J such that probability of sampling node v_i is $\frac{w_{v_i}}{W}$ for

all nodes $v_i \in V$ then we have,

$$\begin{aligned}
\gamma &= \max_{X,Y \in S, v_q \in V} \{1 - J(v_q) + \sum_{v_l \in N(v_q)} J(v_l) ||D_{X,v_l} - D_{Y,v_l}||\} \\
&= \max_{X,Y \in S, v_q \in V} \{1 - \frac{w_q}{W} + \sum_{v_l \in N(v_q)} \frac{w_l}{W} \frac{(1-\beta) w_{lq}}{w_l}\} \\
&= \max_{X,Y \in S, v_q \in V} \{1 - \frac{w_q}{W} + \sum_{v_l \in N(v_q)} \frac{(1-\beta) w_{lq}}{W}\} \\
&= \max_{X,Y \in S, v_q \in V} \{1 - \frac{w_q}{W} + \frac{(1-\beta) w_q}{W}\} \\
&= \max_{X,Y \in S, v_q \in V} \{1 - \beta \frac{w_q}{W}\} \\
&= 1 - \beta \frac{w_{min}}{W}
\end{aligned}$$

Since, $0 < \beta < 1$ we have $\gamma < 1$. Thus the coloring process reaches steady state in time $t(\epsilon) \leq (\frac{W}{w_{\min} \beta} \log \frac{|V|}{\epsilon})$. **Q.E.D**

2.3.3 Computing multiple cascade simulation solutions on a graph

Computing the likely configurations different cascades can take in G is done by estimating the most likely colorings from the steady state distribution. This is done using simulated annealing [26]. The complete procedure, called *StochColor* (Stochastic Coloring) is presented in Algorithm 1.

The colored graph at the end of simulated annealing is the estimated ML state from which the endemic regions are extarcted as a partitioning of G using Algorithm 2. Thus, the coloring process is used to estimate a solution state for simulating C colors on G, with time complexity $O((\frac{W}{w_{\min} \beta} \log \frac{|V|}{\epsilon} + |V| \frac{\log T_f}{\log \alpha})(\frac{|E|}{|V|} + |C|) + |E|)$. The algorithm can be used as is even in the presence of seed nodes for one or more cascades as long as it is ensured that corresponding colors for the seed ndoes are persistent during the simulation (the rapidly mixing result continues to hold in this case).

2.3.4 Parameter Settings

The simulation process can be controlled using the two parameters - number of cascades/colors ($|C|$) and β, however, empirically it was observed that results from *StochColor* are surprisingly robust even over large changes in them and the graph topology

Algorithm 1 StochColor(G,$|C|$,β,α,T_f,ϵ)

Inputs: Graph $G(V, E)$ with non-negative edge-weights, number of colors k, simulated annealing parameters α (cooling rate), T_f (final temperature) and error in steady state distribution ϵ
Output: Endemic regions returned as a partitioning P of graph $G(V, E)$
BEGIN
Randomly assign color c_p from $C = \{c_1, \ldots, c_{|C|}\}$ to each node $v_i \in V$
I $\leftarrow \frac{W}{w_{\min}\beta}\frac{1}{|V|}\log\frac{|V|}{\epsilon}$
while number of iterations $<$ I **do**
  for each $v_i \in V$, sampled according to J **do**
    sample color for v_i according to the distribution where probability of picking color c_p is $p_{v_i}(c_p|G^C_{-v_i})$ (from Eqn 2.3.2)
  end for
end while
Initialize $T_{iter} \leftarrow 1$
while $T_{iter} > T_f$ **do**
  for each $v_i \in V$ **do**
    sample color for v_i according to the distribution where probability of picking color c_p is directly proportional to $(p_{v_i}(c_p|G^C_{-v_i}))^{1/T_{iter}}$
  end for
  $T_{iter} \leftarrow \alpha T_{iter}$
end while
return $P = \text{GetPartitions}(G^C)$

generally dictates the cascade simulation process.

Parameter ϵ

Parameter ϵ measures error between estimated and true steady state distributions, in terms of statistical variation. It offers a run-time vs. accuracy trade-off and results are robust even over orders of magnitude of change. Empirical results showed that $\epsilon = 10^{-20}$ serves well for many applications.

Simulated Annealing parameters, α and T_f

Parameters α and T_f control the cooling schedule of the annealing process (Andrieu et al. 2003). The process starts with initial temperature $T = 1$ and proceeds, with cooling rate α, till $T < T_f$. Larger α and T_f , result in more gradual cooling and more time

Algorithm 2 GetPartitions(G^C)

Input: Colored graph G^C
Output: Partitioning P on G
BEGIN
Initialize $q \leftarrow V$
while $q \neq \phi$ **do**
  $v \leftarrow \text{pop}(q)$
  Create new partition $V_j = \{v\}$
  Initialize $q_j \leftarrow \{v\}$
  while $q_j \neq \phi$ **do**
    $v_s \leftarrow \text{pop}(q_j)$
    for each $v_n | v_n \in N(v_s)$ and $v_n \in q$ **do**
      if color(v_s) = color(v_n) **then**
        remove v_n from q and add to V_j
        push(q_j,v_n)
      end if
    end for
  end while
end while
return $P = \{V_1, V_2, .., V_k\}$

spent searching for the ML state. Empirical results showed that $\alpha = 0.99$ and $T_f = 0.1$ work well for many appli- cations.

Parameter β

As parameter β is increased from 0 to 1, mixing properties of the chain improve, and results with more fine-grained partitions are favored. On the other hand, lower values of β result in more coarse partitions where some of the finer partitions are merged. Empirically, increasing β improved results, however, good quality results were obtained throughout the range of β. For many applications β within range $[0.2, 0.9]$ worked well.

Number of Colors $|C|$

While the number colors significantly impacts the search space of *StochColor*, surprisingly, it was empirically observed that the results are highly robust to, sometimes even over orders of magnitude of, change in $|C|$ and $|C| = 100$ provided good results for many

applications.

2.4 Experiments

2.4.1 Synthetic Networks

Methodology

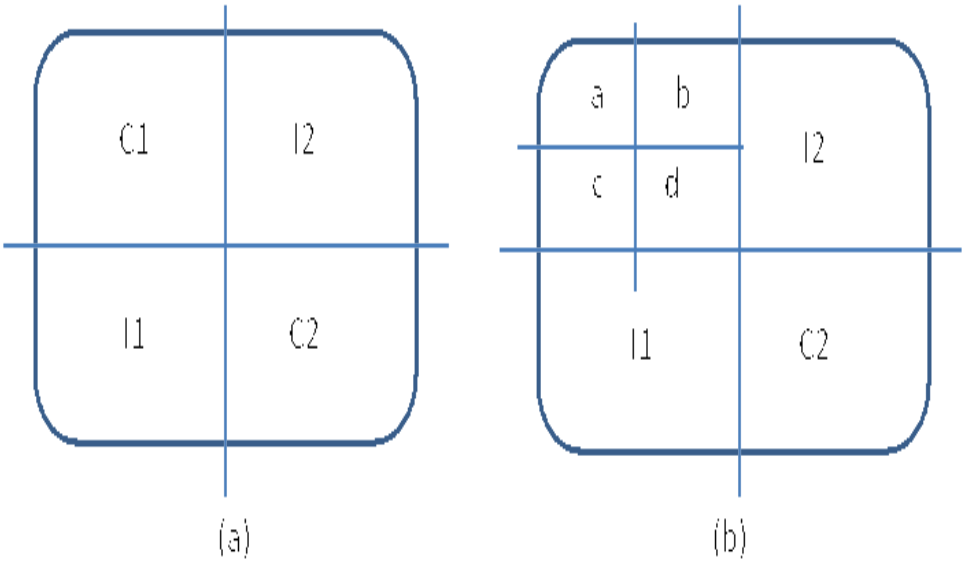


Figure 2.2: Adjacency matrices partitioned to create synthetic graphs

The behavior of StochColor was examined on a variety of synthetic networks with varying levels of communmity structure. These synthetic networks were generated using a modified version of the RMAT algorithm [27]. A graph G is generated by starting with an empty adjacency matrix and iteratively adding edges to it. For each edge, a stochastic process is used to sample the position where the edge is added.

The empty adjacency matrix is divided into four regions (figure 2.2) - $C1, C2, I1$ and $I2$. The process is set up such that on a coarse level G has two communities $C1$ and $C2$ with $I1$, $I2$ containing inter-community edges. In order to add an edge e, first sample one of the four regions $C1$, $C2$, $I1$ and $I2$ with probabilities p, p, q and q respectively. Note that $2p + 2q = 1$ and $p \geq q$. Then further divide the extracted region into four

parts a, b, c and d. If $C1$ or $C2$ is sampled in the first step, divide the corresponding region into four equal parts and sample one of these four regions a, d, c and b having probabilities p_C, p_C, q_C and q_C respectively with $2p_C + 2q_C = 1$ and $p_C \geq q_C$. Once again divide the sampled region into four equal parts and keep repeating the previous step till only a single cell is left. Add the edge at this position. Instead of $C1$ and $C2$ if $I1$ or $I2$ are sampled, repeat the same procedure as $C1$ and $C2$ except instead of using p_C and q_C, use parameters p_I and q_I. Once again, $2p_I + 2q_I = 1$ and $p_I \geq q_I$.

Since, it is required for synthetic graph G to be symmetric, once the desired number of edges have been added, the lower triangle of the final adjacency matrix is copied to the upper triangle (this makes $I1 = I2 = I$). Due to the nature of the parameters, $I1$ should not be too different from $I1$ and the number of edges in the final symmetric matrix is quite close to the original desired value. Finally, in order to make the graph weighted, edge-weights were taken to be the number of times corresponding cells in the adjacency matrix were sampled.

Parameters p and q control the intra-community connectivity w.r.t. the inter-community cut between $C1$ and $C2$. The greater is p, the more edges are added to $C1$ as well as $C2$ and lesser number of edges are added to I. With p_C and q_C, the procedure controls the sub-community structure within $C1$ and $C2$. The higher p_C is, the more likely are $C1$ and $C2$ to have a bunch of smaller sub-communities witin it. For smaller values of p_C, they will have near uniform edge density within. Parameters p_I and q_I behave similar to p_C and q_C but since I can have fewer edges, larger values of p_I cause for inter-community edges to be bunched around the diagonal of I and in case of lower values, they are more spread out.

For the experiments, the ratios $p/q, p_C/q_C$ and p_I/q_I were each set to values - $[1, 3, 10, 25, 50, 100]$ and *StochColor* was run on graphs generated using all possible combinations ($6 \times 6 \times 6 = 216$) of the parameters. For all simulations *StochColor* was run using 100 Colors and $\beta = 0.2$, and the averaged result from 20 runs was used.

Observations

In almost all cases one of the following three outcomes was observed

- Outcome 1: Of the 100 cascades, 98 of them died out and the 2 cascades left were endemic in communities $C1$ and $C2$ each.

p'/q'

p(c)/q(l)						
	{0,1,0}	{0,1,0}	{0,1,0}	{0,0.7,0.3}	{0,0,1}	{0,0,1}
	{0,1,0}	{0,1,0}	{0,1,0}	{0,1,0}	{0,0.9,0.1}	{0,0.05,0.95}
	{0,1,0}	{0,1,0}	{0,1,0}	{0,1,0}	{0,1,0}	{0.05,0.95,0}
	{0.1,0.9,0}	{0.3,0.7,0}	{0.05,0.95,0}	{0,1,0}	{0,1,0}	{0,1,0}
	{0.7,0.3,0}	{0.45,0.55,0}	{0.45,0.55,0}	{0.1,0.9,0}	{0,1,0}	{0,1,0}
	{0.85,0.15,0}	{0.7,0.3,0}	{0.9,0.1,0}	{0.4,0.6,0}	{0.1,0.9,0}	{0,1,0}

(a) Matrix 1

p'/q'

p(c)/q(l)						
	{0,1,0}	{0,1,0}	{0,1,0}	{0,0.7,0.3}	{0,0,1}	{0,0,1}
	{0,1,0}	{0,1,0}	{0,1,0}	{0,1,0}	{0,0.9,0.1}	{0,0.05,0.95}
	{0.5,0.5,0}	{0,1,0}	{0,1,0}	{0,1,0}	{0,1,0}	{0.05,0.95,0}
	{0.1,0.9,0}	{0.1,0.9,0}	{0.05,0.95,0}	{0,1,0}	{0,1,0}	{0,1,0}
	{0.45,0.55,0}	{0.75,0.25,0}	{0.2,0.8,0}	{0,1,0}	{0,1,0}	{0,1,0}
	{0.85,0.15,0}	{0.7,0.3,0}	{0.7,0.3,0}	{0.6,0.4,0}	{0.15,0.85,0}	{0,1,0}

(b) Matrix 2

Figure 2.3: Probabilities of observing the variuous outcomes

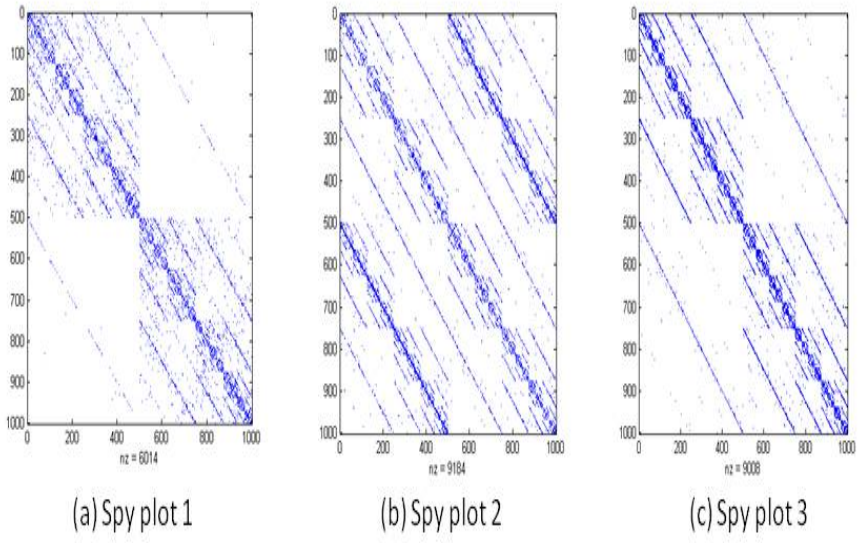


Figure 2.4: Spy-plots for different synthetic graphs

- Outcome 2: Of the 100 cascades, 99 of them died out and the 1 remaining cascade was pandemic in G.
- Outcome 3: Different cascades were endemic in different small tight knit communities.

Matrices in figure 2.3 presents the probabilities of observing each outcome. In this figure, matrices 1 and 2 present results on synthetic graphs and are representative of lower [1,3,10] and higher [25,50,100] values of p_I/q_I respectively. p_C/q_C increases from left to right [1,3,10,25,50,100] and p/q increases from top to bottom [1,3,10,25,50,100]. Each cell has a 3-ary tuple whose first, second and third elements show the odds of outcomes 1, 2 and 3, respectively, for graphs generated using the corresponding combination of parameters (computed using 20 runs for each combination) In the matrices presented, top to bottom is increasing p/q and left to right is increasing p_C/q_C. Matrices 1 and 2 are representative of results for lower values (1,3,10) and higher values (25,50,100) of p_I/q_I respectively. The following trends are common to both the matrices -

- As the inter-community cut decreases, the cascades are more likely to be endemic in communities $C1$ and $C2$ (Outcome 1).
- As the sub-community structure in $C1$ and $C2$ is increased initially the intra-communty connectivty becomes more dense in certain local regions (Fig. 2.4) and outcome 1 is more likely. This is because these local dense regions offer more resistance to the cascade attempting to cross over from community $C1$ to $C2$ and vice-versa.
- If the sub-community structure in $C1$ and $C2$ is further increased, for larger values of p/q, $C1$ and $C2$ become sparse with a bunch of small communities (Fig. 2.4). While both $C1$ and $C2$ are sparse, the small communities are sufficiently well connected for cascades to transition from one community to the other and so outcome 2 becomes more likely.

The difference between the two matrices is outcome 3, which becomes prominent in the upper-right region of matrix 2 (low p/q, high p_C/q_C and high p_I/q_I). At these

parameter values the 2 community structure ($C1$ and $C2$) disappears and instead there are a bunch smaller tight-knit communities with low weight edges connecting them to each other (Fig. 2.4). As a result, there is significant resistance met by the various cascades attempting to transition across these communities.

The key takeaway from the experiment is the behavior of the cascades w.r.t the graph topology. For a cascade to not transition between two partitions in a graph, there must be no local regions of sufficient connectivity between them. Conversely, if a such a region of sufficient inter-partition connectivity (relative to the intra-partition connectivities) then cascades will be able to cross over from one partition to the other.

Increasing number of colors and β for *StochColor's* input parameters does not produce any change in the results. Lowering them both forces the algorithm to start favoring solutions having fewer endemic regions as well as outcome 2. However, the changes are not significant, unless it is outcome 3 where coarser endemic regions are returned. Overall, the primary factor dictating cascade behavior appears to be the graph topology.

2.4.2 Real Networks

Methodology

In order to demonstrate the quality of the solution state, obtained using *StochColor*, the endemic regions in it are treated as a partitioning of the input graph and compared to results from state of the art graph partitioning algorithms on a variety of real world datasets from different domains - 32-bit adder (*add32*), structural engineering (*bcsstk29*), finance (*finance256*), human brain network (*brain*), yeast network (*NDyeast*) and General Relativity, Physics, co-authorship network (*ca-GrQc*) (Table 2.1).

For each dataset, results from the following algorithms were computed -

- *StochColor* - In all experiments values $|C| = 100$, $\beta = 0.9$, $\epsilon = 10^{-20}$, $T_f = 0.1$ and $\alpha = 0.99$ were used and the result corresponding to the median of number of partitions returned over 5 runs was taken.

- *Graclus* [31] - Graclus with base spectral clustering algorithm and 20 steps of localized search. Graclus takes the number of partitions as input which was taken to be the number of partitions returned by *StochColor*.

Graph	**# of Nodes**	**# of Edges**	**Source**
add32	4960	7444	[28]
bcsstk29	13992	302748	[28]
finance256	37376	130560	[28]
brain	998	37926	[29]
NDyeast	1846	2203	[28]
ca-GrQc	5242	14484	[30]

Table 2.1: Networks from Various Domains

- *Metis* [32] - Metis requires number of partitions as input which was taken to be the number of partitions returned by *StochColor*.

Normalized Cut (NCut) was used to measure quality of partitioning. Lower NCut means better partitioning. For a given set of partitions $P = \{V_i\}_{i=1...k}$ on a graph $G(V, E)$, $NCut(P, G) = \sum_{i=1}^{k} \frac{links(V_{p_i}, V/V_{p_i})}{deg(V_{p_i})}$

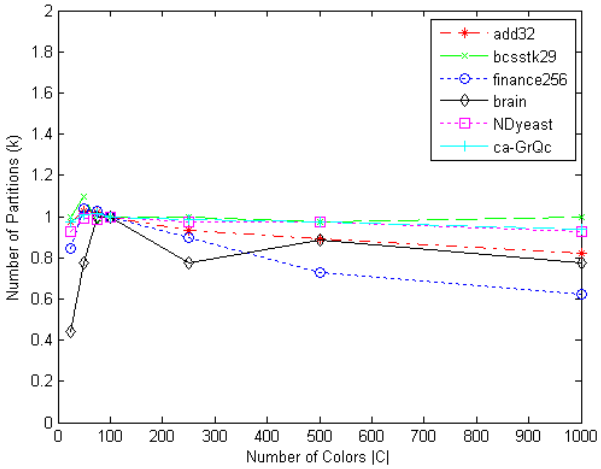
Observations

Graph	*StochColor*		*GraclusSC*	*MetisSC*
	k	NCut		
add32	211	18.68	**16.53**	22.8
bcsstk29	42	**1.39**	6.55	6.94
finance256	248	**29.14**	38.34	54.8
brain	9	**1**.2	**1.10**	1.62
NDyeast	213	**10.87**	58.16	-
ca-GrQc	467	**13.28**	158.25	186.17

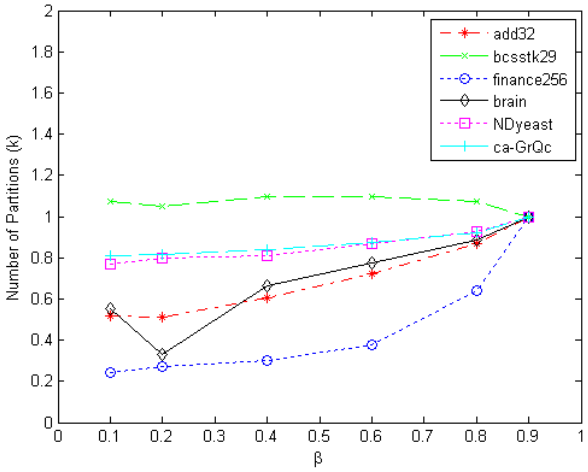
Table 2.2: NCut for the different algorithms across many applications.

Table 2.2 compares *StochColor* with *Graclus* and *Metis* on datasets in Table 2.1. *StochColor* is doing a little worse on add32, comparable on brain and significantly better for all other graphs. *Metis* had memory issues with *NDyeast* and so results for the same are not reported.

Figure 2.5 shows the number of partitions returned by *StochColor* when varying $|C|$ and β parameters respectively. .In sub-figure (a) and (b), $\beta = 0.9$ and $|C| = 100$ respectively. For each dataset the number of partitions is scaled w.r.t. the number of

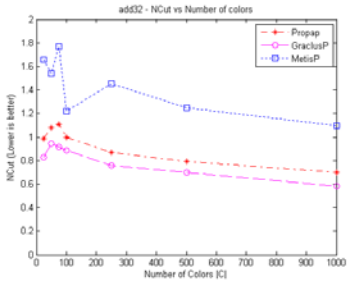


(a) Number of partitions (k) vs $|C|$ (25-1000)

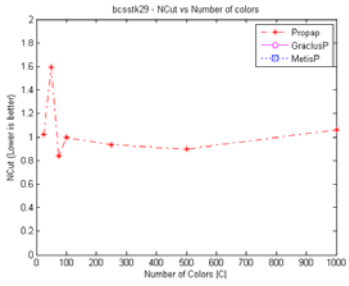


(b) Number of partitions (k) vs β (0.1-0.9)

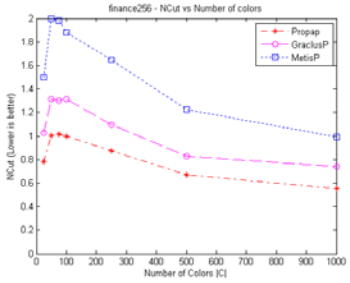
Figure 2.5: Number of partitions vs changing parameters



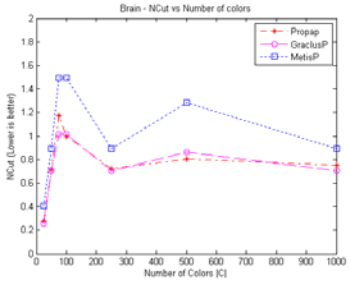
(a) *add32* - NCut vs # of Colors



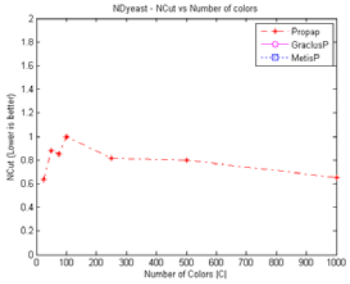
(b) *bsstk29* - NCut vs # of Colors



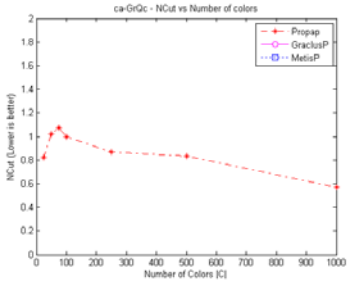
(c) *finance256* - NCut vs # of Colors



(d) *brain* - NCut vs # of Colors

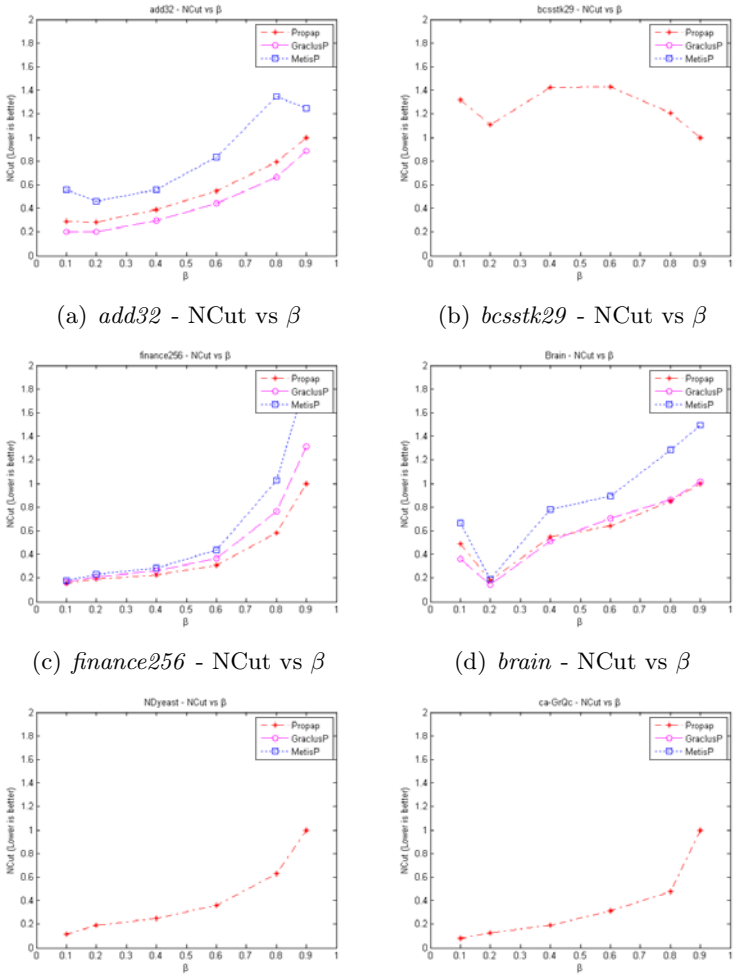


(e) *NDyeast* - NCut vs # of Colors



(f) *ca-GrQc* - NCut vs # of Colors

Figure 2.6: NCut vs changing # of colors (25-1000)



(a) *add32* - NCut vs β (b) *bcsstk29* - NCut vs β

(c) *finance256* - NCut vs β (d) *brain* - NCut vs β

(e) *NDyeast* - NCut vs β (f) *ca-GrQc* - NCut vs β

Figure 2.7: NCut vs changing β (0.1-0.9)

partitions in the result from *StochColor* for $|C| = 100$ and $\beta = 0.9$.

- The number of partitions returned are generally stable even over large changes in $|C|$.
- For β the stability is relatively less and a general trend of larger β returning more partitions can be observed.

Figure 2.6 and Figure 2.7 present the corresponding NCut for direrent partitionings returned by *StochColor* when varying $|C|$ and β respectively. In each plot the NCut values are scaled w.r.t. the NCut for partitioning from *StochColor* for $|C| = 100$ and $\beta = 0.9$. The y-axis scale is from 0-2 and in some cases where NCut values from *GraclusSC* and *MetisSC* are more than twice of that from *StochColor* (# of colors = 100, $\beta = 0.9$), the curve corresponding to Graclus and Metis does not appear in the plot.

- Relative to *GraclusSC*, the quality of partitions from *StochColor* across different parameter values are consistently worse for *add32*, comparable for *brain* and significantly better for all other graphs.
- Increasing the number of colors seems to have a gradual decrease in NCut for all graphs except *bcsstk29*.
- Increasing β generally improves the quality of partitioning in all cases except *bcsstk29* which stays between 1 and 1.4.

Image segmentation

Algorithm	**NCut**	**Precision**	**Recall**	**F-measure**
StochColor	5.37	0.31	0.90	**0.46**
Graclus	**2.91**	0.298	0.92	0.45

Table 2.3: Image segmentation results

In order to provide a visual representation for *StochColor's* solution, results on image data are also presented for *StochColor* and *Graclus* (with number of partitions from *StochColor* as input) Figure 2.4.2(a) is taken from the Berkeley Image Segmentation

(a) Original (b) Segmented *StochColor* (c) Segmented *Graclus*

Figure 2.8: Image segmentation using (b)*StochColor* and (c)*Graclus*

Dataset [33] and a corresponding affinity matrix was computed using J. Shi's code [1] [34]. The resulting graph had 154,401 nodes and 31,210,628 edges. Due to the size, *StochColor* was run for $5000|V|$ steps. Partitions returned were used to segment the image (Figures 2.4.2(b) and (c)). Along with NCut, a a ground truth segmentation, provided by the authors of the dataset, was also used to compute precision/recall (Table 2.3).

- *StochColor* loses out on NCut but does better on precision and f-measure w.r.t. ground truth.
- *StochColor* extracted 64 partitions and the ground truth (extracted manually using human subjects) had an average of 46 partitions.

Runtime for StochColor is significantly more than the same for the other graph partitioning algorithms. However, the latter are multi-level methods specifically designed for efficient graph partitioning. Empirically, it was observed that unless the graph is quite dense (*brain* and image datasets) for most sparse graphs, runtime is about $n(|E| + |C||V|)$ where n is of the order of a few hundreds.

[1] `http://www.cis.upenn.edu/~jshi/software/`

2.5 Conclusions

In this work a generalized version of the liner threshold model, capable of handling multiple competing cascades as well as allowing nodes to switch back and forth between cascades (non-progressive case), is proposed. A stochastic graph coloring process is used to solve the model. This process is shown to be a rapidly mixing Markov chain and the *StochColor* algorithm is presented to compute highly likely states, from the steady state distribution, for the cascading process. Results indicate that graph topology plays a strong role in dictating the behavior of the cascades. Results on real world networks demonstrate the high quality of the solution states computed using *StochColor*. Estimating cascades' spreads can be very helpful in viral marketing scenarios particularly when a new competitor is attempting to enter a market and the need to spreading product awareness is high. Understanding which portions of the network are likely to be homophilic and how many cascades can the topology accomodate are all useful knowledge for crafting marketing campaigns. the extent of cascade spreads for a given network

Chapter 3

Social Topic Communities in Social Networks

3.1 Introduction

Communities play a vital role in understanding the creation, representation, and transfer of knowledge among people, and are an essential building block of all social networks. However, the relationship of one individual in a community to one another is not easily formalized, or necessarily consistent. These individuals, in a social network, connect with each other due some shared interest, kinship, goals etc. The context around these social connections are often reflected in their communcation.

With social interaction playing an increasingly important role in the online world, the capability to extract latent communities based on such interactions is becoming vital for a wide variety of applications. It is a vital knowledge discovery tool for discovering the different communities as well as the context around the social connections existing within these communities. Such information would be valuable in monitoring community sentiments/interests and would be of immense use to a variety of entities including marketing, community managers or leaders and web based services providing portals for these communities.

Traditional methods of community extraction have been primarily link-based. Link-based methods produce communities formed from the explicit links between individuals expressed via some form of measurable interpersonal communication, such as an email

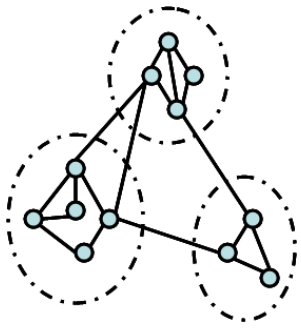


Figure 3.1: Communities in a social network

or instant message. Actors and their communications are then represented as a graph which is partitioned into different communities ([35] and [36]). The essential assumption is that intra-community communication is far more dense than inter-community communication. However, community extraction based only on communication links can result in communities which are topically dissimilar, and overly sensitive to individuals who have widely varying philosophies about the frequency of communication and/or the scope of their audience. Thus, it is possible to have two or more latent communities discussing disparate topics merged into a single community since topical information is not utilized. Further, the assumption that every individual belongs to one and only one community does not necessarily hold true in typical social settings. There can also be individuals who are socially inactive and do not belong to any community.

Topic-based methods, on the other hand, can generate communities which are topically similar. In a purely topic-based method, groups of individuals who communicate about the same (or similar) topics become communities in such a framework. A drawback to this approach is that while the communities are topically similar, the individuals contained therein may not share any explicit communication and, as such, may not actually reflect a "community" in the traditional sense. Additionally, issues of synonymy can plague topic-based methods because localized vernacular is not taken into account during extraction, and so while communities are formed which share the same words, the context those words exist in is neglected. This problem is further compounded in social networks utilizing a homogenous language among individuals, such as a company or academic department.

This chapter presents a probabilistic model for community extraction which allows

actors to participate in multiple communities by leveraging both topic and link information from the social network. In particular, we propose a Bayesian model that follows an intuitive generative scheme for modeling email communication. Unlike much of existing literature, the proposed model extracts communities based on both communication link as well as content information. The underlying assumption behind the model is that actors in a community communicate on topics of mutual interest, and the topics of communication, in turn, determine the communities. Further, the model is probabilistic, and allows actors to be a part of multiple communities. Through extensive experiments and visualization on the Enron email corpus, we demonstrate that the model is able to extract well connected and topically meaningful communities. Additionally, the model extracts relevant topics that can be mapped back to corresponding real life events involving Enron.

3.2 Proposed Approach

This section presents the CART (Community-Author-Recipient-Topic) model for community extraction. CART is a Bayesian generative model which extends the popular ART (Author-Recipient-Topic) model to discover latent community structure based on authors and recipients. In particular, the observed authors and recipients of an email are assumed to be generated from a latent community. Figures 3.2(a) and (b)

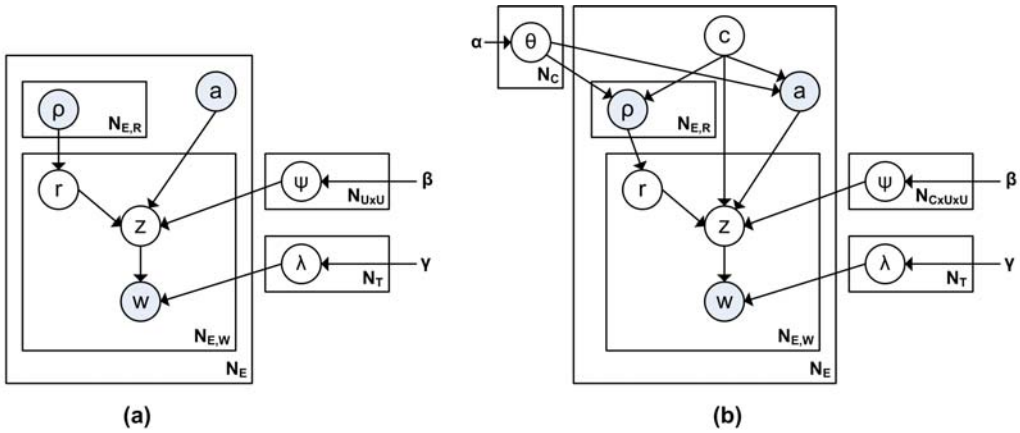


Figure 3.2: Email network modeling using (a) ART (b) CART

illustrate the ART and CART models respectively. The CART model has the following

generation scheme:

1. To generate email e_d, a community c_d is chosen uniformly at random.
2. Based the community c_d, the author a_d and the set of recipients ρ_d are chosen.
3. To generate every word $w_{(d,i)}$ in that email, a recipient $r_{(d,i)}$ is chosen uniformly at random from the set of recipients ρ_d.
4. Based on the community c_d, author a_d, and recipient $r_{(d,i)}$, a topic $z_{(d,i)}$ is chosen.
5. The word $w_{(d,i)}$ itself is chosen based on the topic $z_{(d,i)}$.

Other than the uniform distributions for sampling communities c_d, and recipients r_d from ρ_d, all other discrete distributions used in the generative model have Dirichlet priors as shown in Figure 3.2(b). The author a_d, set of recipients ρ_d, and sequence of words $\mathbf{w}_d$ used in every email e_d are observable from the email log data, and all other variables are latent. The total number of words (W) and users (U) can be determined from the email log data and the number of communities (C) and topics (T) are provided as inputs. From the model, we can see that every email is constrained to belong to one community. This constrains all users involved and the topics of conversation to belong to the same community in the context of that particular email. A subset of the same users and topics may get assigned to a different community in the context of a different email. The basic intuition behind such a model is that users within a community communicate with each other on topics relevant to themselves as well as the community. Thus, we incorporate link as well as content based information in our community extraction model. The joint probability distribution for the various entities (i.e., communities, authors, recipients, topics and words) for a given email e_d is given as

$$\begin{aligned} & p(c_d, a_d, \rho_d, \mathbf{r}_d, \mathbf{z}_d, \mathbf{w}_d) \\ & = p(c_d)p(a_d|c_d) \prod_{r \in \rho_d} p(r|c_d) \prod_{i=1}^{N_d} p(w_{(d,i)}|z_{(d,i)})p(z_{(d,i)}|c_d, a_d, r_{(d,i)}) \ , \end{aligned} \tag{3.2.1}$$

where $\mathbf{r}_d$ is the sequence of latent recipients (selected from ρ_d), $\mathbf{z}_d$ is the sequence of latent topic corresponding to word sequence $\mathbf{w}_d$ in the email, $r_{(d,i)}$ is the latent recipient

and $z_{(d,i)}$ is the latent topic corresponding to the i^{th} word $w_{(d,i)}$, and N_d is the total number of words in the email.

Given an email corpus over a network of users, the CART model enables the discovery of latent communities in the network, as well as the latent social topics of discussion in the corpus. From a Bayesian network perspective, given the set of observable nodes $(\mathbf{a}, \rho, \mathbf{w})$, such latent structure discovery can be carried out by doing inference over the latent nodes $(\mathbf{c}, \mathbf{r}, \mathbf{z})$. Motivated by recent work on sampling based inference for hierarchical Bayesian models [37], inference in the CART model is carried out using Gibbs Sampling. For CART, the Gibbs sampling updates alternate between updating latent communities c_d conditioned on other variables, and updating recipient-topic tuples $(r_{(d,i)}, z_{(d,i)})$ for each word conditioned on other variables. In particular, the conditional distribution of the community assignment of an email e_d is given by

$$\begin{aligned} & p(c_d = c|\mathbf{c}_{-d}, \boldsymbol{\rho}, \mathbf{r}, \mathbf{a}, \mathbf{z}, \mathbf{w}) \\ & \propto \frac{\prod_{u_i \in \{\rho_d, a_d\}} (n^{CU}_{-d,cu_i} + \alpha)}{\prod_{i=0}^{|\rho_d|} \sum_{u=1}^{U} (n^{CU}_{-d,cu} + U\alpha + i)} \\ & \times \prod_{r \in \rho_d} \left(\frac{\prod_{z=1}^{T} \Gamma(e_{d,rz} + n^{(CUU)T}_{-d,(c_d a_d r)z} + \beta)}{\Gamma\left(\sum_{z=1}^{T} (e_{d,rz} + n^{(CUU)T}_{-d,(c_d a_d r)z})\right) + T\beta} \right) , \end{aligned} \tag{3.2.2}$$

where n^{CU}_{-d,cu_i} is the number of times user u_i was generated from community c other than email d, $e_{d,rz}$ is the number of times topic z was generated from recipient r in email d, and $n^{(CUU)T}_{-d,(c_d a_d, r)z}$ is the number of times topic z was generated from community, author, recipient (c_d, a_d, r) other than email d. Further, the conditional distribution of the recipient-topic tuple assignment for a word $w_{(d,i)}$ is given by

$$\begin{aligned} & p(r_{(d,i)} = r, z_{(d,i)} = z|\mathbf{c}_{-d}, \mathbf{a}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-(d,i)}, \mathbf{z}_{-(d,i)}, \\ & \qquad \mathbf{w}_{-(d,i)}, c_d, a_d, \rho_d, w_{(d,i)} = w) \\ & \propto \frac{n^{TW}_{-(d,i),zw} + \gamma}{\sum_{v=1}^{W} n^{TW}_{-(d,i),zv} + W\gamma} \times \frac{n^{(CUU)T}_{-(d,i),(c_d a_d r)z} + \beta}{\sum_{h=1}^{T} n^{(CUU)T}_{-(d,i),(c_d a_d r)h} + T\beta} \end{aligned} \tag{3.2.3}$$

where, $n^{XY}_{-(d,i),xy}$ is the number of times $y \in Y$ was generated by $x \in X$ excluding the i^{th} instance in email d. Detailed derivations of the conditional distribution based

updates are presented in the next section. It is important to note that the presence of a latent node (community c) higher up in the Bayesian network makes the CART model markedly different from much of the recent literature on non-parametric hierarchical Bayesian models. In particular, the Gibbs sampling update for the node c is not as straightforward as latent nodes (such as (r, z)) corresponding to the lower nodes in the Bayesian network.

Using the above updates, a Gibbs sampling simulation is carried till convergence, and the latent node assignments for every email are determined. For a given assignment of latent node values, the communities can be determined as:

$$p(u|c) = \frac{n_{cu}^{CU} + \alpha}{\sum_i n_{cu_i}^{CU} + U\alpha} \tag{3.2.4}$$

where, n_{cu}^{CU} is the number of times user u was generated from community c. The above equation associates a degree of membership for every user belonging to a community. Note that the model allows for mixed membership, i.e., a user is allowed to participate in more than one community. By counting how many times a user is assigned to a particular community, we can determine the top users for every community. Similarly we can also determine the topmost words for every topic. These topmost words and users can be used to analyze (or to put it more correctly *visualize*) the different topics and communities respectively.

3.3 Derivation of CART Updates

The joint probability distribution for the various entities (i.e. communities, authors, recipients, topics and words) for a given email e_d is given as

$$\begin{aligned} &p(c_d, a_d, \rho_d, \mathbf{r}_d, \mathbf{z}_d, \mathbf{w}_d) \\ &= p(c_d)p(a_d|c_d) \prod_{r \in \rho_d} p(r|c_d) \prod_{i=1}^{N_d} p(w_{d,i}|z_{d,i})p(z_{d,i}|c_d, a_d, r_{d,i}) \ , \end{aligned} \tag{3.3.1}$$

where ρ_d is the set of observed unique recipients in the email, $\mathbf{r}_d$ is the sequence of latent recipients (selected from ρ_d) and $\mathbf{z}_d$ is the sequence of latent topic corresponding to each word in the email, and N_d is the number of words in the email.

Lemma 1. *For a given email e_d,*

$$\begin{aligned} & p(c_d = c|\mathbf{c}_{-d}, \boldsymbol{\rho}, \mathbf{r}, \mathbf{a}, \mathbf{z}, \mathbf{w}) \\ & \propto \frac{\prod_{u_i \in \{\rho_d, a_d\}} (n^{CU}_{-d,cu_i} + \alpha)}{\prod_{i=0}^{|\rho_d|} \sum_{u=1}^{U} (n^{CU}_{-d,cu} + U\alpha + i)} \\ & \quad \times \prod_{r \in \rho_d} \left(\frac{\prod_{z=1}^{T} \Gamma(e_{d,rz} + n^{(CUU)T}_{-d,(c_d a_d r)z} + \beta)}{\Gamma\left(\sum_{z=1}^{T} (e_{d,rz} + n^{(CUU)T}_{-d,(c_d a_d r)z})\right) + T\beta} \right) , \end{aligned} \tag{3.3.2}$$

where n^{CU}_{-d,cu_i} is the number of times user u_i was generated from community c other than email d, $e_{d,rz}$ is the number of times topic z was generated from recipient r in email d, and $n^{(CUU)T}_{-d,(c_d a_d, r)z}$ is the number of times topic z was generated from community, author, recipient (c_d, a_d, r) other than email d.

Proof. Using Bayes rule,

$$\begin{aligned} p(c_d = c|\mathbf{c}_{-d}, \boldsymbol{\rho}, \mathbf{r}, \mathbf{a}, \mathbf{z}, \mathbf{w}) &= p(c_d = c|\mathbf{c}_{-d}, \boldsymbol{\rho}, \mathbf{r}, \mathbf{a}, \mathbf{z}) \\ &\propto p(a_d, \rho_d, \mathbf{r}_d, \mathbf{z}_d|c_d = c, \mathbf{c}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-d}, \mathbf{a}_{-d}, \mathbf{z}_{-d}) \\ &= p(a_d, \rho_d|c_d = c, \mathbf{c}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-d}, \mathbf{a}_{-d}, \mathbf{z}_{-d}) \times p(\mathbf{r}_d|\rho_d) \\ &\quad \times p(\mathbf{z}_d|c_d = c, \mathbf{c}_{-d}, \rho_d, \boldsymbol{\rho}_{-d}, a_d, \mathbf{a}_{-d}, \mathbf{r}_d, \mathbf{r}_{-d}, \mathbf{z}_{-d}) \\ &\propto p(a_d, \rho_d|c_d = c, \mathbf{c}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-d}, \mathbf{a}_{-d}) \\ &\quad \times p(\mathbf{z}_d|c_d = c, \mathbf{c}_{-d}, a_d, \mathbf{a}_{-d}, \mathbf{r}_d, \mathbf{r}_{-d}, \mathbf{z}_{-d}) \\ &= T_1 \times T_2 \ . \end{aligned}$$

Now,

$$
\begin{aligned}
T_1 &= p(a_d, \rho_d | c_d = c, \mathbf{c}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-d}, \mathbf{a}_{-d}) \\
&= \prod_{\substack{i=0 \\ u_i \in \{a_d, \rho_d\}}}^{|\rho_d|} p(u_i | c_d = c, \mathbf{c}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-d}, \mathbf{a}_{-d}, u_0, \ldots, u_{i-1}) \\
&= \prod_{i=0}^{|\rho_d|} \int_{\phi_c} \Bigg(p(u_i | c_d = c, \phi_c) \\
&\qquad \times p(\phi_c | c_d = c, \mathbf{c}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-d}, \mathbf{a}_{-d}, u_0, \ldots, u_{i-1}) \Bigg) \, d\phi_c \\
&= \frac{n^{CU}_{-d,cu_0} + \alpha}{\sum_u n^{CU}_{-d,cu} + U\alpha} \times \frac{n^{CU}_{-d,cu_1} + \alpha}{\sum_u n^{CU}_{-d,cu} + U\alpha + 1} \times \cdots \\
&\qquad \times \frac{n^{CU}_{-d,cu_{|\rho_d|}} + \alpha}{\sum_u n^{CU}_{-d,cu} + U\alpha + |\rho_d|} \\
&= \frac{\prod_{u_i \in \{\rho_d, a_d\}} (n^{CU}_{-d,cu_i} + \alpha)}{\prod_{i=0}^{|\rho_d|} \sum_{u=1}^{U} (n^{CU}_{-d,cu} + U\alpha + i)} \; .
\end{aligned}
$$

Further,

$$
\begin{aligned}
T2 &= p(\mathbf{z}_d | c_d = c, \mathbf{c}_{-d}, a_d, \mathbf{a}_{-d}, \mathbf{r}_d, \mathbf{r}_{-d}, \mathbf{z}_{-d}) \\
&= \prod_{r \in r_d} \int_{\psi_{ca_dr}} \Bigg(p(\mathbf{z}_{d,r} | c_d = c, a_d, r, \psi_{ca_dr}) \\
&\qquad \times p(\psi_{ca_dr} | c_d = c, \mathbf{c}_{-d}, a_d, \mathbf{a}_{-d}, \mathbf{r}_d, \mathbf{r}_{-d}, \mathbf{z}_{-d}) \Bigg) \, d\psi_{ca_dr} \\
&= \prod_{r \in r_d} \int_{\psi_{ca_dr}} \left(\prod_{z=1}^{T} \psi_{(ca_dr)z}^{e_{d,zr} + n^{(CUU)T}_{-d,z(c_da_dr)} + \beta} \right) \, d\psi_{ca_dr} \\
&= \prod_{r \in r_d} \left(\frac{\prod_{z=1}^{T} \Gamma(e_{d,zr} + n^{(CUU)T}_{-d,(c_da_dr)z} + \beta)}{\Gamma\left(\sum_{z=1}^{T} (e_{d,zr} + n^{(CUU)T}_{-d,(c_da_dr)z}) \right) + T\beta} \right) \; .
\end{aligned}
$$

That completes the proof. □

Lemma 2. *For a given email e_d,*

$$\begin{aligned} p(r_{(d,i)} = r, z_{(d,i)} = z | \mathbf{c}_{-d}, \mathbf{a}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-(d,i)}, \mathbf{z}_{-(d,i)}, \\ \mathbf{w}_{-(d,i)}, c_d, a_d, \rho_d, w_{(d,i)} = w) \\ \propto \frac{n^{TW}_{-(d,i),zw} + \gamma}{\sum_{v=1}^{W} n^{TW}_{-(d,i),zv} + W\gamma} \times \frac{n^{(CUU)T}_{-(d,i),(c_d a_d r)z} + \beta}{\sum_{h=1}^{T} n^{(CUU)T}_{-(d,i),(c_d a_d r)h} + T\beta} \end{aligned} \tag{3.3.3}$$

where, $n^{XY}_{-(d,i),xy}$ is the number of times $y \in Y$ was generated by $y \in Y$ excluding the i^{th} instance in email d.

Proof. Using Bayes rule,

$$\begin{aligned} & p(r_{(d,i)} = r, z_{(d,i)} = z | \mathbf{c}_{-d}, \mathbf{a}_{-d}, \boldsymbol{\rho}_{-d}, \mathbf{r}_{-(d,i)}, \mathbf{z}_{-(d,i)}, \\ & \qquad \mathbf{w}_{-(d,i)}, c_d, a_d, \rho_d, w_{(d,i)} = w) \\ & = p(r_{(d,i)} = r | \rho_d) \times p(z_{(d,i)} = z | \mathbf{c}_{-d}, \mathbf{a}_{-d}, \mathbf{r}_{-(d,i)}, \mathbf{z}_{-(d,i)}, \\ & \qquad \mathbf{w}_{-(d,i)}, c_d, a_d, \rho_d, r_{(d,i)} = r, w_{(d,i)} = w) \\ & \propto p(z_{(d,i)} = z | \mathbf{c}_{-d}, \mathbf{a}_{-d}, \mathbf{r}_{-(d,i)}, \mathbf{z}_{-(d,i)}, c_d, a_d, \rho_d, r_{(d,i)} = r) \\ & \times p(w_{(d,i)} = w | \mathbf{z}_{-(d,i)}, \mathbf{w}_{-(d,i)}, z_{(d,i)} = z) \\ & = T_1 \times T_2 \ . \end{aligned}$$

Now,

$$\begin{aligned} T_1 &= p(z_{(d,i)} = z | \mathbf{c}_{-d}, \mathbf{a}_{-d}, \mathbf{r}_{-(d,i)}, \mathbf{z}_{-(d,i)}, c_d, a_d, \rho_d, r_{(d,i)} = r) \\ &= \int_{\psi_{c_d a_d r}} \Big(p(z_{(d,i)} = z | c_d, a_d, r, \psi_{c_d a_d r}) \\ &\qquad \times p(\psi_{c_d a_d r} | \mathbf{c}_{-d}, \mathbf{a}_{-d}, \mathbf{r}_{-(d,i)}, \mathbf{z}_{-(d,i)}, c_d, a_d, r) \ d\psi_{c_d a_d r} \Big) \\ &= \frac{n^{(CUU)T}_{-(d,i),(c_d a_d r)z} + \beta}{\sum_{h=1}^{T} n^{(CUU)T}_{-(d,i),(c_d a_d r)h} + T\beta} \end{aligned}$$

Further,

$$\begin{aligned} T2 &= p(w_{(d,i)} = w|\mathbf{z}_{-(d,i)}, \mathbf{w}_{-(d,i)}, z_{(d,i)} = z) \\ &= \int_{\phi_z} p(w_{d,i} = w|\phi_z) p(\phi_z|w_{(d,i)}, z_{-(d,i)})\ d\phi_z \\ &= \frac{n^{TW}_{-(d,i)zw} + \gamma}{\sum_{v=1}^{W} n_{-(d,i),zv} + W\gamma}\ . \end{aligned}$$

That completes the proof. □

3.4 Experiments

We demonstrate the performance of our model on the Enron email corpus. The Enron email corpus[1] is a set of emails belonging to 151 users, mostly senior management of Enron, exchanged between mid-1998 and mid-2002 (approximately 4 years), which includes the Enron crisis that broke out in October 2001. In the current experimental setup, a cleaned version is chosen, in which duplicate, erroneous and junk emails have been removed [38]. The dataset consists of 252,759 email messages. For experimental analysis only those emails (approximately 20,311) which are exchanged between these 151 users were selected. Results were compiled for 8 communities and 25 topics. All the model hyperparameters were initialized with a value of 1. The model was run for a total of 500 iterations and after stabilizing the Markov chain (around 20 iterations), samples were drawn after every 5 iterations.

Of the eight communities extracted, communities 1-4 were more likely to be observed than communities 5-8 (the exact probabilities for communities 1-8 are 0.14, 0.14, 0.16, 0.24, 0.06, 0.09, 0.07 and 0.1 respectively). We observed that for each community, certain *central actors* (central actors are prominent and communicatively active [5]) connect almost all the other actors in that community. This is due to their active communication habits. In our visualizations (see Figure 3), these actors tend to be situated in the central region of the graph.

[1] http://www.cs.cmu.edu/∼enron/

3.4.1 Community Visualization

Figures 3.3 and 3.4 provide visualizations for communities 2 (red) and 4 (green) respectively. The visualization is based on a spring tension model that uses edge weights (based on the number of emails sent between actors). For each community we consider the top 25 users, each of which are assigned colors indicating community membership (green, red, pink, blue, etc.). Any user among the top 25 for more than one community is colored black and any user not among the top 25 for any community is colored white. In both figures, for each community we highlight the subgraph within edge-distance 2 of a chosen central actor. For example, for the green community we highlight the subgraph within edge-distance 2 of node 78,[2] whereas for the red community we do the same using node 73.

Community Structure: From the figures, we can see that when such a central node is picked, most of the top 25 nodes in the community are highlighted and thus can be reached within a distance of 2 (In the green community 21 nodes are reachable and for the red community the number is 24). It is important to observe that all the nodes belonging to the same community that are located in the central portion of the graph are always highlighted. Due to the spring tension model of visualization, any node away from the center has little communication and is relatively (when compared to the central ones) not an active member of the community. Such nodes are the ones missed out when we highlight a subgraph of distance 2 from a central node. This is expected as they are not well connected in general, and need to be reached through a more circuitous path.

Bridging Nodes: There are a few instances where non-community nodes (particularly white colored nodes) act as bridging points between two nodes of the same community (since the highlighted portion includes all nodes within a distance of 2, the number of bridging nodes is actually quite less than the number of non-community nodes highlighted). The presence of these bridging nodes can be explained by the following: (a) Some of these nodes are articulating points, i.e., important hubs and so are responsible for maintaining connectivity. These hubs either do not participate in any community and simply facilitate communication (e.g., node 122 who is a chief operating officer [39])

[2] We chose not to pick a node too much towards the center as such nodes are highly connected and end up highlighting a large portion of the graph along with the community itself, making it more difficult to visualize the communities

or are important hubs but still participate in certain communities (e.g., node 150 is the assistant of Enron president Greg Whalley); (b) They are not in the top 25 but if a larger range was considered they would be included.

The choice of the highlighting node does not affect the results as long as they are close to the center. Communicatively inactive actors, when picked, would highlight the community nodes close to them as well as some of the central nodes, but they often miss nodes which are located further away. These results suggest the proposed model does manage to extract communities such that the communicatively active nodes in them are generally well-connected with each other and act as hubs for connecting the inactive or non-central members of the community.

Table 3.1: Prominent topics extracted from Enron email corpus

Topic 5	0.155	**Topic 9**	0.039	**Topic 10**	0.022	**Topic 11**	0.021
enron	0.021	company	0.006	Sonat	0.003	taliban	0.0008
message	0.011	3d	0.005	dominion	0.002	html	0.0007
original	0.011	germany	0.005	germany	0.002	afghanistan	0.0006
gas	0.008	trading	0.004	mcmichael	0.001	mughniyeh	0.0005
pmto	0.007	nymex	0.003	boyt	0.001	htm	0.0004
fw	0.005	stock	0.002	dth	0.001	terrorist	0.0004
amto	0.005	exchange	0.002	petition	0.001	http	0.0003
Topic 15	0.038	**Topic 16**	0.11	**Topic 17**	0.185	**Topic 21**	0.024
louise	0.007	enron	0.022	enron	0.02	ces	0.005
kitchen	0.005	agreement	0.009	mail	0.01	germany	0.004
john	0.004	sara	0.009	energy	0.008	columbia gas	0.003
mike	0.004	ect	0.008	california	0.007	chris	0.003
meeting	0.003	subject	0.007	power	0.007	cng	0.002
ubs	0.003	corp	0.007	jeff	0.006	transco	0.002
lavorato	0.003	master	0.006	ees	0.006	columbia energy group	0.002

Topic 22	0.024	**Topic 14**	0.051
filename	0.005	ect	0.135
book	0.005	hou	0.065
keiser	0.003	enron	0.044
kam	0.003	corp	0.009
books	0.003	jones	0.007
phillip	0.003	subject	0.007
cad	0.003	pm	0.007

3.4.2 Social Topics

Table 3.1 shows the top 7 words, along with their probabilities, for the most prominent topics discovered by CART in the Enron email dataset. The table also shows the probabilities of occurrence of each topic as well as the probabilities corresponding to the top 7 words given the topic. The dominant topics in the corpus are *Topic 5*, *Topic 16*, and *Topic 17*.

Topic 5 typically consists of common junk words that are encountered in email communication. Many emails contain the terms 'fw' (forwards) and 'original message'. 'Enron' is expected to be quite common and so is also placed in this topic. In many of the emails, the time field is immediately followed by the To field, and as a result the 'am' or 'pm' suffix of the time value is concatenated with 'to' (hence the 'amto' and 'pmto' terms). Note that detection of such a topic shows that, among other things, CART may be helpful in data cleaning.

Topic 16 is more interesting and is about the master agreement for Enron following its filing for bankruptcy. Sara is one of the employees who is actively involved in communications involving the master agreement and so her name shows up as well. 'ect' is short for Enron Capital Resources, one of Enron's subsidiaries. Although a strong topic, it is not as dominating as topics 5 and 17.

Topic 17 is yet another interesting topic which represents the California power crisis. 'Jeff' is the first name of Jeff Dasovich, Enron's Governmental Affairs Executive and 'ees' stands for Enron Energy Services, which played a major role in the California power crisis. The presence of other terms such as 'California', 'power' and 'Enron' is self-explanatory.

Since *Topic 5* consists of junk terms commonly occurring in emails, it is ubiquitous in the social network. *Topic 17* and, to a certain extent, *Topic 16* are related to the Enron crisis, and hence dominate a large percentage of the Enron email corpus.

Certain other less prominent topics were also extracted from the Enron corpus. *Topic 9* is about Enron's participation in the NYMEX (New York Mercantile Exchange). *Topic 10* is regarding Enron's dealings with Sonet (Southern Natural Gas) and Dominion. 'germany', 'mcmichael' and 'boyt' are last names of employees involved in these dealings and 'dth' (decatherm) is a unit of measure for energy widely used by the energy industry. *Topic 11* was about the war in Afghanistan. Communication regarding this

topic consisted of html sources of web documents and so certain terms such as 'http' and 'htm' were also picked up by this topic. *Topic 14* consists of communication involving employee Tana Jones, who is an important hub and possibly an assistant to certain high profile executives. Since a significant portion of communication involves her, her communication patterns are identified as a topic in itself (in the topic 'hou' is short for Houston). *Topic 15* is about UBS's (Union Bank of Switzerland) takeover of Enron Online Services. Louis Kitchen was the president and creator of Enron Online Services. 'Lavorato' is the then Enron CEO's last name and 'Mike' as well as 'John' possibly also represent other people involved. *Topic 16* is similar to *Topic 10* and is in regards to Enron's dealings with Columbia Energy Services ('ces') and the energy transportation firm Transco. Once again the employee Chris Germany emerges. *Topic 22* involves an employee with last name Keiser, and from his communication it seems this employee is responsible for providing energy/power related books and other resource material to members of the organization.

The dominant topic of communication in the Enron email corpus is the Enron crisis, and this is supported by our results as *Topics 16* and *17* are directly related to the same. The model also picks up on several other less important topics and it is likely that in a large organization like Enron many such smaller topics will exist. Overall, the community and author-recipient based topics extracted by the proposed model are meaningful and can be mapped back to their corresponding real-life events involving Enron.

3.4.3 Community Profiles

Topic profiles, across communites, are also presented. Table 3.2 shows plots for topic probabilities given each community. In this table, community 1 focuses on *Topic 17*, whereas most other communities focus on *Topic 16*. *Topic 5* is present across all communities. The plots were constructed using a sample from the Markov chain and so the topic probabilities can vary from the averaged out probabilities in Table 3.1 (for example in the plots *Topic 1* seems to be a little prominent in this sample versus its prominence obtained by averaging many samples). However, the sample is fairly representative and can be used to make general observations regarding community-topic profiles.

From the plots we can see that *Topic 17* is very prominent in *Community 1* as

opposed to *Topic 16* which is far more prominent in all other communities. Note that despite being dominant in a single community, *Topic 17* still dominates *Topic 16* in the entire corpus. This is because each word in an email is associated with a topic and so the length of emails will play an important part in deciding the dominance of a topic. It is quite likely that the number of words assigned to *Topic 17* in emails in *Community 1* are more than the number of words assigned to *Topic 16* in emails in the other communities. Apart from Community 1, all the other communities have similar topic profiles. This is to be expected due to the heavy dominance of *Topics 16* and *17*, and even though there are some differences in the prominence of lower strength topics, their effects are mitigated. If one were to consider only topics then there are two primary communities: *Community 1* (discussing *Topic 17*) and all other communities merged into a single community (discussing *Topic 16*).

Table 3.3 present plots for actor probabilities given each community. We can see that the communities have different profiles for actor participation. The diversity in the profile implies that several actors, though in different communities, are talking about similar things. Despite discussing similar topics, the differences in actor participation profiles across communities 2 to 8 can be accounted for by the social links between actors, which also play an important part in determining community structures.

From the results presented, we can observe that the proposed CART model is capable of extracting well-linked and topically meaningful communities using both the social link and communication content information. Moreover, the probabilistic nature of our model also allows actors to participate in multiple communities, a more realistic assumption compared to limiting each actor to a single community.

3.5 Discussion

In this section we briefly discuss the main issues with evaluating community extraction methods. Techniques developed in the Physics and social science domain (such as [40], [36] [35] and [41]) demonstrate their methods on computer generated random graphs and real world graphs whose community structures are already known. Moreover, these techniques are purely link based techniques which compute hard partitions of the social network graph in order to extract the community structure. In such a scenario it is eaiser

to evaluate community extraction methodologies and a better method should extract communities which agree with the actual community structure. In the absence of actual communities methods that extract communities with higher intra-community linkage and lower inter-community linkage are generally considered to be better. In our case the communities are extracted based on links as well as topics and each user can belong to more than one community (i.e. it is not a hard partitioning of the social network graph). Traditional graph based measures are inappropriate for evaluating goodness of such a community extraction method as the method attempts to find the best trade-off between high linkage and topic similarities between users in a community. For example, in a book reading club if their is a community of 20 people which follows author XYZ's work, then even if these 20 people do not interact much with each other (i.e. low linkage) they are still extracted as a community due to their high topic similarity (i.e. author XYZ). In such a case a purely graph based measure might not be indicative of a good community and novel methods, which take into account links as well as topics, for evaluating goodness of community structure are required. Constructing such measures for evaluating such community extraction methods is a non-trivial issue.

3.6 Related Work

The existing literature on community extraction from social networks is primarily based on the link information of the network. [40] and [36] follow an approach based on iteratively removing highest *betweenness* edges from the social network graph, where betweenness is the number of shortest paths traversing through an edge. The graph is broken into connected components, and each component is checked to see if it is a meaningful community. A second approach, discussed in [35] and [41], is an agglomerative hierarchical algorithm where each node starts out as an individual community and at each step two communities whose amalgamation produces the largest change in *modularity* are merged. Modularity for a given division of nodes into communities C_1 to C_k is defined as $Q = \sum_{i=1}^{k} (e_{ii} - a_i^2)$, where e_{ii} is the fraction of edges that join a vertex in C_i to another vertex in C_i, and a_i is the fraction of total edges that are attached to a vertex in C_i. Recently, [42] has presented an extension of this modularity based approach. Other existing approaches are typically based on such graph partition

schemes, and do not take communication content information into account. Another limitation of such approaches is that each actor's participation is limited to just one community.

Our proposed approach is based on Bayesian generative models from text mining [43, 37]. A recent approach that works with relations between entities is the group-topic (GT) model proposed by [44]. The goal of the GT model is to cluster entities such that entities within a group exhibit similar interaction patterns with entities in another group. If a GT model were to be applied to email communication data, then the result would be a summarization of the underlying social network where for each topic of conversation one would get groups such that actors in one group exhibit similar communication habits with actors in another group. Thus, although the GT model works with related entities with textual attributes on the relations, it attempts to solve a completely different problem and as such is not applicable to community extraction.

The author-recipient-topic (ART) model, proposed by [45], extracts topics based on communication between people. The ART-model works with relations as observed through content communication, and models topics based on author and sets of recipients. As explained earlier, our model naturally builds on the ART model by assuming that the author and recipients of an email belong to the same community in the context of the topic of the email.

The community-user-topic (CUT) models were proposed by [46], where a community is modeled as a joint distribution of topic distributions and user distributions. The model uses Gibbs sampling and entropy computation to filter non-informative samples. The main ideas behind the CUT models are very much similar to our proposed idea in that the CUT models attempt to leverage link as well as communication content information in order to extract communities. However, the underlying semantics of the CUT models are such that there is a loose coupling between how topics and links affect community structure. Specifically, of the two proposed models CUT1 and CUT2, the former is biased towards extracting communities from just the link information and the latter is biased towards extracting communities from just the content information. Community node updates for the CUT models can be derived along similar lines by following the Gibbs sampling update derivations for our proposed model, or the update derivations for the Group-Topic model.

Other examples of related work include using topic modeling for other kinds of social network analysis. Tang et al. [47] present a topic modeling based approach for estimating topic specific influence values. The technique can based to identify topic specific authorities in the network. In [48], the authors present a text mining based approach to discover concepts such as the purpose of users in the social network.

3.7 Conclusions

In this chapter, a Bayesian generative model for community extraction from social networks was presented. The CART model extracts communities based on both communication link as well as content information. Through experiments and visualization on the Enron email corpus, it is demonstrated that the model can extract well connected and topically meaningful communities. Additionally, the model extracts relevant topics that can be mapped back to corresponding real life events involving Enron. Thus, the proposed topic and link based analysis can be used to extract the context from given textual communication data. The topics extracted are the ones responsible for reinforcing the social links between users and the communities extracted provide knowledge relatng the set of users to these topics. This information is valuable to a variety of parties including online business owners and community leaders, as it helps judge the interests of users in the social network.

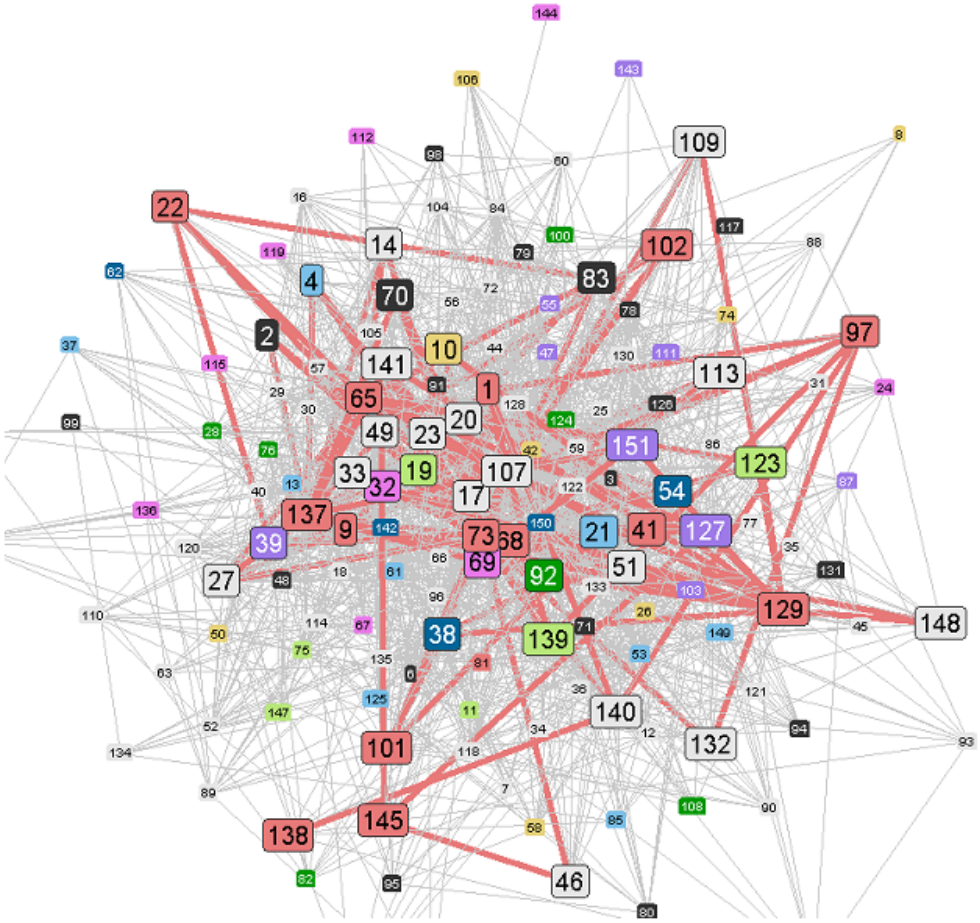


Figure 3.3: Visualizing the red community

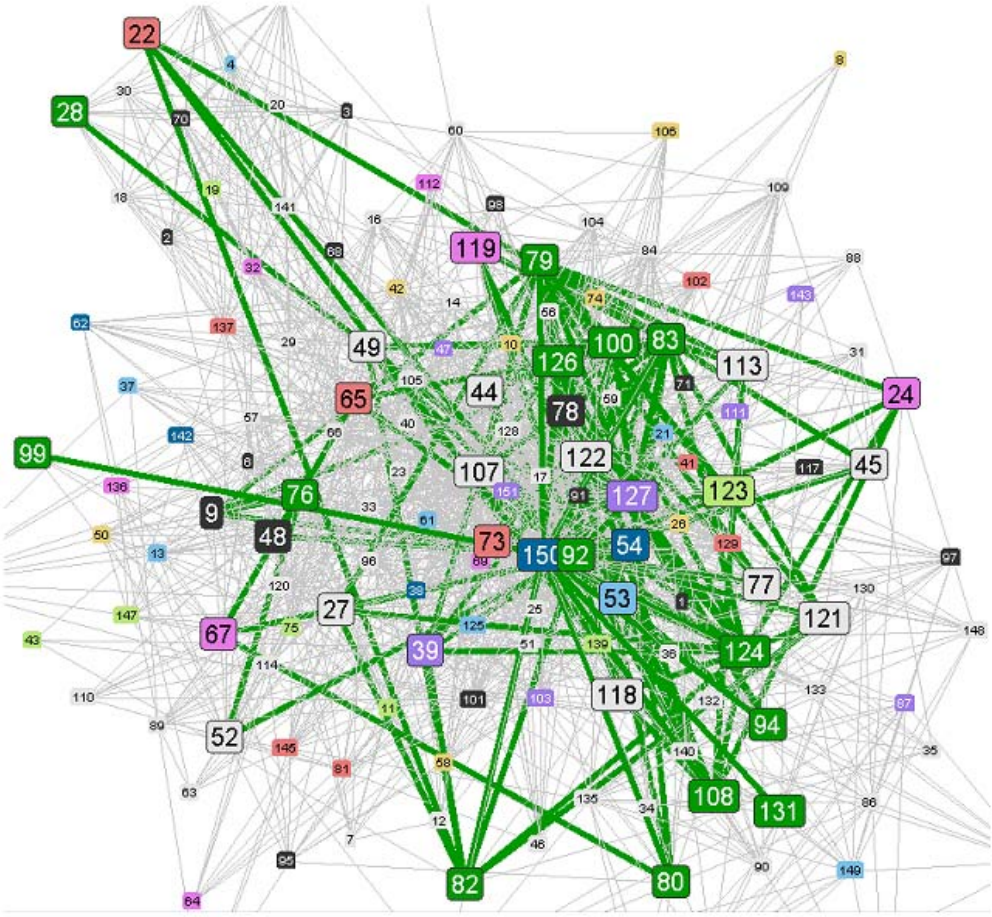
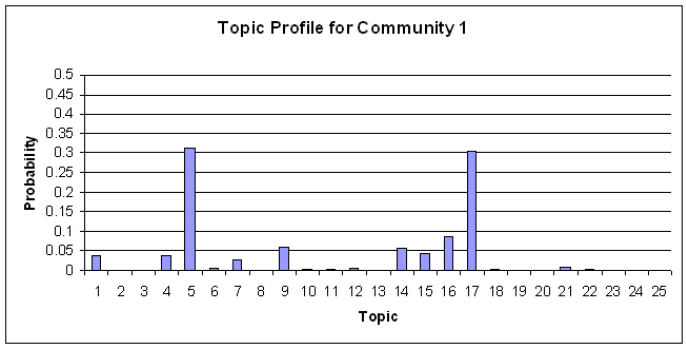
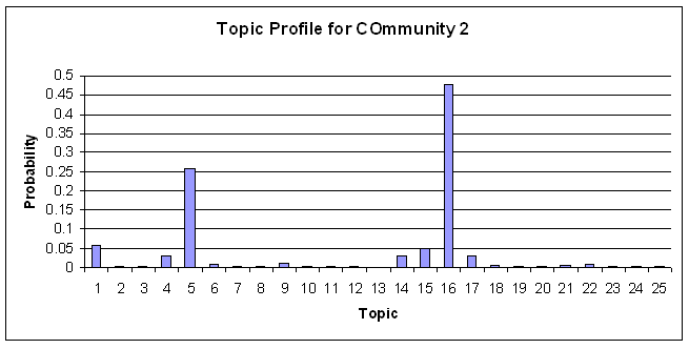
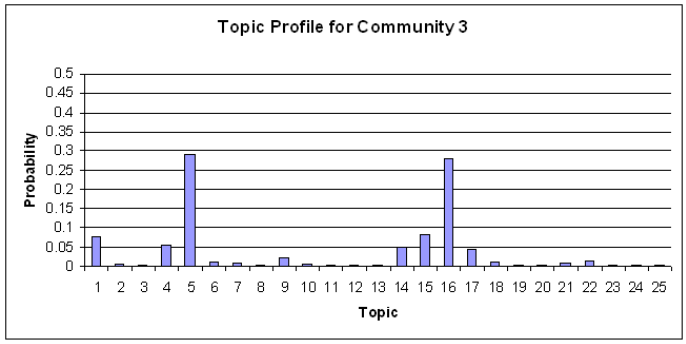
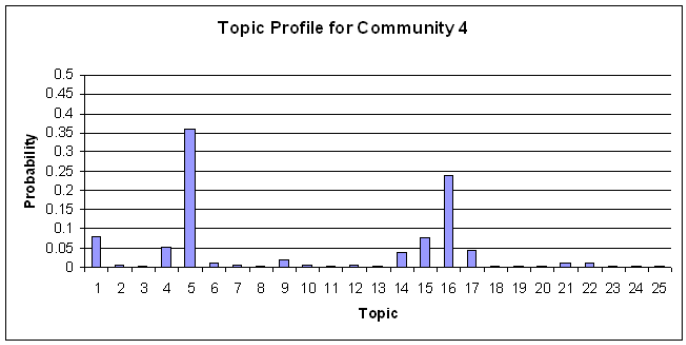


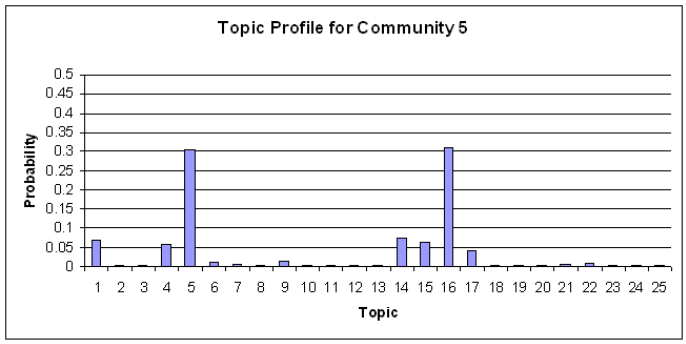
Figure 3.4: Visualizing the green community

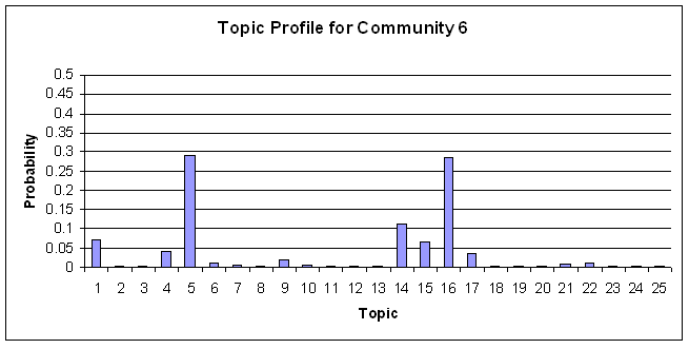

Topic Profile for Community 1
Probability
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Topic

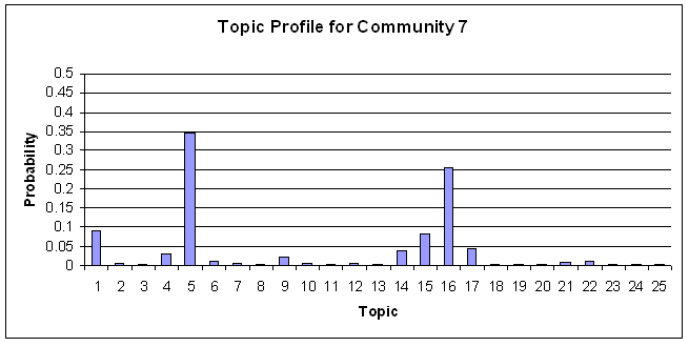

Topic Profile for COmmunity 2
Probability
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Topic


Topic Profile for Community 3
Probability
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Topic


Topic Profile for Community 4
Probability
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
Topic







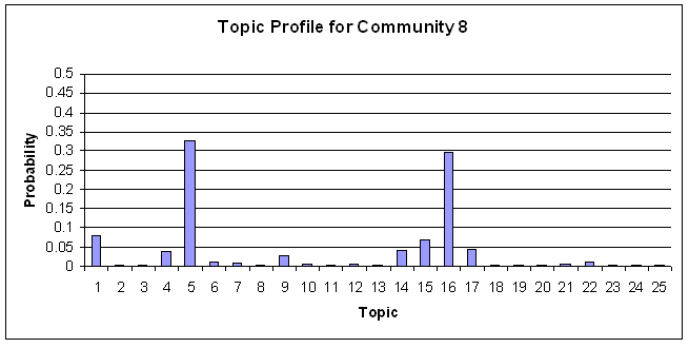
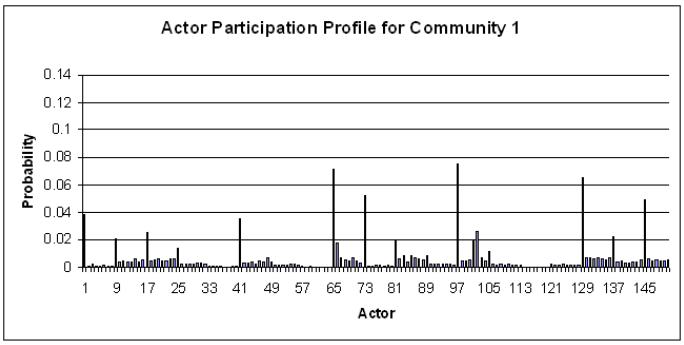
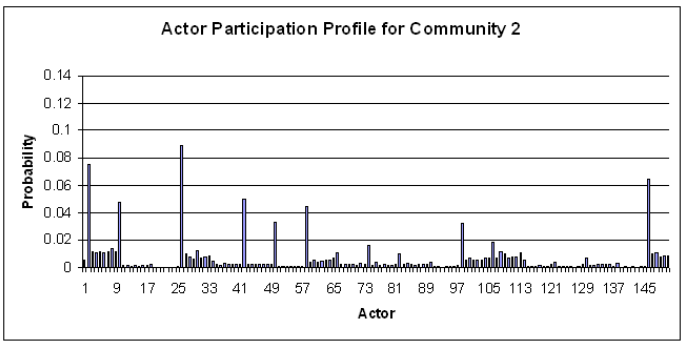
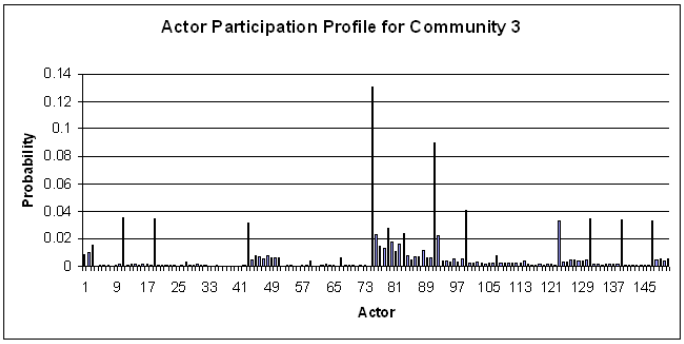
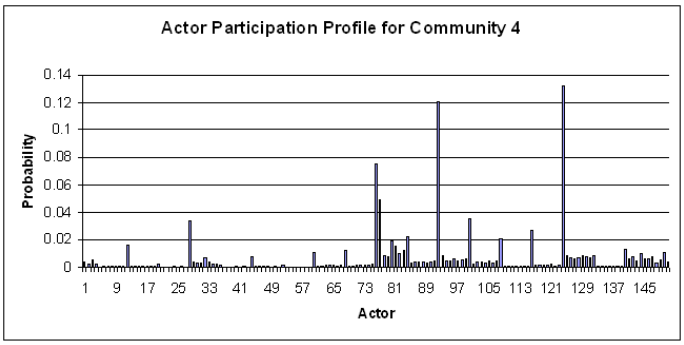


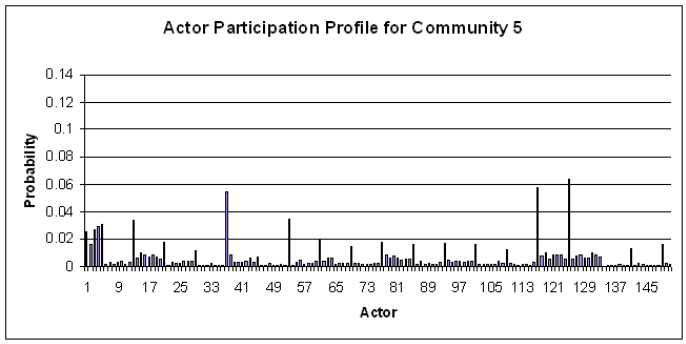
Table 3.2: Topic profile, $p(z|c)$, for each community.

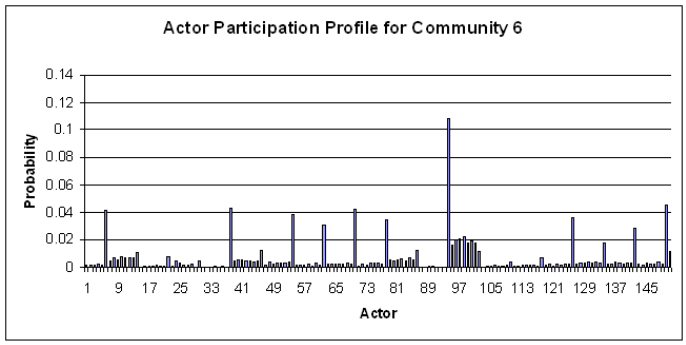

Actor Participation Profile for Community 1
Probability
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137 145
Actor

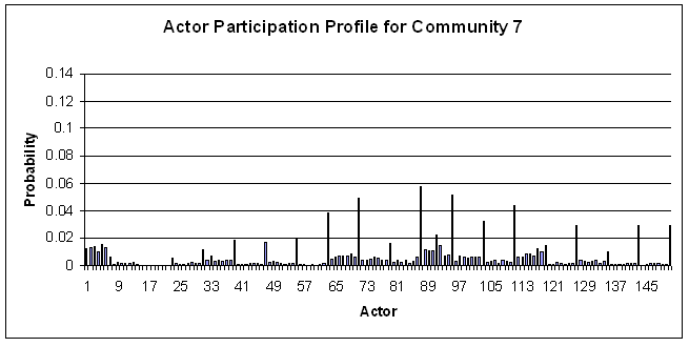

Actor Participation Profile for Community 2
Probability
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137 145
Actor


Actor Participation Profile for Community 3
Probability
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137 145
Actor


Actor Participation Profile for Community 4
Probability
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1 9 17 25 33 41 49 57 65 73 81 89 97 105 113 121 129 137 145
Actor







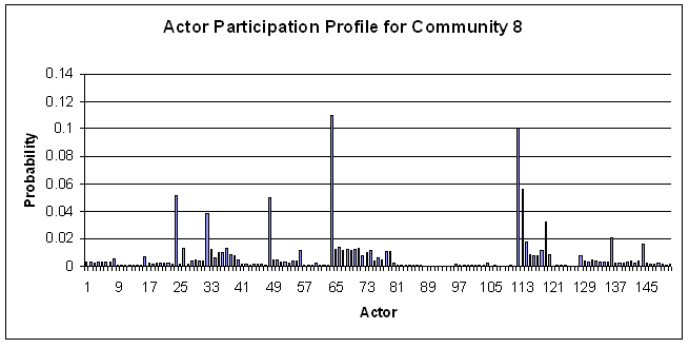


Table 3.3: Actor profile, $p(u|c)$, for each community.

Chapter 4

Modeling Socio-Cognitive Networks

4.1 Introduction

In previous chapters the phenomenon of information flow in social networks was looked at from the network topology as well as topical context perspectives. This chapter concentrates on analyzing the cognitive aspects of a social network and how it relates to information exchange. Broadly speaking there are two aspects to an actors cognition in a social network. The first is cognition pertaining to the social influence/behavior exercised by other actors in the same social network, i.e. what does the actor under consideration know about the social influence/connections of the other actors in the network. Such a network is called a socio-cognitive network (or informally as a who thinks, who knows who network). There are many applications for such networks, particularly in studies pertaining to organizational dynamics ([49], [6], [50]). Second is the analysis of actors' cognition to determine which individual (who) has what information (knowledge), and also how this knowledge spreads in the network. Thus, a step further is to understand "who knows what", which is referred to as a cognitive knowledge network ([51]). Understanding such networks is important in any organizational/group effort as efficient exchange of information among different individuals fosters collaboration and ultimately results in a healthy community.

In this chapter socio-cognitive networks are modeled to track actors' perceptions

regardng relationships in the network. Novel measures using such a socio-cognitive network model are described and applications of the same for knowledge discovery regarding information flow in the network are demonstrated on the Enron email dataset. Cognitive knowledge networks will be discussed in the next chapter.

4.2 Background

Interpersonal interaction plays an important role in organizational dynamics, and understanding these interaction networks is a key issue for any organization, since these can be tapped to facilitate various organizational processes. Widespread use of computer networks for organizational communication provides a unique opportunity to overcome these difficulties and automatically map the organizational networks with a high degree of detail and accuracy. Most research has focused on understanding and analyzing direct social relations i.e. "who knows who" type interactions, and illustrating the importance of such information for an organization. Taking this a step further is socio-cognitive network analysis, which analyzes "who thinks who knows who" in the social network. Based on observed communication (e.g. emails), an actor forms his/her beliefs of probabilities of communication between different actors. Each actor in the network thus maintains his/her beleifs regarding the communication network based on the emails observed by him/her. The set of such networks for all actors is defined as a *socio-cognitive network* (illustrated in Figure 4.1).

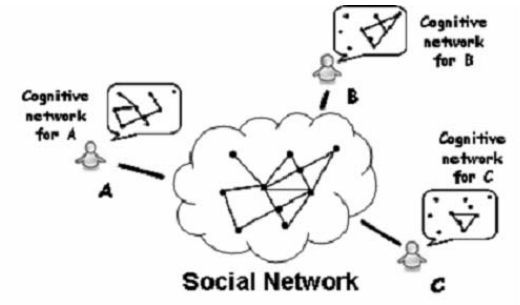


Figure 4.1: Actors perceptions of a social network (Socio-cognitive network)

Modeling of such socio-cognitive networks using electronic communication logs is a novel, challenging problem. For such networks, the model must account for different

social interactions perceived by each actor and mapping them to corresponding actors beliefs. Data sparseness is another issue that needs to be addressed in such an approach. The social bandwidth of an actor limits the number of actors that can be interacted with, resulting in very less interactions between most actors in an actors cognitive network. The dynamic nature of social interactions results in changes in actors cognitions about the social network. For static analysis of socio-cognitive networks, the choice of time window may affect the data sparseness as well as bias in an actors cognition. Smaller time windows exacerbate data sparsity whereas large window sizes may incorrectly show high perceptions about social ties. The proposed approach highlights and discusses all these key issues and presents a model as well as measures for socio-cognitive networks. Email communication data is then used to illustrate applications of socio-cognitive network analyses using the proposed approach.

4.3 Modeling a Socio-cognitive Network

The model captures communication between actors as non-stationary Bernoulli trials and then applies Bayesian inference to estimate model parameters over time. As against a more sophisticated model, this proposed simple model provides a scalable approach that is later shown to perform reasonably well on real data. Every actor participating in the email communication network maintains beliefs regarding the email communications. These are beliefs regarding who communicates with whom, based on the emails that the actor observes. Consider an email communication network consisting of N actors denoted by the set $\mathbf{A} = \{A_i : 1 \leq i \leq N\}$

Let $P_i = P(Sender = A_i)$ denote the probability that an email is sent by actor A_i. Then we have,

$$P_i = \frac{Number\ of\ emails\ sent\ by\ A_i}{Total\ number\ of\ emails\ sent\ in\ the\ network} \tag{4.3.1}$$

Since each email has a unique sender, the events corresponding to an email being sent by different actors are mutually exclusive.

Let $P_{j|i} = Pr(A_j \in Recipients|Sender = A_i)$ denote the probability of A_j being a recipient of an email, given that A_i has sent that email, i.e.

$$P_{j|i} = \frac{Number\ of\ emails\ sent\ by\ A_i\ and\ received\ by\ A_j}{Total\ number\ of\ emails\ sent\ by\ A_i} \tag{4.3.2}$$

Thus, $P(i,j)$, the probability that an actor A_i sends an email to another actor A_j is defined as,

$$P(i,j) = P_{j|i} \times P_i = \frac{Number\ of\ emails\ sent\ by\ A_i\ and\ received\ by\ A_j}{Total\ number\ of\ emails\ sent\ in\ the\ network} \tag{4.3.3}$$

This represents the strength of the actor A_i's communication with actor A_j. The event of an actor A_i being the sender and A_j being a recipient of an email is mutually exclusive to its complement, i.e. the event where either A_i is not the sender or A_j is not a recipient or both. The probabilities of these two events are represented as $P(i,j)$ and $1-P(i,j)$ respectively. We define a Bernoulli distribution over these two events corresponding to email communication between actors A_i and A_j, i.e., $L(i,j) = [P(i,j), 1-P(i,j)]$. where $P(i,j)$ is the parameter of the Bernoulli distribution $L(i,j)$. For the communication network perceived by an actor, there will $N(N-1)$ such distributions, one for every ordered pair of actors $(A_i, A_j), A_i \neq A_j$. Every email exchanged in the network is a Bernoulli trial and every actor maintains a distribution over all possible probabilities $P(x,y)$ for a given ordered pair (A_x, A_y). To capture this information, a Beta distribution is used, with a Bayesian update applied on the parameters of the Beta distribution for maintaining actors' "beliefs".

Definition 4.3.1. (Belief State): The belief state of an actor is defined as a set of $N(N-1)$ Beta distributions, where each Beta distribution $J(i,j)$ is defined over the corresponding Bernoulli distribution $L(i,j)$ representing email communication between actors A_i and A_j.
Thus, the belief state B_k for a given actor A_k is given as

$$B_k = \{J(i,j)_k \mid \forall\ ordered\ pairs\ (A_i, A_j)\ such\ that\ A_i \neq A_j\} \tag{4.3.4}$$

where $J(i,j)_k$, is a Beta distribution over the parameter of $L(i,j)$ and is defined as A_k's belief about probability of email communication from A_i (sender) to A_j (recipient).

Each Beta distribution $J(i,j)_k$ in belief state B_k of an actor A_k has two parameters, $\alpha(i,j)_k$ and $\beta(i,j)_k$. Based on the communication A_k observes, A_k updates the parameters for all $J(i,j)_k$ in B_k. We associate the parameter $\alpha(i,j)_k$ with number of emails, observed by A_k, that have been sent by A_i to A_j, and the parameter $\beta(i,j)_k$ with the number of emails observed by A_k for which either A_i is not the sender or A_j

is not a recipient or both. Each actor A_k starts with an initial belief state B_k, with parameters for all distributions having default prior values, and as actor A_k observes email communication, actor A_k updates his/her belief state.

4.4 Non-stationarity and Time Windows

The communication probabilities between actors are dynamic in nature, i.e. they may change over time. Hence, a Markov time window based approach is described to handle the non-stationary nature of Bernoulli probabilities. At the start of each time window, the parameters for all Beta distributions in a given actors belief state are scaled down by a parameter λ ($0 \leq \lambda \leq 1$). These scaled down posterior parameters from the current time window are used as priors for the next time window. The model parameter λ regulates how much of history is "remembered" by an actor. Higher is the value λ, more is the weight (importance) given to history.

Another important parameter in this model is the length of time window. This problem is similar to the classical problem of segmenting time series in temporal data analysis, since the vector of communication probabilities is analogous to a time series dataset and each segment is analogous to a time window. A Bayesian belief update for each actor occurs at the end of each time window. The choice of length of time window affects the number of emails observed in that time window, and hence the interpretation of results. We assume that the length of time window is a user-specified parameter, and this choice provides sufficient number of emails within each time window. Other approaches such as varying time window length and/or updating different actors belief states at end of different time windows may be adopted, but remain open problems for future research.

To model the temporally varying nature of beliefs, we denote the belief state of an actor at time t as $B_{k,t}$.

Definition 4.4.1. (Belief State at time t): The belief state for the given actor A_k at the given time t, is defined as,

$$B_{k,t} = \{J(i,j)_{k,t} | \ \forall \ ordered \ pairs \ (A_i, A_j) \ such \ that \ A_i \neq A_j\} \tag{4.4.1}$$

where, $J(i,j)_{k,t}$ is the Beta distribution for an ordered pair of actors (A_i, A_j), maintained

by actor A_k at time t.

The belief state of a given actor at time t reflects what the actor believes are the probabilities of the possible strengths of different actors communications in the network at time t. A socio-cognitive network at a given time t is the set of belief states of all actors at that time.

4.5 Socio-cognitive Network Analysis

An actor sees only a part of actual emails exchanged in the network, hence, his/her egocentric beliefs may be different from the sociocentric beliefs for that network. Such analysis is motivated by Simpsons paradox, which states that "A statistical measure computed over a complete population (global analysis) may give completely contradictory conclusion to that of the conclusion obtained from the statistical measure computed over the sub-populations (local analysis)." In addition, the personal beliefs of different actors may differ markedly. Therefore, an important component of belief analysis is to assess the *belief conflict analysis* or *divergence analysis* between (i) the personal beliefs of an actor and the sociocentric beliefs (ground 'reality); as well as (ii) the personal beliefs of different actor. Measures need to be developed to allow comparison of perceptions across actors i.e. measures that lead to quantification of notions such as agreement, consensus, and closeness to reality. The similarity across beliefs of different actors was previously studied in the social sciences domain, using a measure called *perceptual congruence* ([49]). However, the analysis of divergence between actors beliefs and 'ground truth is a novel. In traditional analyses, there was no way of determining the 'ground truth, but in electronic communication analyses, it is possible to observe both actors beliefs as well as ground truth (email server observes all communication and thus has a sociocentric view).

In this sub-section, we present two useful socio-cognitive analyses that are performed using the model proposed in the previous section. The model is used to define novel measures for analysis of (i) closeness between actors perceptions about such organizational networks (agreement), (ii) divergence of an actors perceptions about organizational network from reality (misperception).

4.5.1 Divergence between Beliefs

Given belief states $B_{x,t}$ and $B_{y,t}$ for two actors A_x and A_y at time t, there is a need to measure the similarity between these belief states to determine the similarity between their perceptions. Since $B_{x,t}$ and $B_{y,t}$ are vectors of probability distributions, for computing divergence between $B_{x,t}$ and $B_{y,t}$, the divergence between respective pairs of beliefs in the two sets are computed and then combined. The divergence between respective beliefs of two actors is defined as the KL-divergence across the expected Bernoulli distributions for respective two beliefs. The expected Bernoulli distribution for a belief is the expectation of Beta distribution corresponding to that belief.

Since KL-divergence is an asymmetric measure, the symmetric KL-divergence $KL_{sym}(q||p)$ is used and is defined as $KL_{sym}(q||p) = KL(q||p) + KL(p||q)$

Definition 4.5.1. The similarity between beliefs for the email communication from A_i to A_j for actors A_x and A_y, expressed by Beta distributions $J(i,j)_{x,t}$ and $J(i,j)_{y,t}$, at time t, is defined as,

$$Sim(J(i,j)_{x,t}, J(i,j)_{y,t}) = \frac{1}{1 + KL_{sym}(E[J(i,j)_{x,t}]||E[J(i,j)_{y,t}])} \tag{4.5.1}$$

where, $KL(E[J(i,j)_{x,t}]||E[J(i,j)_{y,t}]) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$
and $p = \frac{\alpha(i,j)_{x,t}}{\alpha(i,j)_{x,t} + \beta(i,j)_{x,t}}, \qquad q = \frac{\alpha(i,j)_{y,t}}{\alpha(i,j)_{y,t} + \beta(i,j)_{y,t}}$

This similarity between two beliefs ranges from 0 to 1, with 0 and 1 indicating minimum and maximum similarity respectively. Based on this definition a novel measure, a-closeness, is defined to quantify similarity in perceptions of two actors.

Definition 4.5.2. (a-closeness): The *a-closeness* measure is defined as the agreement between belief states $B_{x,t}$ and $B_{y,t}$ of actors A_x and A_y respectively at time t and is given by,

$$a - closeness(B_{x,t}, B_{y,t}) = \frac{\sum_{\forall (i,j) \in B_{x,t} \cap B_{y,t}} Sim(J(i,j)_{x,t}, J(i,j)_{y,t})}{\sqrt{n(B_{x,t}) n(B_{y,t})}} \tag{4.5.2}$$

where $n(B_{x,t})$ represents the number of beliefs (communication links) for which actor A_x has observed at least one email and $n(B_{y,t})$ represents the number of beliefs (communication links) for which actor A_y has observed at least one email.

The a-closeness for two belief states is symmetric and ranges between 0 and 1, with lower values representing lesser closeness and higher values representing more closeness. It attains a maximum value of 1 when the two belief states are identical.

The numerator in equation 4.5.2 sums up the similarity between only those beliefs for which both A_x and A_y have observed at least one email[1] . The intuitive reasoning for this is now explained. An email communication network is usually quite sparse, i.e. out of all possible ordered pairs of actors, only a few of them may actually communicate. Hence, the belief states of the actors being compared may be even sparser and for both the actors, the beliefs associated with majority of communications will indicate very low probability of occurring (since no instances of these interactions have been observed). In such a case, it is desirable to disregard such beliefs while measuring similarity between actors' belief states. The situation analogous to computing document similarity, where one computes similarity based only on those words that are present in both the documents. Also, if the whole set of beliefs is considered for every actor, one implicitly assumes that the every actor is equally aware of the presence of all actors as well as all relations in the social network, which is usually unrealistic. The denominator normalizes the numerator with the geometric mean of number of beliefs for which each actor has observed at least one email.

An application of a-closeness is to construct a graph, called *"agreement graph"*, where nodes represent actors while an edge exists between two nodes if the a-closeness measure between those actors is greater than a user-specified threshold μ. This graph captures information about which pairs of actors have similar perception about the email communication network. Classical social network analysis techniques can be applied to such a graph. For example, cliques represent groups of actors having similar beliefs of email communication networks, bridges (i.e. actors sharing beliefs with two cognitively disjoint groups), star structures identify the central actors (i.e. actors in agreement with many other actors) etc. Other global properties such as connectivity, number of components, clustering index, average length of shortest paths etc of the graph can also reveal interesting information.

[1] Other interpretations of closeness between belief states are possible and remains an open research problem.

4.5.2 Divergence of Beliefs from Reality

For comparing an actor's perceptions to "reality", the concept of a *"super-actor"* is introduced, i.e. an actor who observes all the communication in the network. The email server is an example of a super-actor. [2] . Thus, the super-actor's belief state is a benchmark for reality (ground truth) under the closed world assumption. Thus, to study of divergence between an actors belief state and the super actors belief state (reality), a novel measure, called *r-closeness*, is defined below.

Definition 4.5.3. (r-closeness): The *r-closeness* measure is defined as the closeness of an actor A_x's belief state $B_{x,t}$ to super-actor's belief state (reality) $B_{S,t}$ at time t and is given by,

$$r - closeness(B_{x,t}) = a - closeness(B_{x,t}, B_{S,t}) \tag{4.5.3}$$

Higher is the r-closeness for an actor, more realistic are the actors perceptions about the email communication in the network. The mean r-closeness across all actors provides an aggregate measure of the "overall knowledge" or "level of perception" in the network. Higher is the mean r-closeness, the more actors in the network actually know about other actors' communications, i.e. the communication is transparent. A lower mean value for r-closeness indicates that actors generally have misperceptions regarding other actors' communications. The later is usually expected to be observed for a large social network consisting of various diverse groups, where it is difficult for a single actor to keep track of all the communication in the network.

4.6 Experiments on the Enron E-mail Corpus

It is difficult to justify the "goodness" of a solution i.e. how well a model represents the cognition of actors. Hence, in this sub-section, using a real-world dataset, we show how well the proposed model and its analysis explains the events that were observed.

[2] A closed world assumption is made wherein all email communication is said to be sent through the email server, hence observed by it and no other email communication occurs between the actors. This assumption will be relaxed in future research

4.6.1 Data description

The Enron email corpus (`http://www.cs.cmu.edu/~enron/`) is a set of emails between 151 users, mostly senior management of Enron, exchanged between mid-1998 and mid-2002 (approximately 4 years), which includes the Enron crisis that broke out in October 2001. For current experiments, a cleaned version was chosen, in which duplicate, erroneous and junk emails were removed ([38]). The data consists of 252,759 email messages for the set of 151 users. For this experimental analysis, first the entire set of 151 users was chosen, and then only those emails (approx. 20,311) which are exchanged between these 151 users were selected. The length for the time window was chosen to be one month. Results for different values of $\lambda \in \{0, 0.5, 1\}$ were compiled, where $\lambda = 0$ represents no history, $\lambda = 1$ includes complete history and $\lambda = 0.5$ represents an exponential decay of history.

4.6.2 Results on a-closeness

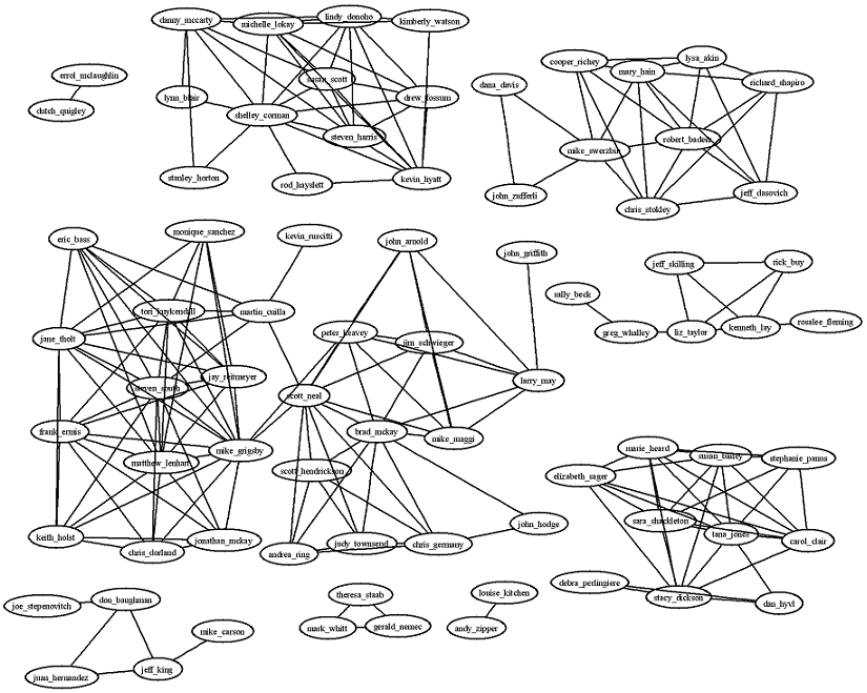


Figure 4.2: Agreement Graph for October 2000 ($\mu = 0.25$, $\lambda = 0.5$)

An agreement graph for the socio-cognitive network is constructed using the a-closeness of actors (employees) at end of October 2000 and October 2001. An edge is drawn between two actors only if the a-closeness between them was more than a certain threshold $\mu \in \{0.25, 0.5, 0.7\}$. In general, the a-closeness values between actors were observed to be low. Figure 4.2 shows the agreement graph for October, 2000[3] . It was observed that the graph consists of many small, disjoint components of actors. A possible reason for this was because big organizations like Enron usually have many organizational groups with high intra-group communication and low inter-group communication. Interesting structures like cliques, bowties and stars were observed in the agreement graph. As the threshold μ was increased, the components were observed to become smaller with some components being split into two or more components. But, there was no change in the general trend of observing "small disjoint components" for the month of October 2000. Except for a few micro-level changes in edges, no significant changes were observed for different values of λ and almost the same clusters of actors were observed. A reason for such lack of changes with λ may be due to the nature of underlying dataset, the nature of analysis (looking mainly at macro level statistics and trends) as well as the choice of time window length. For smaller time windows or for other datasets, interesting, unexpected changes may be observed for different values of λ.

Figure 4.3 shows the agreement graph for October 2001, whose main feature is the one large, connected component. This indicates that there is a considerable extent of the overlap in social perceptions during the crisis period. The connectivity of October 2001 agreement graph also indicates that communication (and hence information) is shared among various actors and pairs of actors were a "few hops" away from each other in terms of cognitive overlap. Such a network is highly conducive towards dissemination of ideas in a social network. Indeed, in case of Enron dataset, the Enron crisis was a "hot topic" that was often discussed in the underlying social network. Also, that the number of nodes in the October 2001 graph was much more than that of the October 2000 graph. Note, an actor was included in the agreement graph only if its a-closeness with at least one actor, crossed the threshold.

[3] Since similar results were obtained for different values of μ and λ, the agreement graphs for only $\mu = 0.25$ and $\lambda = 0.5$ are shown here.

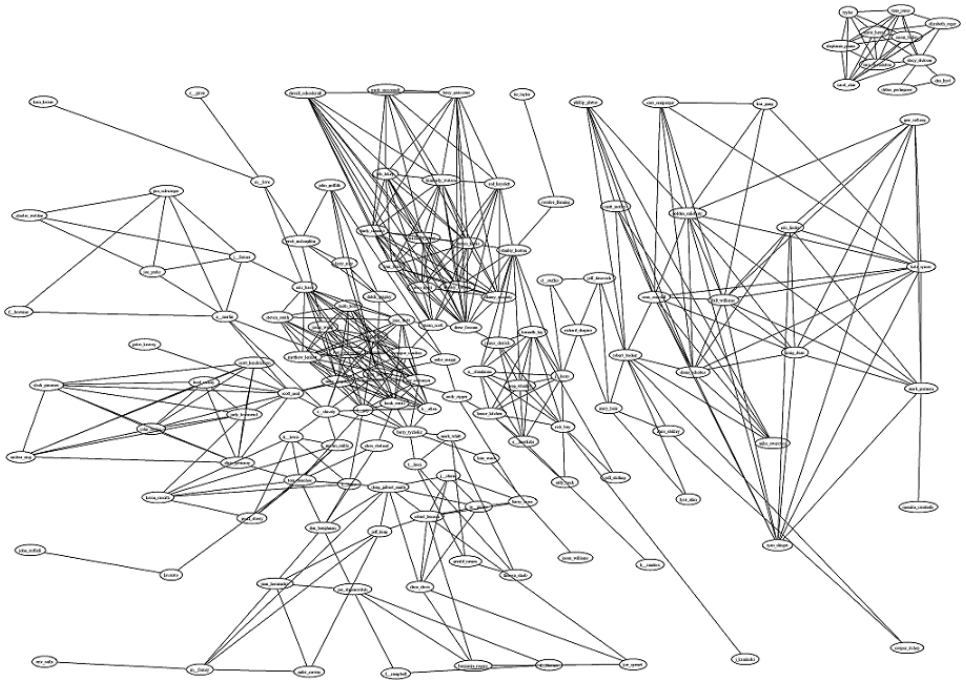


Figure 4.3: Agreement Graph for October 2001 ($\mu = 0.25$, $\lambda = 0.5$)

We also observed interesting structures such as cliques of actors having quite close r-closeness values, and persistent cliques (cliques that exist in both October 2000 and October 2001). For example, a clique of traders such that all traders had a similar low r-closeness measure shows there was agreement in the perceptions of the group, but the entire group is far removed from reality. The second example is a clique of employees which is a persistent clique (disconnected clique in the top right corner of Figure 4.3). For actors present in such a persistent clique, there is probably a strong correlation in their roles, like all such actors worked on the same project. Though insufficient knowledge regarding the domain of data limits the understanding of causes for such structures, the proposed methodology holds promise in finding interesting patterns/structures, from a socio-cognitive perspective, in email communication data that traditional approaches fail to capture.

4.6.3 Results on r-closeness

The r-closeness across actors was examined for two different months, October, 2000, a month with normal email activity in the organization, and October, 2001, a month during the Enron crisis. In each case, users were ranked in the decreasing order of r-closeness. For October, 2000, the actors could roughly be divided into three categories. The first category consists of actors who were communicatively active and observe a lot of diverse communications. These actors occupied the top positions in the rankings. These were followed by the second category actors who also observed a lot of communication. However, their observations were skewed, which lead to skewed perceptions. The third category consisted of actors who were communicatively inactive and hardly observed any communication. These actors had low r-closeness values and were at the bottom of the rankings table. Table 4.1 summarizes the percentages of various actors (according to their formal positions) in the different ranges of r-closeness rankings. Using the rankings for October 2000, the following two statements, relating actor preceptions with hierarchy and observations, were examined.

Table 4.1: Users in different rank ranges of r-closeness (October 2000, $\lambda = 0.5$).

Ranks	N/A	Employee Emp.	Higher Mgmt.	Exec. Mgmt.	Others
1-10	10.3% (4)	4.9% (2)	7.1% (2)	3.4% (1)	7.1% (1)
11-50	17.9% (7)	41.5% (17)	14.3% (4)	31.0% (9)	21.4% (3)
51-151	71.8% (28)	53.6% (22)	78.6% (22)	65.6% (19)	71.5% (10)

S1. *Higher is an actor in the organizational hierarchy, better is his/her perception of the social network.*

From the r-closeness rankings, it is observed that majority of the top positions are not occupied by higher level executive employees. A large chunk of the top 50 ranks consists of employees (around 46.4% of the employees) along with 21.4% of the higher management and 34.4% of the executive management actors (see Table 4.1). A related observation is that most of the higher level executives are communicatively inactive and therefore have fewer perceptions.

S2. *The more communication an actor observes, the better will be his/her perception regarding the social network.*

It is observed that even though some actors observe a lot of communication, they are still ranked low in terms of r-closeness. A main reason for this is that actors tend to participate in only certain sub-communities and less in others. This results in perceptions about the social network that are skewed towards these "favored" sub-communities. Executive management actors observing a lot of communication showed a tendency for this "skewed perception".

Table 4.2: Users in different rank ranges of r-closeness (October 2001, $\lambda = 0.5$).

Ranks	N/A	Employee Emp.	Higher Mgmt.	Exec. Mgmt.	Others
1-10	5.1% (2)	2.5% (1)	3.6% (1)	20.7% (6)	0% (0)
11-50	23.1 % (9)	26.8% (11)	28.6% (8)	37.9% (11)	7.1% (1)
51-151	71.8% (28)	70.7% (29)	67.8% (19)	41.4% (12)	92.9% (13)

Table 4.2 summarizes statistics for r-closeness rankings for month of October 2001. The rankings for the crisis month October 2001 were significantly different from those of October 2000. For both months, the distribution of various actors among the r-closeness rankings was only slightly different for different values of λ (hence, only the results for $\lambda = 0.5$ are discussed). For all values of λ, it was observed that the percentage of management staff among the top 50 ranks increased significantly at the cost of employees being pushed down. Thus, a shift from the normal behavior is observed, indicating that communication perceived by most management level actors is more diverse and evenly distributed as compared to the skewed or no perceptions in October 2000. A possible explaination for this is that during the crisis month, emails were exchanged across different levels of formal hierarchy in the organization, thus exposing management level actors to more diverse communication ([39]). Another possible and intuitively appealing reason (again discussed in [39]) is that during October 2000, on an average, management people sent about 80% but received only 20% of the total email exchanges they participated in. In October 2001, there was a reversal and management people sent only

20% and received about 80% of their total email communication. Since they observed a lot more communication during later period, there was a significant increase in the r-closeness ranks of management level actors during October 2001. Finally, management level actors were also lot more communicatively active in October 2001 than in October 2000 (i.e. they were exposed to a lot more communication during the crisis period and so 80% of October 2001 is greater than 20% of October 2000).

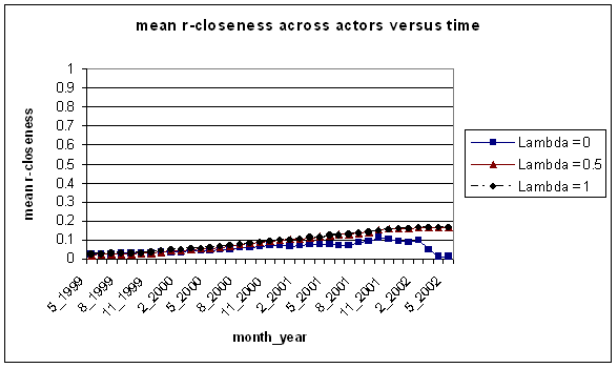


Figure 4.4: Mean r-closeness across all actors at different points in time

Figure 4.4 is a plot of mean r-closeness of all actors over time for different values of λ. An interesting observation was that, for $\lambda = 0$, the mean r-closeness peaks during the crisis month of October 2001, indicating a general increase in the perception of social interactions during the crisis period. After the crisis period, mean r-closeness drops down. For $\lambda > 0$, the plots are almost identical and it was observed that r-closeness increases until the crisis period and after that it stabilized. This could be attributed to increased communication among actors. As almost each actor in the network was involved in some communication, it resulted in increase in the general awareness of an actor. The difference in observation for $\lambda = 0$ and $\lambda > 0$ is due to memory effect introduced by taking $\lambda > 0$.

4.7 Conclusion

The research presented in this chapter shows how a socio-cognitive above model contributes towards measuring similarities in individual actors' perceptions and using it to construct agreement graphs between actors. The proposed analysis can also be used for

quantifying how close are actors' perceptions to ground truth. The agreement graphs along with the r-closeness scores allow estimating the extent of information exchange occurring in a given set of communication logs. While the focus generally is on the "who-knows-who" case, the socio-cognitive aspect will eventually be looked at, as its adds a new dimension and the knowledge discovered from analyzing actor perceptions will additive to that from traditional social network analysis. However, as discussed in this chapter, there are variety of non-trivial challanges pertaining to computation as well as modeling the dynamics of the process that must be examined in significant detail before more sophisticated methods can be developed. This work will further motivate research in developing new computational tools and more sophisticated approaches for electronic communication based socio-cognitive modeling and analysis.

Chapter 5

Modeling a Cognitive Knowledge Network

5.1 Introduction

The previous chapter focussed on a socio-cognitive model where actors' perceptions of relationships were analyzed. Continuing along the same cognition theme, this chapter discusses the perception of knowledge in a social network. Care must be taken as the word knowledge itself is a broad term and can be applicable to many things. Inspired by the fields of belief logic, in this study the notion of knowledge is limited to true/false logical propositions which allow for basic models of reasoning, deduction and evidence combination. A logical proposition of interest is a statement, pertaining to actors or topics of interest in underlying social network, which can be either true or false [52]. Actors believe it to be true or false with a certain probability and in addition, they can be uncertain about this probability value. Different actors, in a network, associate different probability values with propositions being true or false and these probabilities are representative of their sentiments regarding those propositions. For example, in a financial organization one might be interested in knowing what degree of veracity different actors associate with the statement "the stock market is bullish". Other propositions may be whether an actor thinks another actor is an expert on a particular topic. Ideally, one would like to have arbitrary propositions but even if the propositions of interest are fixed, the analysis of actors cognitions is still a hard problem. This is due to the

complex nature of the text and sentiment analysis techniques required. However, the sentiment-proposition framework is quite general in nature and covers a large number of knowledge based cognition scenarios.

A system is presented for constructing cognitive knowledge networks from email communication data, in a sentiment-proposition framework. The proposed approach is based on the Dempster-Shafer theory of evidence and models individuals' perceptions about knowledge. As actors exchange emails with each other, the model accumulates the information in these emails and estimates each actor's perception/opinion regarding different propositions.It is then shown how correlating the knowledge of two or more individuals can help to identify discrepancies between them, and thus identify sources of organizational misperceptions. The proposed approach has been evaluated on the publicly available e-mail logs from the Enron Corporation.

5.2 Representing a Cognitive Knowledge Network

Knowledge is represented as a set of propositions which can be true or false. Each actor entertains certain beliefs regarding the veracity of each of these knowledge propositions. The cognitive knowledge network can be represented as a bipartite graph with actors on one side and knowledge propositions on the other. Edges connecting actors to the knowledge propositions carry labels indicating the degree of veracity an actor associates with a particular knowledge proposition (see Figure 5.1). To express this "degree of veracity" that an actor associates with a knowledge proposition, the Dempster-Shafer theory of evidence combination ([53], [54]) is used. Each edge carries a label of the form (s, p). According to Dempster-Shafer theory, s and p are respectively the minimum and maximum possible chances that an actor A associates with proposition K being true. Both s and p range between 0 and 1 and are called support and plausibility respectively, with p always being greater than or equal to s. The value $(p - s)$ is the uncertainty in actor As cognition regarding the verity of proposition K. Thus, the tuple (s, p) can be viewed as a comprehensive representation of As beliefs regarding proposition K and is also referred to as actor As belief state regarding the knowledge proposition K. For example, consider a social network of people in an organization and the proposition of interest to be 'the company image is good'. Each actor has certain beliefs regarding the

companys public image which is reflected in the degree of truthfulness he/she associates with the proposition under consideration. There can be multiple propositions, like each actor in the social network entertains beliefs regarding every other actor being an expert on a topic.

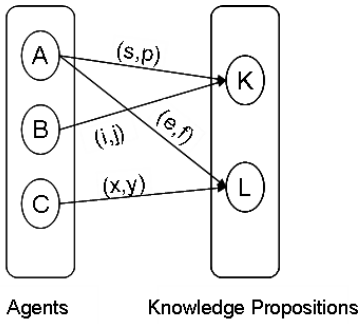


Figure 5.1: Cognitive Knowledge Network as a Bipartite Graph

5.3 Constructing a Cognitive Knowledge Network

Initially all edges in the cognitive knowledge network start with prior labels (0,1). This indicates maximum possible uncertainty i.e. the actor knows nothing about the proposition and believes its truthfulness can range from being absolutely false (support of 0) to being absolutely true (plausibility of 1). As the actor observes emails, based on the contents of these emails, the actor starts refining his/her beliefs. Each email provides evidence 'for' or 'against' the truthfulness of the propositions in cognitive knowledge network. The key step for constructing a cognitive knowledge network is to identify and extract the evidences presented by a given email. Once these evidences are obtained, it is easy to combine them with the existing cognitive knowledge network by updating corresponding actor-proposition edge weights, based on the evidences for the various propositions, as observed by actors. The updates are performed using the evidence combination rule from Dempster Shafer theory. For example, consider an actor A with a initial belief state of (0.8, 0.9) regarding the truthfulness of some knowledge proposition K. Now actor As belief state is to be combined with evidence (extracted from some email received or sent by A) that speaks against the truthfulness of K with a confidence of 0.7. Illustrated in figure 5.2), the edge weights in the cognitive knowledge

network are updated as actors come across evidences "for" or "against" corresponding propositions (edge (A, K) represents actor A's belief state regarding proposition K). The s and p values in actor A's belief state are 0.8 and 0.9 respectively. From these, the basic probability assignment for actor As belief state will be, $m_A(\phi) = 0$, $m_A(`true') = 0.8$, $m_A(`false') = 0.1$ and $m_A(`trueorfalse') = 0.1$. (Note that according to the rules of Dempster-Shafer theory, $m(\phi) = 0$, $m(`true') = s$, $m(`false') = 1-p$ and $m(`trueorfalse') = p - s$) The basic probability assignment for the given evidence is $m_E(\phi) = 0$, $m_E(`true') = 0$, $m_E(`false') = 0.7$ and $m_E(`trueorfalse') = 0.3$. The evidence associates a confidence of 0.7 with K being false. The rest of the probability i.e. 0.3 is uncertain and in such a case K can be true or false. Hence, it is assigned to the 'true or false' probability mass. Using the rule of combination from Dempster-Shafer theory, the updated edge weight in A's knowledge network can be computed by combining his prior belief with the evidence,

$$
\begin{aligned}
m(\phi) &= 0 \\
m('true') &= \alpha\Big[m_A('true')m_E('true') + \\
&\qquad m_A('true')m_E('true\ or\ false') + \\
&\qquad m_A('true\ or\ false')m_A('true')\Big] \\
m('false') &= \alpha\Big[m_A('false')m_E('true') + \\
&\qquad m_A('false')m_E('true\ or\ false') + \\
&\qquad m_A('true\ or\ false')m_E('false')\Big] \\
m('true\ or\ false') &= \alpha\Big[m_A('true\ or\ false') \times \\
&\qquad\quad m_E('true\ or\ false')\Big]
\end{aligned}
$$

where,

$$
\alpha = \frac{1}{1 - \Big[m_A('true')m_E('false') + m_A('false')m_E('true')\Big]}
$$

On solving,

$$m(\phi) = 0$$
$$m('true') = 0.54$$
$$m('false') = 0.39$$
$$m('true\ or\ false') = 0.07$$

The new support $s' = m('true') = 0.54$, and new plausibility $p' = 1 - m('false') = 0.61$ Thus, the updated belief state for agent A is the tuple $(s', p') = (0.54, 0.61)$.

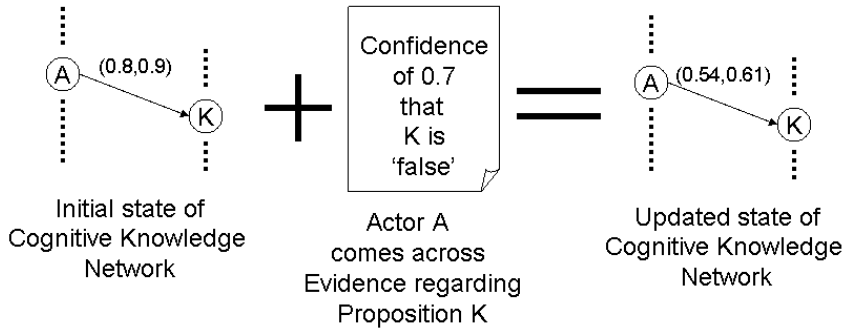


Figure 5.2: Updating beliefs in a cognitive knowledge network

5.4 Evidence Acquisition from Email Text

An important task in constructing cognitive knowledge network is the extraction of evidences from email text. This evidence extraction can be split into two steps, (i) identifying various propositions that a given email talks about, and (ii) extracting the sentiment regarding that proposition in that email i.e. to what degree does the email speak for or against the truthfulness of a given proposition.

The first task in proposed approach is extracting the propositions of interest from the e-mail's text message. [55] shows an interesting approach for automatically identifying semantic features (topics) from emails as well as clustering emails, thus reducing the need for manually read emails. Active research into knowledge proposition extraction is being pursued. However, for preliminary experimental results shown later, we have manually defined the knowledge proposition of interest.

Once the knowledge propositions of interest have been defined, a piece of evidence consists of a Boolean sentiment value indicating whether it speaks for or against (positive or negative) the knowledge proposition and a confidence value or degree (between 0 and 1) for this claim (similar to [56]). Thus, an important task is to extract the senders opinions about knowledge propositions of interest. For this, it is believed that sentiment classification techniques will be a useful tool. Such sentiment analysis using automated techniques has gained interest in computer science research community with the advent of internet. The availability of data about the users opinions/reviews on the Web has triggered the evaluation of several (machine learning, NLP or text mining) approaches for sentiment analysis (see [57], [58], [59] and [60]). [61] illustrates an approach for converting sentiments into a rating measure. However, sentiment analysis remains a difficult task and more research is still remains to be done. [58] have illustrated the difficulty of using bag-of-words machine learning methods for sentiment classification. In addition, sentiment analysis still remains a very domain-specific problem, where classifiers trained for one domain may not perform well in others ([62]). Hence, in present research, the sentiments of users were manually identified for a pre-defined knowledge proposition. A justification for this is that the number of emails, that were relevant/had the knowledge proposition of interest, was small.

5.5 Constructing a Cognitive Knowledge Network

The concepts discussed so far are now combined to present the complete model architecture for constructing and updating the cognitive knowledge network, as knowledge propagates in the underlying social network. As shown in Figure 5.3, the proposed model architecture consists mainly of three components. This model is assumed to be able to observe all email communication, e.g the model resides in the email server. As emails arrive, the evidence acquisition module analyzes the text of each email message and acquires the evidence for each propositions of interest. An email can act as at most one piece of evidence for a given proposition. If an email expresses sentiment about more than a single proposition, then multiple evidences may be extracted, one evidence for each of propositions that the email talks about. The knowledge updating module takes as input all the evidences compiled from an email, the recipients and the sender

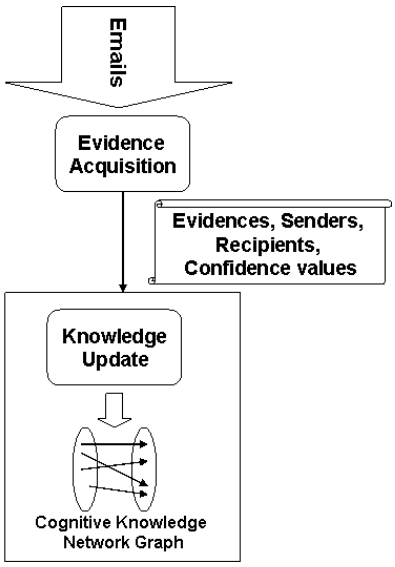


Figure 5.3: Model Architecture

of these evidences (i.e. the actors who 'view'/'come across' these evidences) and their confidence values, and updates the cognitive knowledge network graph (as illustrated previously). As more emails are exchanged in the network, new evidences are gathered from these emails. The cognitive knowledge network graph can then be recorded at regular time intervals for further analysis purposes.

5.6 Experiments on the Enron Email Corpus

Experiments using the proposed sentiment-proposition framework are now presented on the Enron email corpus. The labeled data made available by the Enron Email Analysis Project at the University of California, Berkeley was used
(source `http://bailando.sims.berkeley.edu/enron_email.html`). This data consisted of a subset of all email messages (approximately 1700 labeled emails). These emails were classified into 8 major categories namely, business-related, purely-personal, personal but in a professional context, logistic arrangements, employment arrangements, document editing/checking, empty messages due to missing attachments and empty messages. The email messages in the first category i.e. the business-related category are further classified into sub-categories namely, regulations and regulators, internal

projects, company image (current), company image (changing), political influence/ contributions/contacts, California energy crisis/California politics, internal company policy, internal company operations, alliances/partnerships, legal advice, talking points, meeting minutes, and trip reports.

Table 5.1: Sentiment and confidence values for email evidences.

Knowledge Proposition K: 'The company image is/remains good'		
Category Label	**Sentiment (for/against)**	**Confidence value**
Very Good	For	0.9
Good	For	0.5
Slightly Good	For	0.1
Neutral	NA	NA
Slightly Bad	Against	0.1
Bad	Against	0.5
Very Bad	Against	0.9

Table 5.2: Number of emails in each category.

Category Label	# of emails
Very Good	5
Good	15
Slightly Good	22
Neutral	43
Slightly Bad	15
Bad	15
Very Bad	3
Total = 118	

For current analysis, only one proposition of interest is chosen, i.e. *'The company image is/remains good'*. For this, only those emails falling in two sub-categories, the company image (current) as well as company image (changing), were used. These two sub-categories consisted of a total of 118 emails. The contents (text message) of each of these emails were examined manually and judged for its sentiment regarding the company image. The emails were manually classified into seven categories indicating the impact on the company image namely, very good, good, slightly good, neutral, slightly bad, bad and very bad. With each of these categories, a specific confidence and

for/against sentiment were assigned, as shown in Table 5.1. Table 5.1 shows the number of emails assigned to each category.

It should be noted that emails that display neutral sentiment are discarded and no evidences are extracted from them. Since, there is only one knowledge proposition of interest, only one piece of evidence is extracted per email. The evidences extracted from these emails along with their sender and recipients were fed to the knowledge updating module (in time-ordered way) and the users beliefs were allowed to evolve over time. A total of 118 users were identified to be involved in these email exchanges. The initial s and p value for each user was taken to be 0 and 1 respectively. Due to the small size of the data set, we chose to record the knowledge network graph at the end of every year i.e. 1999, 2000 and 2001. Among the 75 non-neutral emails, the number of emails belonging to the year 1999, 2000 and 2001 were 3, 21 and 51 respectively. The timeline of these emails was from December 1999 to October 2001.

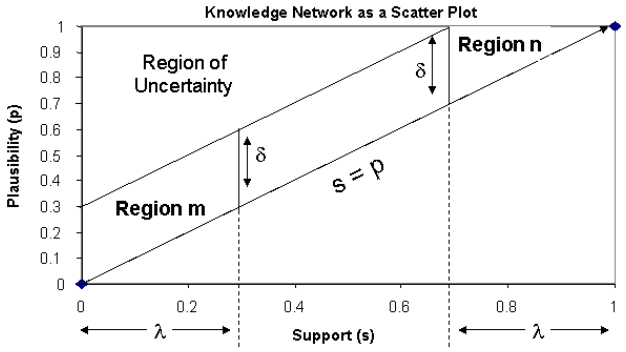


Figure 5.4: Cognitive Knowledge Network as a scatter-plot

A scatter plot was created for the cognitive knowledge network at the end of each year. Actors were plotted in the x-y plane with support s along the x-axis and the plausibility p along the y-axis (Figure 5.4). The line $s = p$ represents points where there is no uncertainty regarding the probability of the verity, of the given knowledge proposition. As one moves farther above this line, the uncertainty in the belief increases (region of uncertainty). The region m near the origin i.e. points which have uncertainty less than some δ and with s value at most some λ, contains points which believe the proposition more likely to be false with low uncertainty. Similarly, the top-right hand region n, consists of points which believe the proposition more likely to be true with

low uncertainty. Since $s \leq p$ always, all points always lie above the line $s = p$.

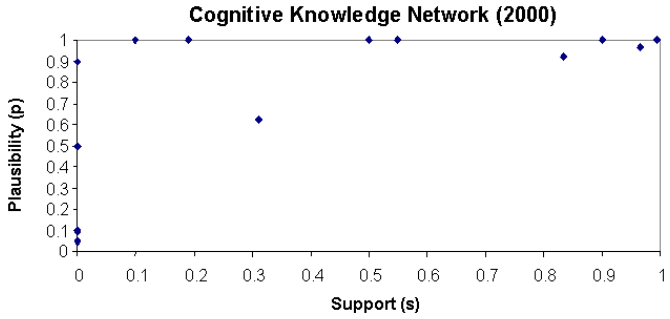


Figure 5.5: Cognitive Knowledge Network plot for the year 2000

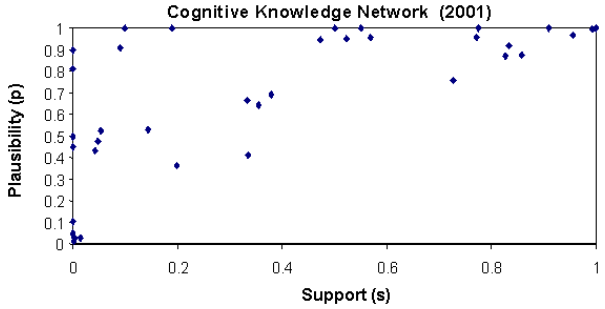


Figure 5.6: Cognitive Knowledge Network plot for the year 2001

The plot for 1999 is not shown as not enough data was available to provide anything interesting. During the year 2000, there were many events such as positive press articles, a highly positive article in time magazine and talk of the most innovative company of the year award, that sought to improve the company image, as well as certain events such as miscommunication within company resulting in bad press and bad press due to environmental and human rights violation in overseas ventures, that tarnished the company image. The plot for 2000 is shown in Figure 5.5. During the year 2001, the company image dropped mainly due to negative press generated during the California power crisis. However, things started improving when there was talk of a possible merger with a rival company (Dynergy). The plot for 2001 is shown in Figure 5.6. The interesting part about the result was the lack of consensus among users regarding the company image. Note the existence of a significant number of points in the regions n and m (see Figures 5.4, 5.5 and 5.6) for both 2000 and 2001. The number of uncertain

people was also quite significant. The fact that the points were pretty much spread out in both the graphs, leading to the inference that there exists a lack of harmony among the users' perceptions regarding company image, throughout both years.

5.7 Conclusions

Knowledge is an important commodity and organizations would like to know where it is and to make it easily accessible to individuals. Thus, knowledge based network analysis is an important field of study in social sciences. In this research an evidence-theory based methodology is presented for constructing and maintaining a cognitive knowledge network in an electronic communication environment. Experimental results using the proposed approach on a subset of the Enron email data show how employee sentiments can be estimated, allowing to identify different 'camps' of ideas as well as their popularities. The proposed approach has various applications in an organizational environment. It can be used to ensure consistent and correct knowledge distribution as well as resolve misperceptions among users. Other uses include, analyzing the evolution of cognitive knowledge statements. Such analysis can help identify sources of information as well as the rate/extent of information exchange.

Chapter 6

Data Driven Influence Value Computation

6.1 Introduction

The process of information exchange allows actors to share their sentiments, opinions and knowledge regarding any topic of interest. One consequence of this is the influence actors' have on each others behavior. Certain individuals in the network are "influential" in that many people see them and get ideas from them. However, this value is difficult to assess. Individuals can also influence others in powerful ways. Friends, enemies, relatives, and co-workers, for example, can influence each other to buy cars, quit smoking, eat certain foods, etc. In any group of people, however, some are likely to be more influential than others. This influence may work on everyone in a group, or on only one or a few members of the group. Again, however, it can be difficult to measure these influences. Knowing the value of each persons influence could be immensely valuable. For example, it could enable preferential treatment of those users whose influence brings significant value to the network.

This chapter presents a framework for estimating the value of influence of actors in a given social network. The desired outcome is a value for the influence each person exercises over others. The value is estimated using data on actors' interactions and activities. With knowledge of these values, a marketer, community organizer or advertiser can discover who is influential and in what situations, and how much that

influence is worth. This may allow them to treat and interact with these people to improve the system, to create more value, to increase participation, and further other social outcomes.

6.2 Background

A social network of actors (denoted as $G(V, E)$) generally has a context around it. This context is also responsible for the actors coming together to form the network in the first place. Data collected from such social environments provide information on this context along with the extent of activities in which actors can participate. For example, in multiplayer co-op video games, actors may undertake quests (solo and as a group), farm items, kill monsters and/or other actors' avatars, form and/or join guilds, mentor other actors, trade, chat, etc. Many of these activities influence, directly or indirectly, the actions of every actor in a variety of ways that are dictated by the nature of the environment, the actors themselves, and the activities. In case of a commercial environment, activities performed by various users, directly or indirectly, also impact the revenue generated from these systems.

People influence each other and while the actual influence itself is difficult to measure, it is possible to quantify the resulting change in actors' behaviors. The actions of every participant can have a positive or negative impact on others in their local network or "neighborhood." Thus, while actors are participating and performing activities in any social system, they are also impacting (and being impacted by) the actions of other actors in that system. While influence by itself is a very broad notion, for pragmatic purposes, the "social influence value" for a given actor, is defined as the quantification of the impact that actor has, for a well defined set of behaviors of interest, on other actors in the network. Examples of such behaviors of interest include the amount of time a user spends in a system, satisfaction, response rate to advertisements served to the user, money spent, etc.

Figure 6.1 illustrates an example of valuations of influence that users can exercise on others. User u_c, for example, has an unusually large value associated with him/herself. This is because, not only is user u_c providing direct revenue, but also influencing his/her neighbors. This influence can be a large part of the value of a users presence. User u_cs

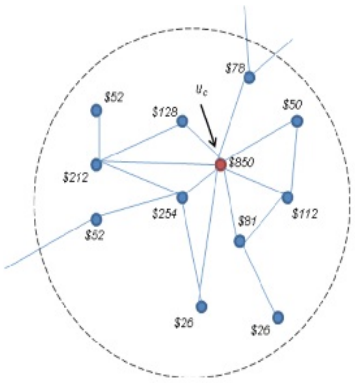


Figure 6.1: Users, in a social network, affecting other users around them.

influence on others in the network is included in his/her valuation. As a result, the actual value of user u_c may be a lot more than indicated by the direct revenue generated from him/her. Influence value estimation captures and quantifies this effect for each user in the network.

Traditionally actor to actor influence in a social network has been looked at from a viral marketing perspective ([13], [16], [63]). The problem studied in this literature is that of spread maximization. The objective is to pick out that set of k actors (k is a user parameter) which will maximize the spread/awareness of some commodity by being early adopters and promoting it via word of mouth. This "word of mouth" transfer of awareness, formally refered to as a cascading process with the cascade being the commodity propagating in the network, is modeled and the desired set of nodes is discovered. Other literature in this area has been dedicated to looking at different variations on the classic problem setting as well as more optimal methods for simulating cascade propagation ([64],[65]). However, in all the above, the focus is primarily on the cascading process itself and the methods assume that some measure of pairwise actor-actor influence are available to use. The research presented in this chapter is complementary to these approaches in that the influence value scores computed can be used as input to these propagation algorithms and network value scores after simulating influence propagation can be computed.

Learning problems regarding influence have only recently been looked at. One such work estimates the structure of the network itself by studying log traces of commodity

propagation ([66]). The core idea is to estimate the most likely structure of the network that best explains the spread of the commodity. In ([67]) influence propagation is modeled using a Bayesian generative process and most likely parameter values, regarding the spread of the cascades and receptivity of actors to those cascades, that best explain the spread observed in given data logs. A technique using data logs to learn influence probabilities based upon occurences of influenceee actors repeating actions of influencers is presented in ([68]). The above methods are similar in that they learn parameters related to the cascading process. The main problem with such approaches is the assumption of this specific setting having cascades propagating in the network. The assumption restricts the definition of influence. Influence need to be a result of actors setting a precedent and others adopting it. It can also come in other forms such as - actors participating in mutual activities develop influence over each other, heirarchies defined by the system can also result in influence (example, a forum moderator holds some power and can have influence over the common posters), etc. This research presents a general framework that addresses thise limitation and allows the practitioner to have considerable control over how influence can be captured, thus improving applicability.

Other closely related work includes Tang et al. [47] in which a topic modeling based approach for estimating topic specific influence values is discussed. The technique can based to identify topic specific authorities in the network. While this work has a similar general idea of using the data to estimate influence values, however, the problem as well as the solution is specific to the setting discovering topic authorities in cases where textual data associated with actors is available.

6.3 Data Driven Influence Value Computation

Consider a social network of users $G(V, E)$. It is assumed that data on each user as well as his/her actions is available in the form of user logs. Profile information on each user, if available, may also be included. The first step is to identify a behavior of interest. These are typically behaviors that directly impact the value of the community. For example, in an online game, the bahvior of interest can be how much time users spend playing the game. The more time users play, the more is the revenue generated, the more longevity the game has and the greater are the chances of the player base

increaasing.

Once such a behavior of interest and corresponding value has been defined, the next step is to construct a data mining prediction model. The data on activities ($\{A_i\}_{u_i \in V} \in D$), profile($\{P_i\}_{u_i \in V} \in D$), environment ($Env \in D$) as well as any other suitable source is used to learn a function predicting the expected behavior of interest for every user i.e. $v_k = f(D, G, u_k)$, $\forall u_k in V$. An example of such a function can be a model predicting a vector of probabilities of users responding positively to a set of advertisements.

In order to measure the influence of a given user u_a on another user u_b, the expected behavior of user u_b is first computed as $v_b = f(D, G, u_b)$. The removal of user u_a is then be simulated by removing all data related to user u_a from D and re-computing the expected behavior of u_b, $v_b^{V-\{u_a\}} = f(D - \{u_a\}, G(V - \{u_a\}, E - \{e(u_a, n_i)\}), u_b)$, $\forall n_i \in N(u_a)$, where $N(u_i)$ denotes the set of neighbors of user u_i. The distance between the expected behaviors of user u_b, with user u_a being present and absent from the data, is taken as the measure of user u_as influence on user u_b i.e. $Inf(u_a, u_b) = d(v_b, v_b^{V-\{u_a\}}) \geq 0$. User u_as influence on every other user can be similarly estimated. The valuations of all of these pair-wise influences are then aggregated to obtain an overall value of influence for user u_a as $Inf(u_a) = Aggr(\{Inf(u_a, n_i)\}_{n_i \in N(u_a)})$. Thus, a measure is provided of how much user u_as presence, reflected in the data, impact the activities and consequently the value generated from other users. The core idea behind the approach being that a given actor's influence can be estimated by measuring the expected change in the bahvior of other actors when the given actor is present and absent in the social environment. Figures 6.2 and 6.3 illustrate the proposed appraoch. First, data on user activities, user characteristics, the environment and the user network is used to learn a function that can predict values for the behavior of interest. Next, the learnt function is used to estimate change in value of user u_a's neighbors when he/she is removed from the network. These changes in value are defined as the respective influence u_a has on his/her neighbors. Influence values on every neighbor can be aggregated to obtain an overall influence score for u_a.

The key question then is how does one aggregate pair-wise influence scores into an overall score for a given user. Note that the influence of neighbors over an actor's behavior is not additive i.e. if neighbor n_1 influences a change of 20 mins of playtime and neighbor n_2 influences a change of 10 mins of playtime, then the combined effect

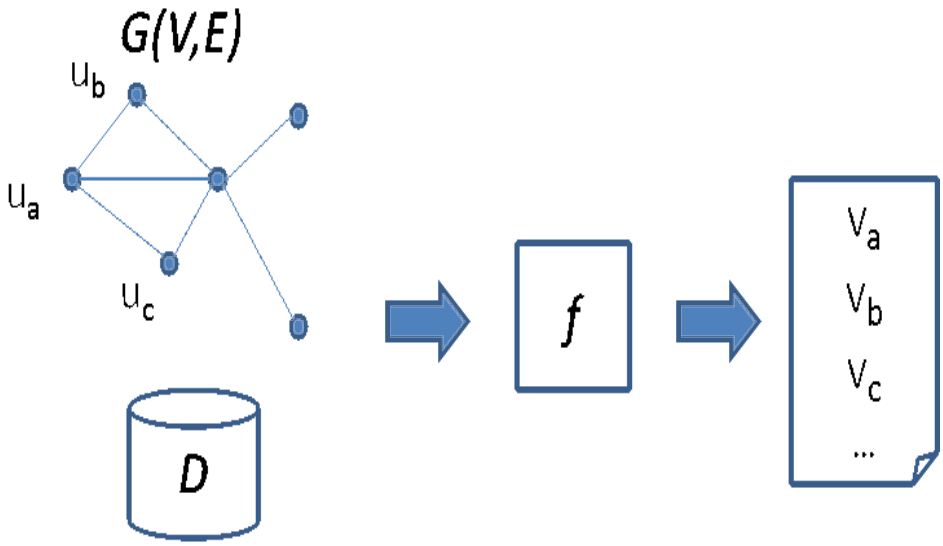


Figure 6.2: Model behavior of interest from data

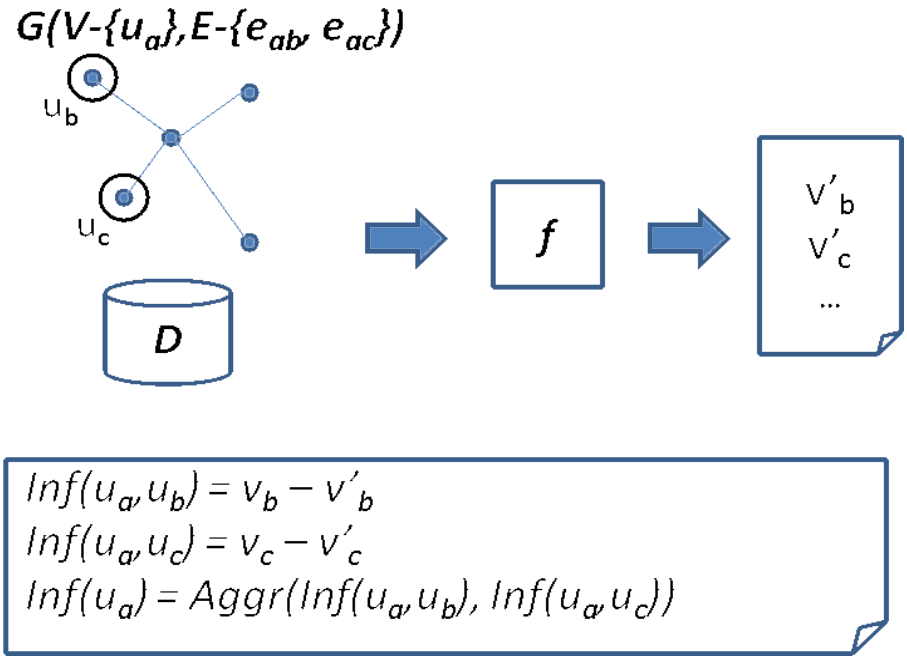


Figure 6.3: Use the behavior of interest model to compute influence

of neighbors n_1 and n_2 need not be the sum of these quantities. This is because the effect of influence is not independent. This issue is much more clearly illustrated in cases where influence can be positive and negative. For example, neighbor n_3 causes user u_a's playtime to reduce by 15 mins. It is nontrivial to combine the effects of n_1, n_2 and n_3. In the proposed approach, influence is defined as a non-negative distance between behaviors and thus, restricts the definition of influence to exclude any kind with sentiment. In the case of restricting influence to be non-negative, the problem now is that user u_a can still credit neighbor n_1 and n_2 with values of 20 and 10 mins respectively, however, the combined effect of removing neighbors n_1 and n_2 will most likely be lower than 30 mins of value. Thus, neighbors n_1 and n_2 are given more credit then they are actually due for their influence on user u_a. An alternative would be to account for the joint effect by removing both neighbors n_1 and n_2 and then measuring change in user u_a's behavior. However, accounting for all possible combinations of neighbor sets is computationally infeasible. Therefore, some approximation is required to account for these joint effects.

The key issues arises from the desire for credit handed out by an actor to his/her influencers being consistent with the joint effect of the neighborhood. For each user u_i, the expected behavior v_i is a result of two factors - (1) nature inherent to user u_i, independent of any social effects, (2) the influence of other users around u_i. The total influence value that u_i credits to $N(u_i)$ must be the same as that corresponding to the net influence of $N(u_i)$. This value is defined as the influence credit for user u_i and is denoted as $Ic(u_i)$. It is important for $Ic(u_i)$ to be consistent with the net effect of the neighbor-hood as this ensures that no artificial influence credit is handed out.

The main issue is that $\sum_{\forall n_j \in N(u_i)} Inf(n_j, u_i)$ is not equal to $Ic(u_i)$. In order to ensure that the credit handed out matches the influence from neighbors, it is proposed to scale $Inf(n_j, u_i)$ with an appropriate factor. In case of joint influence from a set of neighbors, the influence credit is allocated to these neighbors in the same proportion as their respective marginal influences (i.e. $Inf(n_j, u_i)$).

Once $Ic(u_i)$ is computed, the scaling factor can be trivially obtained from it as $\alpha(u_i) = Ic(u_i) / \sum_{n_j \in N(u_i)} Inf(n_j, u_i)$. The question now is how does one compute the value $Ic(u_i)$? The combined effect of the neighborhood influence is computed in a manner similar to that of computing influence for a single neighbor. All data on user

u_i's neighbors, $N(u_i)$, is excluded and the model f is used to estimate a value for the case when user u_i is "socially isolated." By removing all edges from G, such a condition can be simulated for every node and one can compute $v_i^I = f(D - \{N(u_i)\}, G(V - \{N(u_i)\}, E - \{\nu(N(u_i))\}, u_i), \forall u_i \in V$, where $\nu(S)$ is the set of edges incident on atleast one node in set $S \subset V$. The combined influence value of the neighborhood can then be computed as $Ic(u_i) = d(v_i, v_i^I)$. Since, it is computationally infeasible to account for all combinations of neighbors when accounting for joint influence effects. Therefore, the influence value to be credited to individual neighbors is approximated and the total credit to be received by the neighborhood, $Ic(u_i)$ is divided among the neighbors in the same proportions as their marginal influence $\{Inf(n_j, u_i)\}_{\forall n_j \in N(u_i)}$. Thus, the influence value to be credited to some $n_j \in N(u_i)$, is a scaled version of $Inf(n_j, u_i)$. This new "Scaled Influence Value" is defined as $ScInf(n_j, u_i) = \alpha(u_i) Inf(n_j, u_i)$, where scaling factor $\alpha(u_i) = Ic(u_i) / \sum_{n_j \in N(u_i)} Inf(n_j, u_i)$. The overall influence score for any given user can now be computed by simply summing up the scaled infleunce values for every neighbor, $Inf(u_i) = \sum_{n_j \in N(u_i)} ScInf(u_i, n_j)$. The proposed approach assumes that the marginal influence effects of neighbors is representative of their net effect, hence, the limitation on restricting the approach to unsigned influence.

If $Inf(u_i) = \sum_{n_j \in N(u_i)} ScInf(u_i, n_j)$ then, sum of influence values of all nodes $\sum u_k \in V Inf(u_k) =$ total all influence credit handed out by influencees $\sum u_k \in V Ic(u_i)$. According to $Inf(u_i) = \sum_{n_j \in N(u_i)} ScInf(u_i, n_j)$, any given node u_i hands out exactly $Ic(u_i)$ value to its influencers. Thus, the value $\sum_{u_i \in V} Ic(u_i)$ is simply redistributed, by the influencees, among the influencers. The total influence value $\sum_{u_i \in V} Inf(u_i) = \sum_{u_i \in V} Ic(u_i)$.
Thus, the the total influence value being attributed to users, is equal to the total value from the net effects of local neighborhoods and no artificial value is created or destroyed during the process.

The proposed approach is a general framework that can apply to a wide variety of social systems. The key requirements are the availability of data on user interactions as well as their activities in a given social environment. The nature of the data mining model f is domain dependent and will differ for different scenarios. However, it is important that the model predict some behavior of interest, which can be captured by the data. The model should also account for any social effects which impact the users'

behavior.

Influence by itself is a broad term and in most practical scenarios it is generally represented by some bottomline metric. Thus, the definition of influence itself can vary from application to application and it is dificult to have a specific method which can cover a wide variety of cases. The proposed approach is flexible in that it allows the practitioner to specify the complexity and manner with which influence affects behavior is modeled by adjusting f. Any nuances specific to the problem can be accounted for and the corresponding influence value scores computed. Algorithm 3 summarizes the proposed approach and presents the required steps for computing influence values scores for all actors in a given social environment with a well defined behavior(s) of interest with a corresponding value, model f and data on their interactions, activities, the environment etc. The algorithm is of $O(|E| * C(f))$ where $C(f)$ is the time cost of adjusting the dataset to remove a given set of users and using f to compute a value on the altered data.

6.4 Computing Network Value from Influence Value Scores

In the previous section, the influence value score computed for a user, measures the influence of that user on its neighborhood only. A given user's interactions are limited to his/her neighbors and those are the people he/she directly affects. However, these neighbors can influence their neighbors and the initial user commands some indirect influence over its neighbors' neighbors. Thus, influence can cascade to larger portions of the network. The network value of a node is a measure of such aggregate influence, accounting for the direct as well as indirect influences. Network Value computation techniques generally focus on the propagation aspect and assume that network edges are representative of the influence any given user has over its neighbors. These approaches can be used in conjunction with the proposed approach of influence value computation.

A network value propagation algorithm which takes some network as input can be applied on $G(V, E')$, where if $e(x, y) \in E$ then probabilities of influence for both ordered pairs $p(x, y) = Inf(x, y)/v_y, p(y, x) = Inf(y, x)/v_x \in E'$ (where $p(x, y)$ denotes probability of x influencing y), to propagate the pairwise influence value scores into a network

Algorithm 3 computeDataDrivenInfluenceValues($G(V,E)$,D,f)

Inputs: Graph $G(V,E)$ with non-negative edge-weights, Data on user behavior, characteristics as well as the environment (D), behavior value model f, behavior distance function d
Output: $PairWiseInf = \{ScInf(u_i,u_j), ScInf(u_j,u_i) | \forall e(u_i,u_j) \in E\}$, UserInf = $\{Inf(u_i) | \forall u_i \in V\}$
BEGIN
$Icvals(1 \dots |V|) = 0$
$PairWiseInf(1 \dots |V|, 1 \dots |V|) = 0$
$UserInf(1 \dots |V|) = 0$
for each $u_i \in V$ **do**
  InvScaleFactor = 0
  for each $n_j \in N(u_i)$ **do**
    $PairWiseInf(n_j,u_i) = d(v_i, v_i^{V-\{n_j\}})$, where $v_i = f(D,G,u_i)$ and $v_i^{V-\{n_j\}} = f(D-n_j, G(V-n_j, E-\{\nu(n_j)\}), u_i)$
    InvScaleFactor = InvScaleFactor + $PairWiseInf(n_j,u_i)$
  end for
  $Icvals(u_i) = d(v_i, v_i^I)$, where $v_i^I = f(D-\{N(u_i)\}, G(V-\{N(u_i)\}, E-\{\nu(N(u_i))\}), u_i)$
  InvScaleFactor = InvScaleFactor/$Icvals(u_i)$
  for each $n_j \in N(u_i)$ **do**
    $PairWiseInf(n_j,u_i)$ =InvScalingFactor $PairWiseInf(n_j,u_i)$
  end for
end for
for each $u_i \in V$ **do**
  $UserInf(u_i) = \sum_{\forall n_j \in N(u_i)} PairWiseInf(u_i,n_j)$
end for
return $PairWiseInf, UserInf$

value for each user. The edge weights in E' are representative of independent probabilities of one neighbor influenceing another and fit well in the well studied independent cascade model framework.

Traditional network value computation methods assume the availability of these pair-wise influence values and the transformed network, $G(V,E')$, is much more representative of the influence users have over their neighbors. This now allows the practitioner to leverage state of the art techniques for computing network values, based upon the data driven influence values, using cascade simulation. Algorithm 4 presents an algorithm for the same. Good candidate algortihms for computing network value using

$G(V, E')$ are discussed in [16]. Optimizations of these techniques are also applicable ([65]).

Algorithm 4 computeDataDrivenNetworkValues($G(V, E)$,D,f)

Inputs: Graph $G(V, E)$ with non-negative edge-weights, Data on user behavior, characteristics as well as the environment (D), behavior value model f, behavior distance function d
Output: Network Values $\forall u_i \in V$ using data driven influence values
BEGIN
$vals^N(1 \dots |V|) = 0$
$PairWiseInf(1 \dots |V|, 1 \dots |V|) = 0$
for each $u_i \in V$ **do**
  for each $n_j \in N(u_i)$ **do**
    $PairWiseInf(n_j, u_i) = d(v_i, v_i^{V-\{n_j\}})$, where $v_i = f(D, G, u_i)$ and $v_i^{V-\{n_j\}} = f(D - n_j, G(V - n_j, E - \{\nu(n_j)\}), u_i)$
  end for
  for each $n_j \in N(u_i)$ **do**
    $PairWiseInf(n_j, u_i) = PairWiseInf(n_j, u_i)/v_i$
  end for
end for
$NetworkValues$ = computeNetworkValue($G(V, E)$,$PairWiseInf$)
return $NetworkValues$

6.5 Experiments

Experiments and their results for data driven network value computation are presented on two real life datasets - (i) User interaction and game play logs from the popular Massively Multi-player Online Role-Playing Game (MMORPG) Everquest 2 (EQ2) (`http://www.everquest2.com/`), and (ii) Co-authorship data from the computer science DBLP network `http://kdl.cs.umass.edu/data/dblp/dblp-info.html`.

6.5.1 MMORPG User Grouping Data

Experiment Setup

Experiments are presented by applying the proposed approach on user interactions observed in a Massively Multiplayer Online Role Playing Game (MMORPG). MMORPGs

provide users with a persistent fantastical virtual world, in which they can log in with a virtual character (referred to as the user's "avatar"). Users can then complete in game quests or objectives, designed by the game developers. As users complete these in game tasks, their avatars become stronger. Along with becoming stronger, user avatars also gain gold and other in game items which can be used to improve the avatars' appearance and capabilities. An avatar's appearance reflects its strength and can be seen by everyone else in the game. Thus, improving avatar strength/appearance is the main incentive for users to accomplish in game objectives. An avatar's strength is typically represented by an integer value level ranging from zero to some positive number, generally around fifty to seventy, referred to as the avatar/character level. At any time there can be thousands of users logged in and most characters populating a MMORPG world are avatars controlled by humans. Consequently, socialization is a very important part of experiencing any MMORPG. Users form groups with each other in order to collectively complete difficult tasks such as killing boss monsters, crafting expensive items etc. Other forms of socialization such as trade, mentoring other users, chat etc are also available.

For the experiments, the data set consists of variables describing user session, achievement as well as social behavior for a three month time period. The variables used are average session length, number of quests completed, number of groups participated in, average inter session length, number of unique users interacted with, number of avatars owned by every user (users can have up to six avatars), average level of avatars for every user, number of levels advanced in the last three months, rate of completing quests w.r.t. time and rate of joining groups w.r.t. time. Along with these user specific variables, the grouping network, describing who formed how many groups with whom, is also available. The data consisted of only those users who had participated in atleast one group during the three month period and there were $53,836$ such users with $5,521,156$ edges between them. Finally, information on whether each of these users churned or not within a few months, is also available. The number of churners was about 12.22% of the users in the dataset.

For modeling influence, player time to live is designated as the behavior of interest and influence was measured in terms of how much change users induce in their neighbors' time to live. Note however, that time to live can be increased or decreased and thus

influence can be positive or negative. In order to get around this issue, the proposed approach is first used to compute influence in terms of the magnitude of change in behavior, and then based upon whether the time to live values are decreasing or increasing, the corresponding pairwise influence values are made negative or positive respectively. These signed pairwise values are then used to compute signed influence values by simply adding the signed influence credit each user gets from his/her neighbors.

In Figure **??** the various aspects of the proposed influence computation method are presented w.r.t the MMORPG environment. For given user u_i, time to live is defined as ($T_i = 1/p_i$, where p_i is churn probability of user u_i). A data mining model f, using the user specific variables with churn label being the target variable, was built using *Weka* [69]. The model had f-measures of 0.946 and 0.575 on churners and non-churners respectively. The precision recall for the churner class was 0.627 and 0.531 respectively. For a given user, the churn $p_i = f(u_i)$ is first computed. The removal of some neighbor, n_j, is simulated by adjusting u_i's social variables (number of groups participated in, number of users interacted with and rate of joining groups) to remove n_j's interactions from them to get an adjusted record $u_i^{-n_j}$. The churn probability $p_i^{-n_j} = f(u_i^{-n_j})$ and time to live $T_i^{-n_j} = 1/p_i^{-n_j}$ are representative of expected behavior from u_i, if n_j was absent from the network. The unscaled influence of n_j on u_i is defined as $Inf(n_j, u_i) = |T_i - T_i^{-n_j}|$. Similarly, the influence credit handed out by a node, $Ic(u_i)$ is defined as $|T_i - T_{i-N(u_i)}|$ i.e. the change in time to live for u_i with his/her neighbors being present and absent from the network respectively. Defining $Ic(u_i)$, d and $Inf(n_j, u_i)$ is sufficient for computing influence values. One important point to note is that the change in time to live can be directional or signed i.e. it can be positive (time to live increases) or negative (time to live decreases, possibly due to griefing, trolling etc.). In order to account for this, each influencee credit value was also given a positive or negative sign based upon the change in time to live being positive or negative respectively.

Results and Observations

Figure 6.4 displays influence values sorted from lowest to highest. From the sharp drops and rises at the ends of the plot one can infer that there are very few nodes having significant influence. This is expected as in most social networks there are few nodes

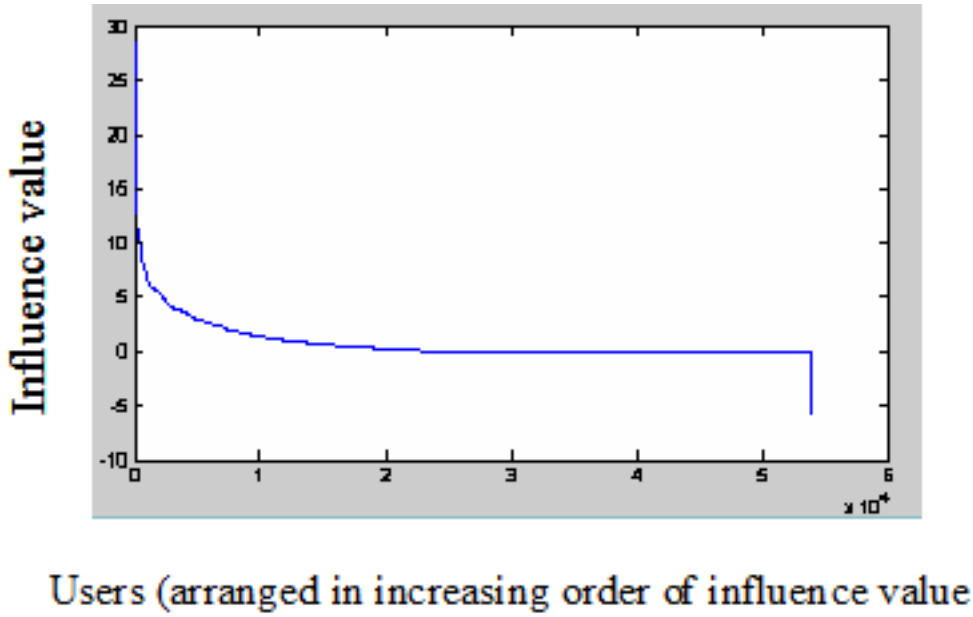


Figure 6.4: Influence values in the MMORPG network

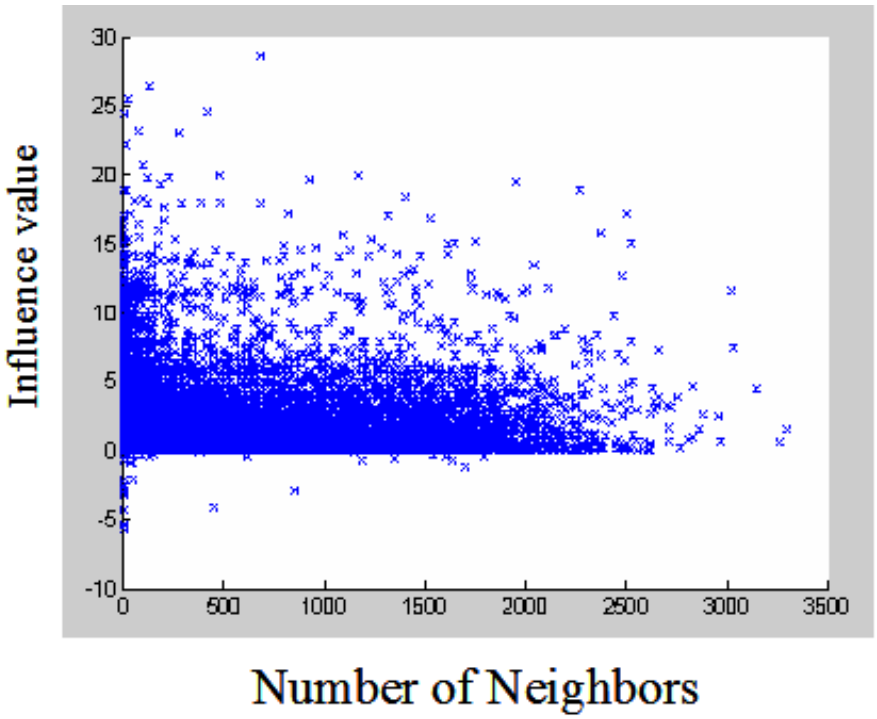


Figure 6.5: Number of neighbors compared to influence value

of prominence that can command strong influence and the larger majority tend to have comparatively much lower values for the same. Figure 6.5 shows a plot of number of neighbors versus influence value. The plot shows that influence values are largely unaffected by the number of neighbors. It was observed that users are most likely to have one time groups with most of their neighbors, however, some users had a smaller set of neighbors with which they had much more intense and frequent interaction. The more such focused interactions a user has, the higher will be his/her's influence.

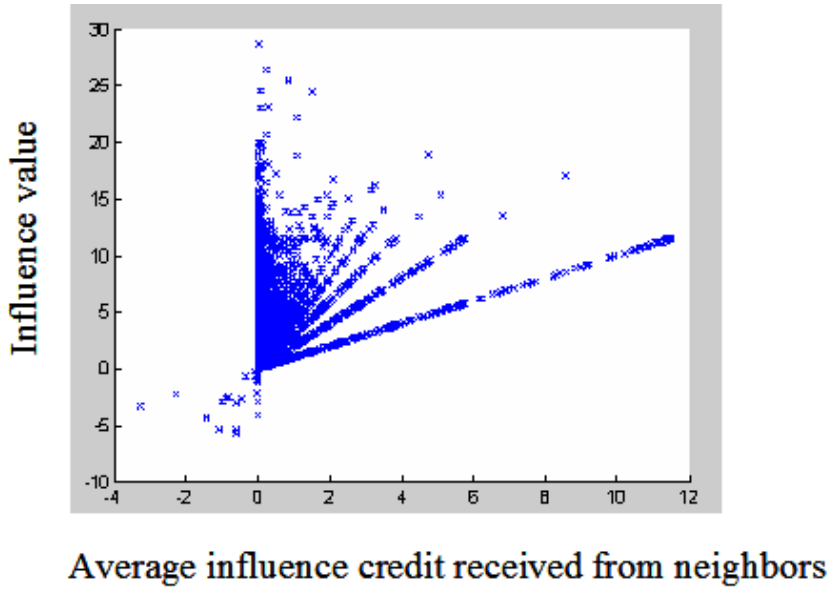


Figure 6.6: Average influence credit from neighbors vs. influence value

Figure 6.6 and figure 6.7 reveal something interesting about the network. Figure 6.6 plots the mean influence credit received from a neighbor versus influence value, and shows that user influence increases w.r.t average credit from a neighbor. In figure 6.6 which plots standard deviation of influence credit from neighbors versus influence value, it an be seen that users with higher standard deviation in credit from neighbors, tend to exercise higher (positive or negative) influence over their neighbors. From this observation, one can infer that influence in the network is typically of the form where users preferentially treat a select few of their neighbors. In figure 6.6 even in cases where users have low average influence credit from a neighbor, it is still more likely the case that those users have strong interactions with a select few of their neighbors and very

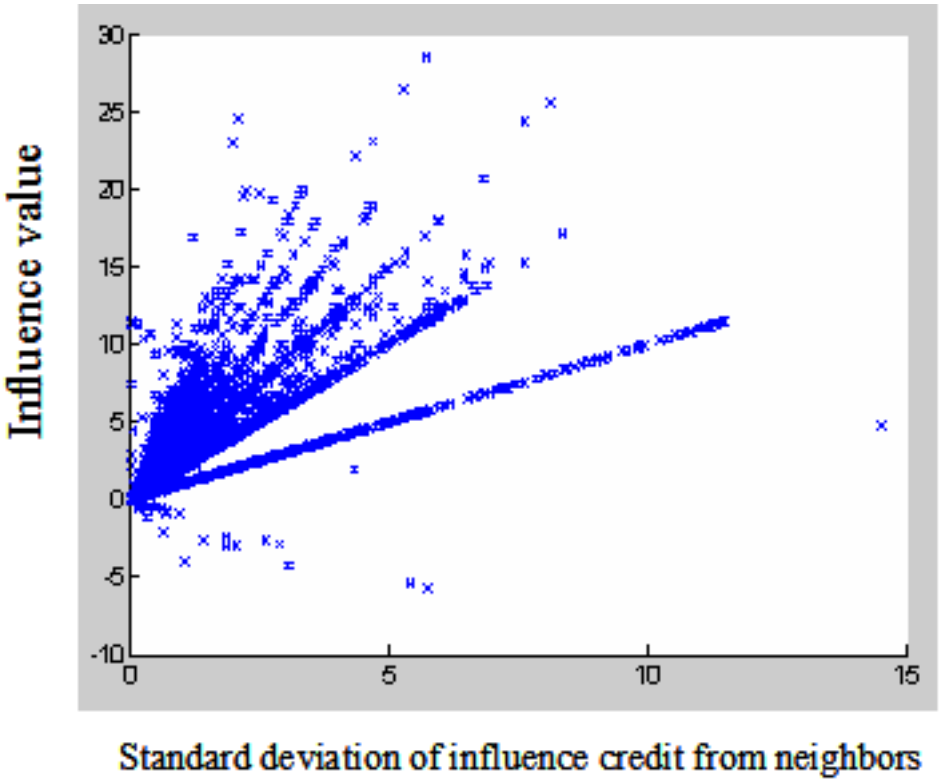


Figure 6.7: Standard deviation of influence credit from neighbors vs. influence value

low (most cases just a single grouping instance) interactions with the rest. This makes sense as grouping with users is a significant investment in terms of time as well as effort. Since, every individual has a social bandwidth, it is expected for users' influence to be focussed over fewer neighbors.

6.5.2 DBLP Co-authorship data

Experiment Setup

The DBLP dataset has information on cs publications listed in the DBLP Computer Science Bibliography `http://kdl.cs.umass.edu/data/dblp/dblp-info.html`. This is derived from a snapshot of the bibliography as of April 12, 2006 and consists of $456,002$ authors having $1,292,397$ co-authorship links between them. Influence is defined as, how much shift in topic affinities does an influencer causes in the topics of publications of the influencee. Similar to the MMORPG exepriment, Figure **??** describes the various aspects of the proposed influence computation method in the DBLP dataset. It presents definitions for $Ic(u_i)$, d and $Inf(n_j, u_i)$, sufficient for computing influence values.

Results and Observations

Figure 6.8 presents the top 20 influencers in the DBLP dataset. Almost all of the candidates are well known computer scientists in their respective areas. Some of the entries are due to data errors in name disambiguation C. Chang (#1), Wei Li (#19), Wen Gao (#11). It was observed that the top influencers have multiple long time faithful collaborators. Figure 6.9 shows the DBLP page for one such collaborator, who imparts high influence credit to, one of the top influencers, Philip Yu.

Figure 6.10 presents the DBLP data influence values from lowest to highest. It is observed that few authors have very high influence, while the lager majority has relatively much lower influence scores.

Figures 6.11 and 6.12 show that influence score increases with number of neighbors as well as weighted degree. Co-authorship is a social relationship that requires significant investment to initiate and the investment continues to remain significant for further collaboration. Therefore, it is expected that the network topology is a reasonable indicator of influence. The more an author publishes with more collaborators, the

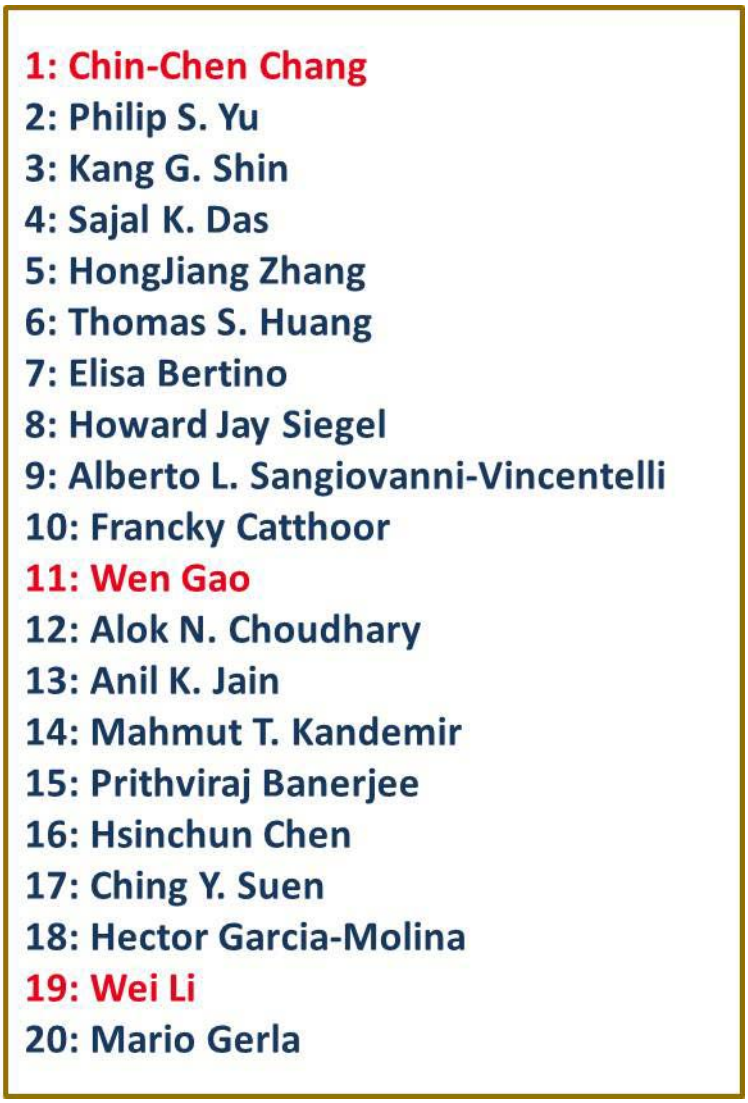
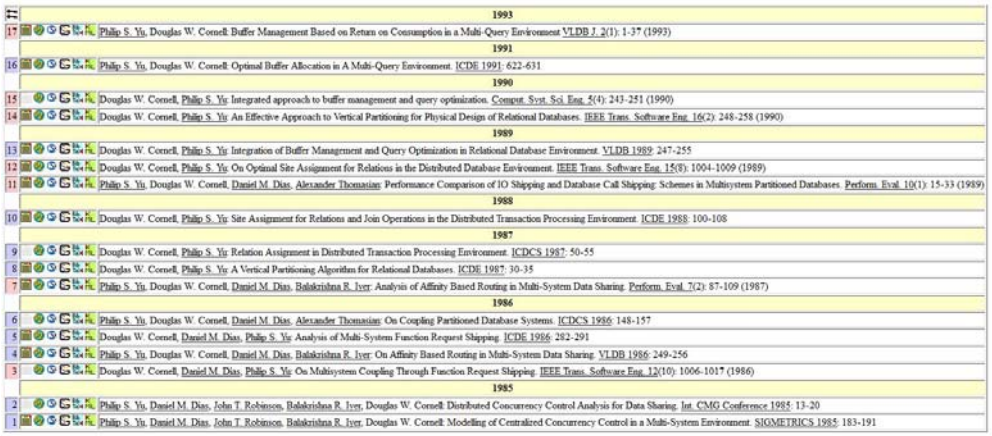


Figure 6.8: Top 20 influencers in the DBLP dataset



#	
	1993
17	Philip S. Yu, Douglas W. Cornell: Buffer Management Based on Return on Consumption in a Multi-Query Environment. VLDB J. 2(1): 1-37 (1993)
	1991
16	Philip S. Yu, Douglas W. Cornell: Optimal Buffer Allocation in A Multi-Query Environment. ICDE 1991: 622-631
	1990
15	Douglas W. Cornell, Philip S. Yu: Integrated approach to buffer management and query optimization. Comput. Syst. Sci. Eng. 5(4): 243-251 (1990)
14	Douglas W. Cornell, Philip S. Yu: An Effective Approach to Vertical Partitioning for Physical Design of Relational Databases. IEEE Trans. Software Eng. 16(2): 248-258 (1990)
	1989
13	Douglas W. Cornell, Philip S. Yu: Integration of Buffer Management and Query Optimization in Relational Database Environment. VLDB 1989: 247-255
12	Douglas W. Cornell, Philip S. Yu: On Optimal Site Assignment for Relations in the Distributed Database Environment. IEEE Trans. Software Eng. 15(8): 1004-1009 (1989)
11	Philip S. Yu, Douglas W. Cornell, Daniel M. Dias, Alexander Thomasian: Performance Comparison of IO Shipping and Database Call Shipping Schemes in Multisystem Partitioned Databases. Perform. Eval. 10(1): 15-33 (1989)
	1988
10	Douglas W. Cornell, Philip S. Yu: Site Assignment for Relations and Join Operations in the Distributed Transaction Processing Environment. ICDE 1988: 100-108
	1987
9	Douglas W. Cornell, Philip S. Yu: Relation Assignment in Distributed Transaction Processing Environment. ICDCS 1987: 50-55
8	Douglas W. Cornell, Philip S. Yu: A Vertical Partitioning Algorithm for Relational Databases. ICDE 1987: 30-35
7	Philip S. Yu, Douglas W. Cornell, Daniel M. Dias, Balakrishna R. Iyer: Analysis of Affinity Based Routing in Multi-System Data Sharing. Perform. Eval. 7(2): 87-109 (1987)
	1986
6	Philip S. Yu, Douglas W. Cornell, Daniel M. Dias, Alexander Thomasian: On Coupling Partitioned Database Systems. ICDCS 1986: 148-157
5	Douglas W. Cornell, Daniel M. Dias, Philip S. Yu: Analysis of Multi-System Function Request Shipping. ICDE 1986: 282-291
4	Philip S. Yu, Douglas W. Cornell, Daniel M. Dias, Balakrishna R. Iyer: On Affinity Based Routing in Multi-System Data Sharing. VLDB 1986: 249-256
3	Douglas W. Cornell, Daniel M. Dias, Philip S. Yu: On Multisystem Coupling Through Function Request Shipping. IEEE Trans. Software Eng. 12(10): 1006-1017 (1986)
	1985
2	Philip S. Yu, Daniel M. Dias, John T. Robinson, Balakrishna R. Iyer, Douglas W. Cornell: Distributed Concurrency Control Analysis for Data Sharing. Int. CMG Conference 1985: 13-20
1	Philip S. Yu, Daniel M. Dias, John T. Robinson, Balakrishna R. Iyer, Douglas W. Cornell: Modelling of Centralized Concurrency Control in a Multi-System Environment. SIGMETRICS 1985: 183-191

Figure 6.9: DBLP page of a top influencer's collaborator

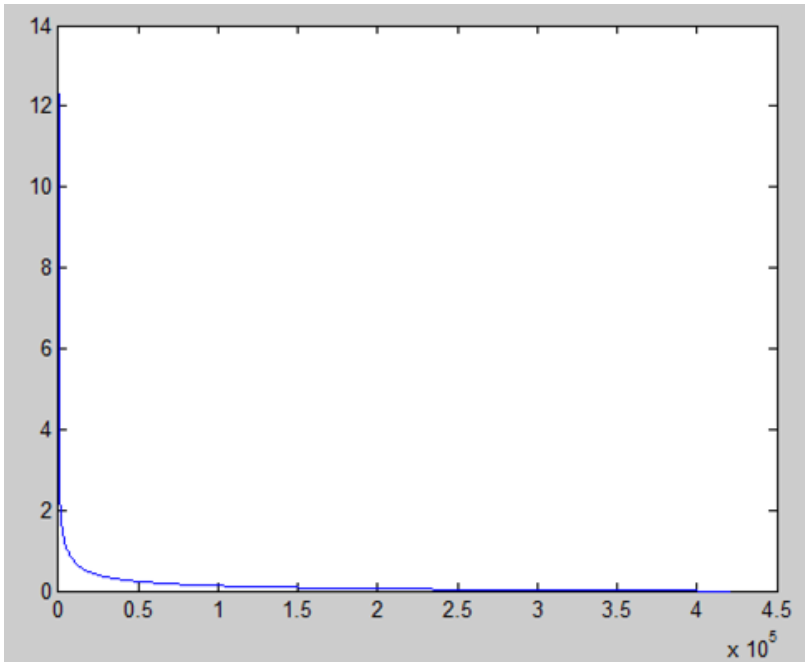


Figure 6.10: DBLP influence values from lowest to highest

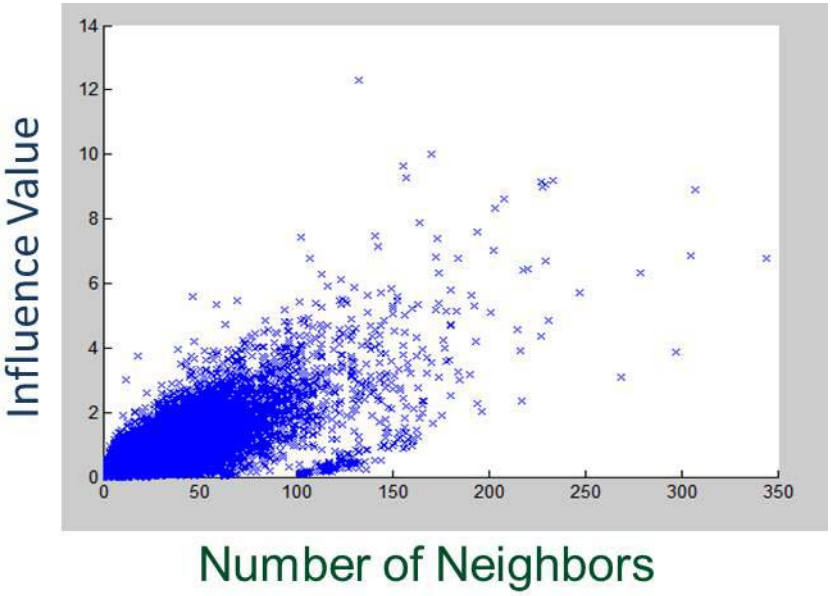


Figure 6.11: Influence score versus number of neighbors

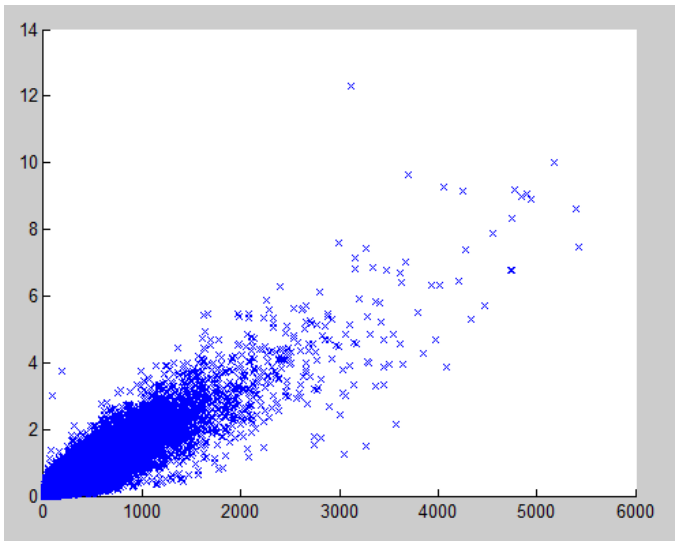


Figure 6.12: Influence score versus weighted degree

higher is his/her influence value.

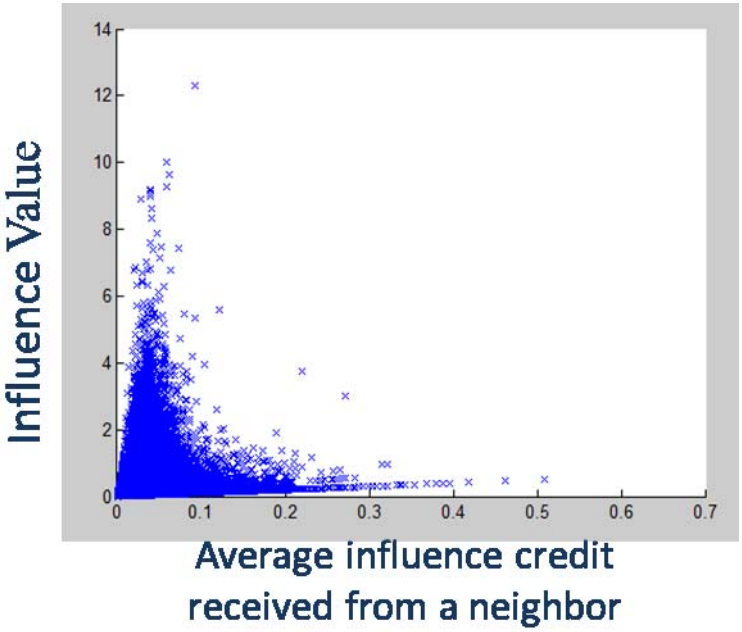


Figure 6.13: Average influence credit received from neighbors vs. influence value

Figure 6.13 shows that influence initially increases, then decreases w.r.t. average credit from a neighbor. From Figure 6.14 it is observed that influence decreases w.r.t standard deviation in credit received from neighbors. This implies that, unlike the MMORPG network, the top influencers are authors having low to moderate influence over a large number of collaborators. This difference might be attributed to the social investment involved in both relationships with the social interactions being much more indicative of influence in the DBLP network versus the MMORPG grouping network.

6.6 Conclusions

This chapter presents a framework for computing influence values using user behavior data. The proposed approach is applied on real social data from a massively multiplayer online game. Results show that the analysis was useful in discovering trends regarding the nature of influence. The top influencers were users surrounded by easily influenced nodes and in most cases influence was accumulated by focussing on a select few neighbors

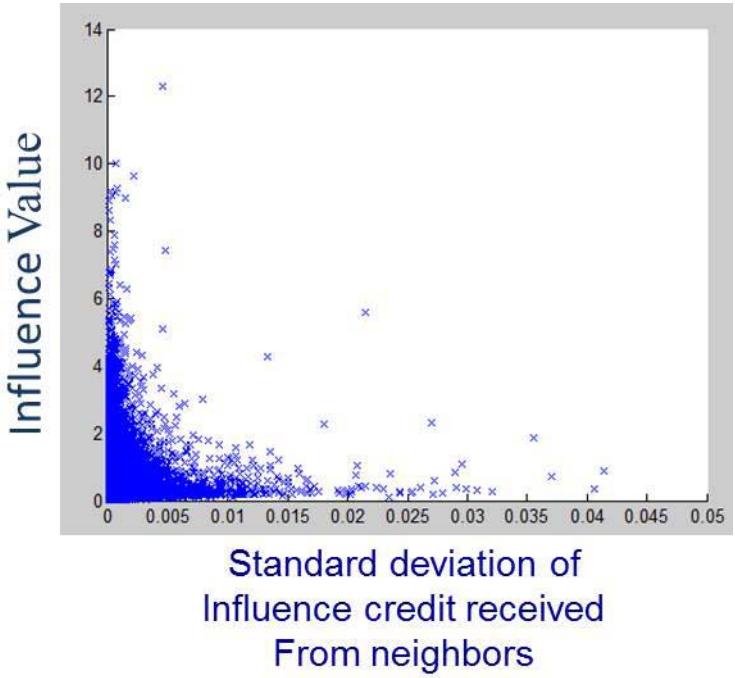


Figure 6.14: Standard deviation of influence credit from neighbors vs influence value

versus a larger set. Such information can be used to map out effects of cascade processes much more accurately. The results from such an analysis would be of immense value for any marketing efforts, community leaders, or any other party interested in the social outcomes of the user community. The influence values discovered from data can also be used with cascade process based techniques as they are much more representative of links indicating cascade propagation.

Chapter 7

Conclusion and Discussion

This thesis presents a multi-faceted analysis of information flow in social networks. Different types of analysis yeild different latent information regarding the nature of the underlying social network. With the internet becoming integral to everyday life, users naturally require the means to socialize with each other online. Socialization is very important part of any individuals personality and it affects a variety of variables regarding user behavior. Since socialization is such an important part of the overall experience, it is very important for online businesses to monitor and maintain factors responsible for healthy sustainence of the communities around their commodities.

Information flow based communities can be discovered via simulating multiple cascades. These latent clusters provide information on sub-communities that result when users in local regions tend to have much more in common than the rest of the network. This technique network also yeilds the extent of information flow in different regions of the network. Such information can be used by business owners to identify the different population segments in the user base as dictated by their interactions, further exploration of each cluster can result in strong insight on the nature of people in each of these segments.

Topic based analysis of communication content is a very effective way of discovering the topics of interest common to actors in the social network. Generally individuals come together to form a social network due to some common shared interest or goal. These shared interests/goals form the context of the network and are a prominent part of the actor-actor communciation occurring in the network. Often topical trends precede

some significant event and thus are strong indicators of the same. This sort of analysis can provide information on the different topics that users are currently talking about and allow businesses to forsee any developing trends in the community, allowing them sufficient time to react appropriately.

Actors' cognitions can be estimated via combining the knowledge from communication each actor receives. Measures of actors' beliefs similarity to other actors as well as reality are indicators of the process of information exchange and informal actor roles in the network. Constructing cognition of knowledge literals can be used to estimate sentiments of actors. These techniques are quite valuable to judge the sentiment of the community and as well as forsee trends of any event significance in the community.

Actors' influence can be estimated to discover which actors are beneficial or harmful to the community. Actors that are valuable to the community can be provided incentives and preferential treatments as rewards. Meanwhile, those that are harmful can be judged to be expelled form the community for their behavior. Influential actors can also be provided incentive to be early adopters and propagating the interest in a given commodity to larger parts of the network.

A toolbox consisting of the above set of techniques will be quite valuable for any commercial online system that allows users to interact with each other. Prominent examples of such social systems include Massively Multiplayer Online Games (MMOGs) such as World of Warcraft, Media sharing websites such as flickr, self moderating forums such as Reddit and social sites like Facebook, Twitter. Socialization is a very important aspect of human nature and as the internet becomes a bigger part of our lives, the popularity of such social platforms is only going to go up. This socialization proviedes a variety of advantages for all parties involved. Online businesses can now make it convenient for users to discuss and promote their products to others thus increasing their customer base. Allowing users a place to discuss and interact provides a sense of belonging and promotes a community feeling. All of these factors result in a positive experience for the user which in turn has a significant impact on the popularity of the products/services offerred. Understanding the trends and knowledge in the data generated by these systems is key for any online business as it helps ensure a healthy and active existence for communities around their products/services.

References

[1] Rob Cross, Nitin Nohria, and Andrew Parker. Six myths about informal networks and how to overcome them. *MIT Sloan Management Review*, 43(3):67–75, 2002.

[2] B. A. Nardi, S. Whittaker, and H. Schwarz. NetWORKers and their Activity in Intensional Networks. *Computer Supported Cooperative Work*, 11(1-2):205–242, 2002.

[3] R Dye. The buzz on buzz. *Harvard Business Review*, 78(6):139–146, 2000.

[4] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 57–66, New York, NY, USA, 2001. ACM.

[5] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Number 8 in Structural analysis in the social sciences. Cambridge University Press, 1 edition, 1994.

[6] D. Krackhardt. Cognitive social structures. *Social Networks*, 9(2):109–134, 1987.

[7] Peter V. Marsden. Network Data and Measurement. *Annual Review of Sociology*, 16:435–463, 1990.

[8] Barry Wellman. Computer networks as social networks. *Science*, 293:2031–2034, September 2001.

[9] M. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, 83(6):1420–1443, 1978.

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

[11] J. Goldenberg, B. Libai, and Muller. Using complex systems analysis to advance marketing theory development. *Academy of Marketing Science Review*, 2001.

[12] J. Goldenberg, B. Libai, and E. Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, pages 211–223, August 2001.

[13] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. pages 61–70. ACM Press, 2002.

[14] Stephen Morris. Contagion. *Review of Economic Studies*, 67(1):57–78, January 2000.

[15] Jennifer Wortman. Viral marketing and the diffusion of trends on social networks. *Science*, 285(11):8363–8374, 2008.

[16] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM.

[17] Stephen Foster, Walt Potter, Jiang Wu, Bin Hu, and Yu Zhang. A history sensitive cascade model in diffusion networks. In *Proceedings of the 2009 Spring Simulation Multiconference*, SpringSim '09, pages 5:1–5:8, San Diego, CA, USA, 2009. Society for Computer Simulation International.

[18] Tim Carnes, Chandrashekhar Nagarajan, Stefan M. Wild, and Anke van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *Proceedings of the ninth international conference on Electronic commerce*, ICEC '07, pages 351–360, New York, NY, USA, 2007. ACM.

[19] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In Sadagopan Srinivasan, Krithi Ramamritham, Arun Kumar, M. P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *WWW*, pages 665–674. ACM, 2011.

[20] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *In WINE*, pages 306–311, 2007.

[21] Harold Hotelling. Stability in competition. *The Economic Journal*, 39(153):41–57, 1929.

[22] Hee-Kap Ahn, Siu-Wing Cheng, Otfried Cheong, Mordecai J. Golin, and René van Oostrum. Competitive facility location along a highway. In *Proceedings of the 7th Annual International Conference on Computing and Combinatorics*, COCOON '01, pages 237–246, London, UK, UK, 2001. Springer-Verlag.

[23] Otfried Cheong, Sariel Har-Peled, Nathan Linial, and Jiri Matousek. The one-round voronoi game, 2003.

[24] Jan Kostka, Yvonne Anne Oswald, and Roger Wattenhofer. Word of Mouth: Rumor Dissemination in Social Networks. In *15th International Colloquium on Structural Information and Communication Complexity (SIROCCO), Villars-sur-Ollon, Switzerland*, June 2008.

[25] Venkatesan Guruswami. Rapidly mixing markov chains: A comparison of techniques. *A Survey*, 2000.

[26] Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1):5–43, January 2003.

[27] Deepayan Chakrabarti, Yiping Zhan, and Christos Faloutsos. R-mat: A recursive model for graph mining. In *In SDM*, 2004.

[28] Timothy A. Davis. The university of florida sparse matrix collection. *NA DIGEST*, 92, 1994.

[29] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Reto Meuli, Christopher J Honey, Van J Wedeen, and Olaf Sporns. Mapping the structural core of human cerebral cortex. *PLoS Biology*, 6(7):15, 2008.

[30] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. The dynamics of viral marketing. *ACM Trans. Web*, 1, May 2007, physics/0509039.

[31] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors: A multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell*, 29:2007, 2007.

[32] G. Karypis, V. Kumar, and Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48:96–129, 1998.

[33] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *in Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, 2001.

[34] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

[35] M. E. J. Newman. Fast algorithm for detecting community structure in networks. September 2003, cond-mat/0309508.

[36] Joshua R. Tyler, Dennis M. Wilkinson, and Bernardo A. Huberman. Communities and technologies. chapter Email as spectroscopy: automated discovery of community structure within organizations, pages 81–96. Kluwer, B.V., Deventer, The Netherlands, The Netherlands, 2003.

[37] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, April 2004.

[38] J. Shetty and J. Adibi. Enron email dataset. Technical report, 2004.

[39] J. Diesner and K. M. Carley. Exploration of Communication Networks from the Enron Email Corpus. *Proceedings of Workshop on Link Analysis, Counterterrorism and Security, SIAM International Conference on Data Mining 2005*, pages 3–14, 2005.

[40] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[41] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[42] Joerg Reichardt and Stefan Bornholdt. Statistical mechanics of community detection, 2006. cite arxiv:cond-mat/0603718.

[43] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[44] Xuerui Wang, Natasha Mohanty, and Andrew Mccallum. Group and topic discovery from relations and text. In *In Proc. 3rd international workshop on Link discovery*, pages 28–35. ACM, 2005.

[45] Andrew Mccallum, Andrs Corrada-emmanuel, and Xuerui Wang. Topic and role discovery in social networks. In *In IJCAI*, pages 786–791, 2005.

[46] Ding Zhou, Eren Manavoglu, Jia Li, C. Lee Giles, and Hongyuan Zha. Probabilistic models for discovering e-communities. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 173–182, New York, NY, USA, 2006. ACM.

[47] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 807–816, New York, NY, USA, 2009. ACM.

[48] Sebastín A. Rios, Felipe Aguilera, Francisco Bustos, Tope Omitola, and Nigel Shadbolt. Leveraging social network analysis with topic models and the semantic web. In *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Volume 03*, WI-IAT '11, pages 339–342, Washington, DC, USA, 2011. IEEE Computer Society.

[49] Maureen R Heald, Noshir S Contractor, Laura M Koehly, and Stanley Wasserman. Formal and emergent predictors of coworkers' perceptual congruence on an organization's social structure. *Human Communication Research*, 24(4):536–563, 1998.

[50] David Krackhardt and Jeffrey Hanson. Informal networks: The company behind the chart. *Harvard Business Review*, 71(4):104–111, jul/aug 1993.

[51] Noshir S. Contractor, Dan Zink, and Mike Chan. Iknow: A tool to assist and study the creation, maintenance, and dissolution of knowledge networks. In *In Toru Ishida (Ed.), Community Computing and Support Systems, Lecture Notes in Computer Science 1519*, pages 201–217. Springer-Verlag, 1998.

[52] S Kripke. Kripke s. - a completeness theorem in modal logic.pdf. *The Journal of Symbolic Logic*, 24(1):1–14, 1959.

[53] A P Dempster. A generalization of bayesian inference. *Journal of the Royal Statistical Society*, 30(2):205–247, 1968.

[54] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[55] Michael W. Berry and Murray Browne. Email surveillance using non-negative matrix factorization. *Comput. Math. Organ. Theory*, 11(3):249–264, October 2005.

[56] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, October 2003.

[57] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[58] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[59] Janyce Wiebe, Theresa Wilson, and Matthew Bell. Identifying collocations for recognizing opinions. In *In Proc. ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, pages 24–31, 2001.

[60] Xue Bai, Rema Padman, and Edoardo Airoldi. Sentiment extraction from unstructured text using tabu search-enhanced markov blanket. In *In Proceedings of the Workshop on Mining the Semantic Web, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 24–35. Springer-Verlag, 2004.

[61] Bo Pang and Lillian Lee. Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[62] Anthony Aue and Michael Gamon. Customizing sentiment classifiers to new domains: A case study. In *RANLP*, 2005.

[63] David Kempe, Jon Kleinberg, and va Tardos. Influential nodes in a diffusion model for social networks. In *IN ICALP*, pages 1127–1138. Springer Verlag, 2005.

[64] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 420–429, New York, NY, USA, 2007. ACM.

[65] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 199–208, New York, NY, USA, 2009. ACM.

[66] Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 1019–1028, New York, NY, USA, 2010. ACM.

[67] R. Agrawal, M. Potamias, and E. Terzi. Learning the nature of information in social networks. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, ICWSM '12, 2012.

[68] Amit Goyal, Francesco Bonchi, and Laks V.S. Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, pages 241–250, New York, NY, USA, 2010. ACM.

[69] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.