5-3-2017

INFLUENCE ANALYSIS TOWARDS BIG SOCIAL DATA

Meng Han

INFLUENCE ANALYSIS TOWARDS BIG SOCIAL DATA

by

MENG HAN

Under the Direction of Yingshu Li, Ph.D. and Zhipeng Cai, Ph.D.

ABSTRACT

Large scale social data from online social networks, instant messaging applications, and wearable devices have seen an exponential growth in a number of users and activities recently. The rapid proliferation of social data provides rich information and infinite possibilities for us to understand and analyze the complex inherent mechanism which governs the evolution of the new technology age. Influence, as a natural product of information diffusion (or propagation), which represents the change in an individual's thoughts, attitudes, and behaviors resulting from interaction with others, is one of the fundamental processes in social worlds. Therefore, influence analysis occupies a very prominent place in social related data analysis, theory, model, and algorithms.

In this dissertation, we study the influence analysis under the scenario of big social data. Firstly, we investigate the uncertainty of influence relationship among the social network. A novel sampling scheme is proposed which enables the development of an efficient algorithm to measure uncertainty. Considering the practicality of neighborhood relationship in real social data, a framework is introduced to transform the uncertain networks into deterministic weight networks where the weight on edges can be measured as Jaccard-like index. Secondly, focusing on the dynamic of social data, a practical framework is proposed by only probing partial communities to explore the real changes of a social network data. Our probing framework minimizes the possible difference between the observed topology and the actual

network through several representative communities. We also propose an algorithm that takes full advantage of our divide-and-conquer strategy which reduces the computational overhead. Thirdly, if let the number of users who are influenced be the depth of propagation and the area covered by influenced users be the breadth, most of the research results are only focused on the influence depth instead of the influence breadth. Timeliness, acceptance ratio, and breadth are three important factors that significantly affect the result of influence maximization in reality, but they are neglected by researchers in most of time. To fill the gap, a novel algorithm that incorporates time delay for timeliness, opportunistic selection for acceptance ratio, and broad diffusion for influence breadth has been investigated. In our model, the breadth of influence is measured by the number of covered communities, and the tradeoff between depth and breadth of influence could be balanced by a specific parameter. Furthermore, the problem of privacy preserved influence maximization in both physical location network and online social network was addressed. We merge both the sensed location information collected from cyber-physical world and relationship information gathered from online social network into a unified framework with a comprehensive model. Then we propose the resolution for influence maximization problem with an efficient algorithm. At the same time, a privacy-preserving mechanism are proposed to protect the cyber physical location and link information from the application aspect. Last but not least, to address the challenge of large-scale data, we take the lead in designing an efficient influence maximization framework based on two new models which incorporate the dynamism of networks with consideration of time constraint during the influence spreading process in practice. All proposed problems and models of influence analysis have been empirically studied and verified by different, large-scale, real-world social data in this dissertation.

INDEX WORDS: Algorithm, Influence Analysis, Big Data, Social Network, Data Mining, Data Privacy, Cybersecurity

INFLUENCE ANALYSIS TOWARDS BIG SOCIAL DATA

by

MENG HAN

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy
in the College of Arts and Sciences
Georgia State University

2017

INFLUENCE ANALYSIS TOWARDS BIG SOCIAL DATA

by

MENG HAN

Committee Co-Chairs: Yingshu Li
Zhipeng Cai

Committee: Anu G. Bourgeois
Yanqing Zhang
Yi Zhao

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2017

DEDICATION

This dissertation is dedicated to my parents Shuyan Wu and Fengxiang Han for their endless support, love and passion during my Ph.D. years. I cannot finish my Ph.D. without their love and encouragement. Also dedicate to my lovely wife Weiqiong Zhu, all my family members and my past adventurous youth.

ACKNOWLEDGEMENTS

Undertaking this Ph.D. has been a truly life-changing experience for me and I would never have been able to finish my dissertation without the guidance of the committee members, the help from my group, and the support from my family and friends. I would like to show my great appreciation to all of them.

First and foremost, I wish to express my deepest gratitude to my advisors, Dr. Yingshu Li and Dr. Zhipeng Cai. They provided me with a great environment for doing research. They always gave me the greatest tolerance and instruction during my graduate studies. They inspired me in many aspects, and also supported my research financially. Dr. Li has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under her guidance. I am extremely grateful for what she has offered me. By Dr. Cai's broad academic vision and incisive academic perspectives, he provided me many suggestions on my research with foresight and sagacity. Besides research, Dr. Cai also gave me a lot of help on my study and career planning.

I would also like to thank my committee members, Dr. Anu Bourgeois, Dr. Yanqing Zhang, and Dr. Yi Zhao. Dr. Bourgeois, who gave me great support for my Ph.D. study, working, and my job searching. Special thanks go to Dr. Zhang and Dr. Zhao, who were willing to spare time to participate in my defense committee at their earliest convenience.

I am very thankful to the professors and staffs at our department, especially Dr. Raj Sunderraman, for his help with my Ph.D. study, teaching, research, and for his great support in my job hunting; Dr. Yi Pan, for his profound insights and suggestion; and Dr. Sushil K. Prasad for his advising during the time when I was working for the student branch of IEEE at Georgia State University. I thank Ms. Tammie Dudley, Ms. Adrienne Martin, Mr. Jamie Hayes, Mr. Paul Bryan, Ms. Venette Rice, and Ms. Celena Pittman for their patient help, which makes my life much easier and more fruitful.

I would like to express my deepest appreciation to my Master advisor Prof. Jianzhong Li, who brought me to the research community and encouraged me to make progress in the research world. I also appreciate the help and support from many other academic mentors, especially Prof. Jinbao Li, Dr. Wei Zhang, Dr. Zhaonian Zou, and Dr. Jinbao Wang. Many thanks go to my group members and following students. Special thanks go to my group members Dr. Mingyuan Yan, Dr. Guoliang Liu, Dr. Dongjing Miao, Zhuojun Duan, Xu Zheng, Zaobo He, Ji Li, Yi Liang, and Yan Huang, my friends Dr. Jing (Selena) He, Dr. Shouling Ji, Dr. Chunyu Ai, Dr. Yunmei Lu, Dr. Yuan Long, Dr. Chenguang Kong, Peisheng Wu, Jin Zhang, Huan Kuang, and Qixi Wu who helped me and brought me happiness beyond the research.

My parents deserve special mention for their love, encouragement, and support throughout my life. The dissertation is impossible without their support. My deep appreciation goes to my wife, Weiqiong Zhu. Her support, encouragement, patience and unwavering love were undeniably the bedrock of my achievements in the past years. Her tolerance of my occasional bad moods is a testament in itself of her unyielding devotion and love. Her constant companion for tens of thousands miles travel to the desert, oceans, mountains brought me the greatest courage to work harder for my Ph.D. degree and my career.

Last but not least, it is a pleasure to thank everybody who made the dissertation possible, as well as express my apologies that I could not mention personally one by one.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- OSN - Online Social Network
- IM - Influence Maximization
- IC - Independent Cascade
- LT - Linear Threshold
- DP - Dynamic Programming
- RW - Random Walk
- DS - Dominating Set
- TIC - Time Constraint IC Model
- TLT - Time Constraint LT Model
- WCC - Weakly Connected Component
- SCC - Strongly Connected Component
- SNAP - Stanford Large Network Dataset Collection
- ICOT - IC Model with Opportunistic Selection and Time Decay
- BICOT - Broadly Influence Maximization Problem Under ICOT Model

Chapter 1

INTRODUCTION

1.1 Background and Motivations

A set of social actors and a set of links among them construct a social network. The definition of a social network could be dated back to the late 1800s when both Emile Durkheim and Ferdinand Tonnies foresaw the phenomena of social groups [139]. The researchers in the fields of psychology, anthropology and mathematics work independently for the developments of social networks. According to the definition raised by Rashotte *et al.* [118], *influence*, an important concept in social data, is "the change from an individual's thoughts, feeling, attitudes, and behaviors that result from interaction with other individual or group." Influence, the natural product of information diffusion (or propagation), is one of the fundamental processes taking place in social networks. Therefore, influence analysis occupies an prominent place in social data.

Sociologists and other related scientists never stop trying to explore social networks since the social networks also construct the modern social foundation. Many researchers have sought to test or examine that whether there is an influence and how did people influence each other in social networks In 2007, Nicholas *et al.* [39] published their years of research results based on the historical data about the spreading of obesity over 32 years. In the same research field, David *et al.* [8] proposed another idea about the spread of obesity in social networks based on the simulations which further considered the group effect in obesity spreading. Both works tried to explore how does the influence diffusion in social networks affect obesity. Their models also consider obesity as a "contagious" phenomenon that can be caught if most social contracts are deemed obese. The interaction of social networks with environmental factors could not be explored because it was not accounted for the general model where the social networks were proposed as a means to mitigate the obesity epidemic.

Even so, prior to the Internet, quantitative data of social network were scanty, the further influence analysis in social networks was still in the slow-lane. What the results as mentioned above have in common is that they are all from the real experiments with actual social lives on questionnaires or laboratory tests, which are limited by the experiment size. The above results are hard to be expanded to large social entities. Also, the methodologies mentioned above cannot be applied to large scale social networks. In the Internet Age, each month, more than 1.3 billion users are active on Facebook and 190 million unique visitors are active on Twitter. Furthermore, 48% of Facebook users who are 18-34 years old check their online page when they wake up, and 98% of 18-24 year old people are involved with at least one kind of social media [1]. Online Social Networks (OSNs) have seen a rapid rise in the number of users and activities in the past years such as Facebook, Twitter, LinkedIn, *etc.*, which means influence analysis in social networks has entered a new epoch. As an emerging part of social networks, OSNs represent most characteristics of traditional social networks in a digital version on a large scale. OSNs have kept growing for more than one decade and occupied an increasingly more important position in social networks. In OSNs, we can get more research results that were once unimaginable before. OSNs are not just a large continent size recreation or entertainment platform. Many OSNs could also be used for work purpose such as watching the market/competitors, which significantly and positively impact employees' performance to some extent [93] [153].

The emergence of OSNs and the accompanying massive amounts of social data pose a number of both computational challenges and opportunities to academia and industry, especially those involving influence analysis. As far as we know, although influence analysis in social data have attracted a lot of attentions, there are still many challenges for both academia and industry.

To analyze influence, the first problem is understanding the relations in social data. But in real life, uncertainty exists in many kinds of networks. The uncertainty may result from network components themselves or external factors. How to figure the uncertainty

[1] http://www.statisticbrain.com/facebook-statistics/

of influence in social data out is the very first challenge. However, in practice, a clear relationship among pair of nodes is difficult to detect in huge uncertain complicated networks. Due to the increase of complexity in modern networks especially social networks (Facebook, Twitter, and LinkedIn, *etc.*), it becomes more and more difficult to efficiently identify the relationship in networks. Thus, we propose an uncertainty generation framework to capture the uncertainty among social network to help further understand the influence propagation.

After the analysis of uncertainty of influence in social data, based on many pieces of literature results, as one of the most popular research areas, many works for influence analysis is focusing on the influence maximization in social networks. For example, [88, 95] proposed several models to simulate the influence diffusion process. However, influence maximization in social networks is still a developing problem. Due to the complexity and diversity of the phenomena, researchers are still facing a lot of challenges concerning how to analyze and utilize the influence in OSNs. However, many challenging issues are upcoming. First, OSNs are dynamic networks; hence, the change of network topology directly affects the diffusion of influence. Besides, almost all of the classical models, such as *IC* and *LT* together with their many derived varieties involving Monte Carlo simulation, which is incredibly resources intensive. Since the influence maximization problem is unfortunately *NP-hard*, it is almost impossible to find the most optimal influential node set in a large scale network. It is even harder to pursue the optimal node sets that can maximize the influence in a dynamic social network. Besides all the other challenges, updating a network to reflect its dynamic nature with time is extremely resource consuming in large social networks. Therefore, we are interested in proposing an efficient integrated solution to select the most influential nodes in dynamic social networks considering the challenges and features of OSNs.

Next, to further model and maximize the influence from a reality aspect, *Timeliness* refers to the phenomena that the effect of influence would decay with time; *acceptance ratio* measures the percentage of influence which gets a response; and influence *breadth* aims at maximizing influence not only by having more users; are three very important factors of the model and resolution. We address the problem of identifying the node set which maximizes

influence in practical social networks. We proposed a model incorporates influence decay function, opportunistic selection and broader maximization accommodating to three factors: timeliness, acceptance ratio, and breadth. We take the first step to explore the relationship between the breadth and depth of influence and propose the model *BICOT*. Comparing to previous related works, the number of communities is incorporated to measure the breadth of the influence, which is also very novel.

Furthermore, most of the existing works, unfortunately, neglect the fact that the sensed location data in the cyber-physical world could also play an important role in the influence prorogation process. Even though a few works consider a little sensed information to enhance the influence maximization, the privacy protection issue of the sensed data (location, social relationship) was directly exposed to the public. We address the problem of maximizing influence in both cyber-physical world and online social world with privacy preserving and propose a very comprehensive framework as the resolution for the problem. We firstly merge both the sensed location data from cyber-physical network and relationship data from online social network into a unified framework with one novel model, then an efficient algorithm to solve the influence maximization problem is provided within our resolution. Besides, our model could not only support the influence maximization problem, but also support other applications related to both sensed location data and online social data. Furthermore, our privacy-preserving mechanism could protect the sensitive location and link information from the application aspect.

Last but not least, the influence analysis in big scale data is still a very challenging problem. In order to further study the special features of influence analysis in big social data. We implement two models of influence maximization with time constraint on the up-to-date platforms Hadoop and Spark. According to the result of our real data experiment, we explore the potential of new computational framework to our research problem and demonstrate the expansibility of influence analysis in big scale data.

1.2 Organization

The rest of this dissertation is organized as follows: Chapter 2 summarizes the related literatures. Chapter 3 investigates the uncertainty of influence in social data. Chapter 4 studies the probing techniques to update the dynamic of social data with very few cost and maximizes the influence in the social data. Chapter 5 explores the influence maximization problem with considering of timeliness, acceptance ratio in both depth and breadth. Chapter 6 proposes the resolution of influence maximization for the data from both cyber physical world and online social network with privacy preservation. Chapter 7 develops the applicability of influence maximization in a large scale with the modern big data platforms. Chapter 8 conducts the future research directions. And the last chapter concludes this dissertation.

Chapter 2

RELATED WORKS

Other than results from social science we mentioned in Chapter 1. Many kinds of literature have been proposed from different aspects of influence analysis. At an American high school, Salath *et al.* [123] obtained high-resolution data of proximity interactions during a typical day, and their work helps with the reconstruction of a social network for infectious disease transmission by using wireless sensor network technology [60, 61]. Through simulations, they showed that targeted immunization using the contact-network data is much more efficient than random immunization. Stehle *et al.* [129] reported a similar result like [123] in a French primary school. The team headed by Stehle also provided several public-health implications of infectious diseases by collecting a period of history data in their experiments. By analyzing the real tests in two middle schools in Germany, Ralf *et al.* [152] aimed to test the operating social mechanisms that underlie the efficacy of bullying prevention programs. Kwon [91] *et. al.* analyzed how individual differences affect user's intentions to use social network services with a Technology Acceptance Model (TAM) in their psychology-based research.

The emergence of OSNs and the accompanying large amounts of data pose a number of both computational challenges and opportunities to academia and industry, especially those involving influence analysis. Fig.2.1 shows the number of high impacted publications regarding influence maximization in the recent years. [1]

To understand OSNs further deeply, many research results have been published recently. By combining content-based and network-based approaches, Tang *et al.* [135] proposed some techniques to predict influence. Two medical data sets have also been tested to evaluate their proposed techniques UserRank and Weighted in-degree. Based on Goyal and Kearns' work

[1] The statistic result focuses on data mining and social analysis and may not include all the literatures in all relevant areas.

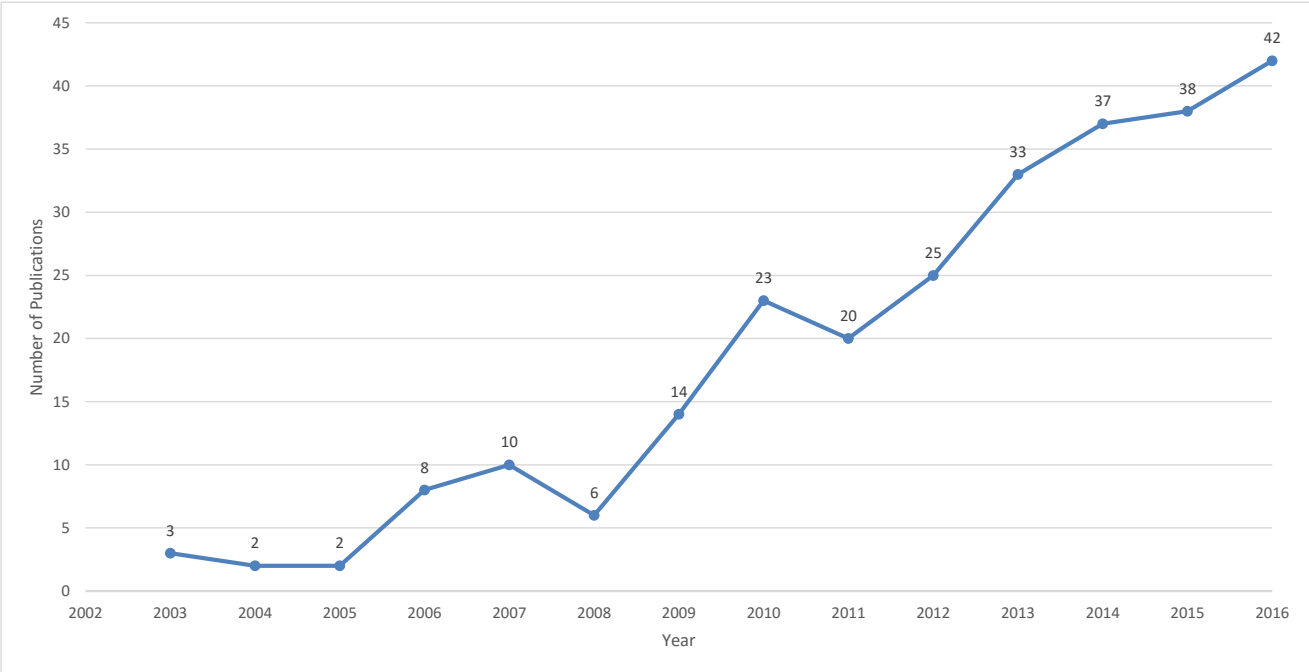


Figure 2.1. Publications regarding influence analysis in the recent years

in [54], He *et al.* [68] studied the Price of Anarchy of the competitive cascade game under a linear threshold model in a theoretical aspect. Considering the price of a product in a social network, Francis *et al.* [12] investigated the problem of how to find an optimal monopoly pricing and the relationship between the consumers and their neighbors. From a tie-strength perspective, Zhao *et al.* [171] addressed the information diffusion problem in social networks such as how fast does the information propagate, what is the role of weak ties for information diffusion, and so on. The authors in [171] also gave some business suggestions for the cost-efficient and secured information propagation for online social networking sites such as pushing information to friends using a strong-tie-first strategy, and preventing privacy by removing active weak ties from local communities from another perspective. Rakesh and Agrawal summarized the results of their recent investigations around the nature of information, people and their relationships in social networks [2]. Their work includes information diffusion [3], analysis of opinion formation [10, 40], and the factors influencing an individual's continued relationship in a social group [15].

Influence maximization is one of the key topics for the analysis of influence, the *IC*

(Independent Cascade) and *LT* (Linear Threshold) model [88] together with their extensions set the foundation for most of the existing cascading researches. Since Kempe *et al.* [88] formulated the influence maximization problem as an optimization problem, a series of empirical studies have been performed on influence learning [49, 121], algorithm optimization [131, 53, 51], scalability promotion [141, 30], influence of group conformity [133], and influence of location exploration. Anadiotis, Christos, *et al.* [7] illustrated how information-centric networks can effectively address practical issues rising in multimedia applications and social networking technologies. In [7], the authors provided a certain observation that the influence in social networks is information-centric especially within a community, which motivated us to maximize the influence from the community aspect to the global network [5]. Leskovec *et al.* [95] modeled the outbreak detection problem and proved that the influence maximization problem is a special case of their new problem. In [95], a Cost-Effective Lazy Forward (*CELF*) scheme taking the advantage of the submodular property is proposed. It achieved 700 times speedup in selecting seed vertices compared with the classical greedy algorithm [88]. *CELF* set a milestone of using brute force solution to solve the influence maximization problem. But as indicated in [30], solutions proposed in *CELF* lack of scalability. Therefore, Chen *et al.* [30] developed several efficiency heuristic algorithms based on the arborescence structure, which is a structure that could handle million-sized graphs. The proposed algorithm spreads influence similar to the way that the greedy algorithm does, while achieving more than six orders of magnitude faster than the greedy algorithm. Most recently, Borgs *et al.* [13] first proposed a result, which avoids the limitation of traditional greedy algorithm. Their research showed that a drastically different technique for influence maximization under the *IC* model. From the perspective of the opposite, [13] defined a reverse reachable (RR) set for node v in the network is the set of nodes that can reach v. Then by sampling algorithm, the algorithm generated a certain number of random possible world of RR sets from the network. Follow the rationales that if a size-k node set S could covers most RR sets, then S has a higher probability to maximize the expected spread among all size-k set in the network. Their theoretical result showed that when parameter τ is set to

$\Theta(k(m+n)\log n/\epsilon^3)$, the algorithm could run in time linear to τ, and return a $(1-1/e-\epsilon)$-approximate with a constant probability. Furthermore, Tang *et al.* [137] proposed a more practical framework TIM which guarantees the same theoretical complexity bound of [13] and keeps at least probability $1-n^{-l}$. TIM supports a triggering model, which is a more general model includes both IC and LT as special cases. Next, [136] proposed a martingale approach to support a larger class of information diffusion models while providing same accurate results with small computation overheads.

2.1 Understand Influence on Social Edges

There are many existed works in other social network analysis research such as community detection and network clustering based on relationships in traditional certain networks, and some models for uncertain data mining. The survey paper [92] and [1] are very good references.Following are the works mining and detecting relationships in certain network [101, 83], such as mining communities in YouTube [16] and mining interest groups in mobile social networks [155]. The work of [156] developed an algorithm that can identify the nodes which bridge clusters and nodes marginally connected to outliers. Since this technique needs a parameter to control the minimum similarity in a graph, the algorithms in [14] overcome the difficulty by finding only the best cluster boundaries to cluster a network. However, all the above-mentioned works did not consider uncertainty in networks.

The inherent uncertainty of influence in networks has to be considered for conducting accurate analysis and mining. The model in [182] was established for uncertain graphs (also named as probabilistic graphs) [77], in which each edge is associated with an existence probability. The following works studied different issues considering uncertainty in networks. Jin *et al.* [84] introduced a sampling scheme which discovers highly reliable subgraphs with high probability. Since the shortest path in an uncertain graph is different from the ones in a deterministic graph, two new kinds of queries appear which are threshold-based shortest path queries [162] and distance-constraint teachability queries [85]. Considering the uncertainty in networks, the work in [116] introduced a framework for processing k nearest neighbor

(k-NN) queries. After proposing some novel distance functions in uncertain graphs [67], the authors designed a sampling algorithm which can prune the search space efficiently. Unfortunately, these works and models cannot deal with community detection in uncertain networks. Moreover, all the above works did not consider the common neighbor factor, which has a critical impact on identify clear relationships in uncertain graphs.

To the best of our knowledge, our proposed uncertainty relationship in network is the first one to study the uncertainty generation problem based on the neighborhood relationship in uncertain networks [65, 66].

2.2 Dynamic Probing for Influence

Besides literatures focusing on the fundamental influence maximization problem and its several variants mentioned above, existing literatures related to our work can be divided into two groups: the dynamic network models and community-based influence maximization.

Zhuang *et al.* [180] considered structure changing by probing a subset of nodes in a social network to estimate the real influence diffusion process. However, different from us, [180] focuses on investigating a network by probing several nodes with the highest influence increment instead of using the community scope. In a large-scale network, probing and updating several specfic nodes is very time-consuming, and it is hard to calculate the influence increment for each node. In contrast, the community, which is a natural structure in social environments has more comprehensive information and are much easier to manipulate. On the basis of the discussion above, different from [180], our work takes advantages of both probing techniques and community features. We are interested in the dynamic probing algorithm by taking a community as the unit instead of a node. As we stated, a community is a very natural structure in social networks. For a dynamic network, probing one node means updating the node itself and all the links related to the node. Nodes within the same community have more links and relationships that can affect each other. Probing the network node by node will result in a lot of duplicated operations and costs [62]. In our solution, we look for the most active community and probe it. The cost is reduced by probing the

community as a unit.

On the other hand, taking the community into account brings more benefits for the influence maximization. Individuals within the same community have more frequent contacts and thus are more likely to be influenced by each other. In contrast, individuals from different communities have much less contact with each other and thus are less likely to influence each other. This community property suggests that it might be a good approximation to choose influential nodes within communities instead of the whole network. Obviously, it will be more efficient to select influential nodes within communities. Based on the previous discussion, we employed dynamic programming to take full advantage of the community and probing techniques. Reference [180] only provided the node probing method while excluded the community probing and other techniques related to communities. The algorithm in [180] cannot benefit the efficiency of the influence maximization either.

Li *et al.* [102] addressed the problem of finding densely connected subgraphs that satisfy the query conditions considering the influence of a community in a network. However, their method is based on the concept of k-core, which is not an influence diffusion expectation model but network structure model. This kind of model could not provide influence expectation measures which actually do not completely follow the information diffusion process. The drawback of core-based model has been proved and verified in many other experiment in literatures [26] and [120] to some extent. Wang *et al.* [150] tried to reduce computation cost by dividing a network into many communities. The greedy algorithm was first run in each community and the increments of the expected influence for each community were calculated. A dynamic programming algorithm was first proposed to select the optimal community. Then, the most influential nodes from each community were compared and selected. This process runs iteratively until the top-k influential nodes were found. Different from our work, timeliness had not been considered in their model. Besides, their partition method just divides a network into disjointed communities for the purpose of reducing computation cost without incorporating any other advantage of a network. Additionally, in real applications, communities in a social network naturally overlap because people have multiple roles

in different situations for most social networks.

To the best of our knowledge, none of the existing approaches consider merging the dynamic network probing problem from the community scope with the community-based divide-and-conquer techniques together to solve the influence maximization problem [64].

2.3 Broaden the Influence Propagation

Besides the fundamental influence maximization problem and several variants mentioned above, there are two kinds of previous works related to our influence exploring with community problem: dynamic network models and structural analysis for influence diffusion. The phenomena of time delay in influence diffusion have been explored in statistics. Timeliness concerned by us, different from time decay, emphasize more on the delivery time of influence. The observation in [82] shows that the heterogeneity of human activities has an important effect on influence diffusion. Thang et al. [42] modeled influence maximization by limiting the influence of nodes that are within d hops from the seeding for some constant $d \geq 1$. The authors proposed algorithm VirAds which guarantees a relative error bound of $O(1)$ when the network follows power-law. They also provided theoretical analysis to show the hardness of the model. They further extended the previous algorithm to obtain a near optimal solution with a ratio better than $O(\log n)$. Chen et al. [29] proposed the Independent Cascade model with meeting events (IC-M) to capture time-delay. Differently, our model not only considers the time decay and acceptance ratio of influence in dynamic networks, but also take structural breadth of a network into account.

Wang et al. [150] tried to reduce the computation cost by dividing a network into many communities. They run the greedy algorithm in each community and calculate the expected influence increase of each community. Then, a dynamic programming algorithm was proposed to select the optimal community, then choose the most influential nodes from each community [158] afterward. This process runs iteratively until the top-k influential nodes are obtained. However, they do not take the timeliness into account and their disjoint communities partition could not take care the overlap of different communities in real.

None of the existing approaches considers the time sensitivity, acceptance ratio and both the influence spreading breadth and depth together to the best of our knowledge [63].

2.4 Influence in Cyber-physical World

Social network is constructed by a set of nodes and connections among those nodes, which not only exist in the online social world but also exist in sensed cyber-physical world. Thus far, most of existing works focus on the influence model and maximization algorithm optimization, but unfortunately neglected the fact that the sensed data in cyber-physical world could also spread the information and enforce the influence during the process of message diffusion. There is only one notable literature considering sensed location information for influence maximization proposed by [97]. But the objective, which focus on answering the influence spread query of particular nodes vertices within a special region, is not a classical influence maximization requiring computing top-k influential nodes [6]. Therefore, to fill the gap between the sensed cyber-physical world and online social network, we are working on a comprehensive resolution to solve the classical influence maximization in a heterogenous basis.

With the development of modern mobile devices, the connection between cyber-physical network and online social network is significantly strengthened [45]. The sensed location data has been studied from different aspects for a while but not been taken into account to the classical influence maximization problem yet. Mining location record is one of most common and important works for location knowledge analysis [177] [25]. But without corresponding social information in the online social world, only sensed location data is still very limited. Different from traditional location knowledge analysis, we put both the sensed cyber physical data and online social data together to generate a novel heterogeneous network to solve the influence maximization problem. Two real life datasets Brightkite and Gowalla are studied in our evaluation section. Originally, the two datasets study friendships [38] in online social networks and users' movements in the physical world.

Furthermore, another part of related literatures are discussing how to preserve the

sensed location information and identity. It is a very hot topic regarding the sensed location privacy protection [59, 108]. We can roughly divide these research into three categories: mainly focusing on the location privacy preserving mechanism [11], primarily focusing on adding perturbed data actual user trajectories [46], and mainly focusing on the analysis and search for the location privacy metric that allows relatively fair comparison between utility and privacy [154].

To sum up, in the relevant literatures, a comprehensive framework combining both sensed cyber-physical information and online social information together for influence maximization with privacy preserving is not available so far [58].

2.5 Influence Analysis in Big Data

After many works appeared to solve the influence maximization problem, one of very important problem came out is how to deal with the data in large scale. Several efforts have been made to adapt influence model to scalable. Chen *et al.* [30] showed that computing influence spread in the independent cascade model is #P-hard problem. Then addressed the scalability issue, they proposed efficiency heuristic algorithm by restricting computations on the local influence regions of nodes. Additional, a tunable parameter for users to control the balance between the running time and the influence spread of the influence was also proposed [109]. Considering an independent influence path as an influence evaluation unit, an approximation algorithm named as Independent Path Algorithm (*IPA*) were proposed to approximate the influence. The parallel versions of *IPA* speeds up further as well as the number of CPU cores increases, which can be adapted to a larger size of datasets [89]. Another greedy algorithm named *SMG*, which stands for State-Machine Greedy is proposed recently by M. Heidari *et al.* [76]. The main idea improves the speed of greedy algorithms by preventing recalculation done by older methods. *SMG* improved the traditional greedy algorithm from time complexity of triggering nodes in the startup queue, reducing the time of graph construction, and preventing re-traverse of nodes. According to their experiment, *SMG* has a better performance than *CELF*. However, *SMG* does not consider the effect of

structure on the time complexity which still an open problem in these kinds of research. We still have not seen any results take both the influence with time constrain and influence decaying process into account. Furthermore, along with the increasing of data scale, only single machine implementation struggles to satisfy the need in the big data era. Different from previous related works, we proposed a time constraint model with influence decay to catch the main feature of influence in real life. And we also deploy our models and algorithms to the up-to-date platforms for further work reference [57].

Chapter 3

UNDERSTAND INFLUENCE ON SOCIAL EDGES

3.1 Introduction

Networks such as the Internet, social networks [99, 74, 143, 128, 147, 148], wireless networks [127, 44, 124, 157, 166, 107, 174, 80, 98, 176, 168], biological networks [138], *etc.* are now indispensable in our daily life. Most networks are deterministic on the aspects of network settings [146, 81, 22, 172, 173, 126, 19, 167], routing strategies [56, 28, 165, 27, 70, 36, 170, 164, 21, 18, 24, 20, 17], coverage [48, 4, 103, 140], traffic patterns [159, 160], user information *etc.* In the recent years, there has been tremendous interests in mining and discovering implicit knowledge from various networks [34, 179, 55, 35, 33, 100, 69].

In real life, uncertainty exists in all kinds of networks. The uncertainty may result from network components themselves or external factors. On the one hand, most networks whose structures and features are changing all the time are dynamic. For example, in a social network, a group of colleagues forms a community when they are in the same company. In due time, such a colleague relationship may be broken as some of them begin to work in another company while some of them start graduate studies. On the other hand, uncertainty is caused by the data generation process, and the variety of networks. Different data acquisition techniques and data description methods may result in incomplete and inaccurate data which aggregates network uncertainty. Therefore, how to identify relationships in networks considering uncertainty is very stringent.

However, in practice, a clear relationship among pairs of nodes is hard to detect in huge uncertain complicated networks. Due to the increase of complexity in modern networks especially social networks (Facebook, Twitter, and LinkedIn, *etc.*), it becomes more and more difficult to efficiently identify relationship in networks. Following are the emerging challenges.

First, even today, how to model and define uncertainty in real life is still an open problem. To conduct experiments on uncertain networks, almost all the existing works are evaluated based on the PPI (Protein-Protein Interaction) network, which is a very famous uncertain graph database representing protein interactions for different organisms obtained from the STRING database [1]. Furthermore, although uncertainty exists, there is no representative uncertain data set can be used for applications such as social networks and wireless networks, because there is no convincing model or method to generate it. Second, representing uncertain structural data(graph) is much harder to manipulate than deterministic graphs. Even if we have a reasonable uncertainty model, the cost of managing and mining such uncertainty in networks is still very expensive. To estimate an expected relationship in uncertain graph usually incurs high computation cost, and sometimes even impossible for huge networks with millions of nodes and edges. Hence, computation overhead becomes a big challenge. Third, besides uncertainty, deciding relationships among nodes itself is a challenging problem. Since different applications have various demands, the relationships among nodes are affected by many factors, which are quite hard to identify relevance among nodes. In real applications, especially in social networks, a relationship is not only reflected by the link between a pair of nodes but also affected by a node's neighbors. If two nodes have many common neighbors, there might be a more strong relationship between them even if there is no direct link exists between them. Therefore, we should take the common neighbors into consideration.

Facing the aforementioned challenges, this chapter has the following contributions. First, one effective way for modeling uncertainty is to approximate the dynamic feature of a network by a static model endowed with some additional features. Therefore, we propose two basic models to describe uncertainty in dynamic networks for different applications. In these two models, historical information is utilized to predict future relationships. Second, considering the expensive cost of managing and mining uncertainty, we employ the sampling technique to take care of uncertain possible worlds. Furthermore, the Chernoff bound and

[1] `http://string-db.org/`

the Hoeffding inequality are used to guarantee the accuracy of the obtained results. Last but not the least, we design a method for relationship detection in uncertain networks. The entities in a same community or group with relationship usually interact frequently, share similar properties and generate common features. In our solution, two-hop expectation distance is adopted to approximate the expected number of common neighbors. This method can also serve as a framework for measuring the expected number of common neighbors in uncertain graphs. Some existing community detection networks clustering and other algorithms designed for certain graphs can then be employed in uncertain graphs based on our framework.

The rest of this chapter is organized as follows. Section 3.2 presents the preliminaries and problem definition. Section 3.3 illustrates the sampling scheme and theoretical analysis. Evaluation results based on real and synthetic data sets are shown in Section 3.4. Section 3.5 concludes our chapter.

3.2 Data Model and Problem Definition

In this section, we formally present the data models considered in this chapter. Similar to deterministic graphs, uncertain graphs may be undirected or directed and carry additional labels on edges. For simplicity and clarity, we consider undirected simple uncertain graphs. However, our discussion can be extended to directed graphs straightforwardly. We assume the edges in a graph are independent, which is common in real applications.

Topology changes mainly result in uncertainty in a network. We propose the following four different basic uncertainty models to reflect topology changes. We assign a weight to each snapshot G^i $(0 \leq i \leq t)$, and we design different weight assignment scenarios for different models.

(M 1) **Constant model**. This model is used for the case where all the snapshots have the same impact on network uncertainty. Therefore, we assign a same weight to each snapshot. As shown in Fig.7.1, $f_1(G^i)$ assigns constant weight c to each snapshot.

(M 2) **Linear model**. This model is used for the case where the snapshots have a linear

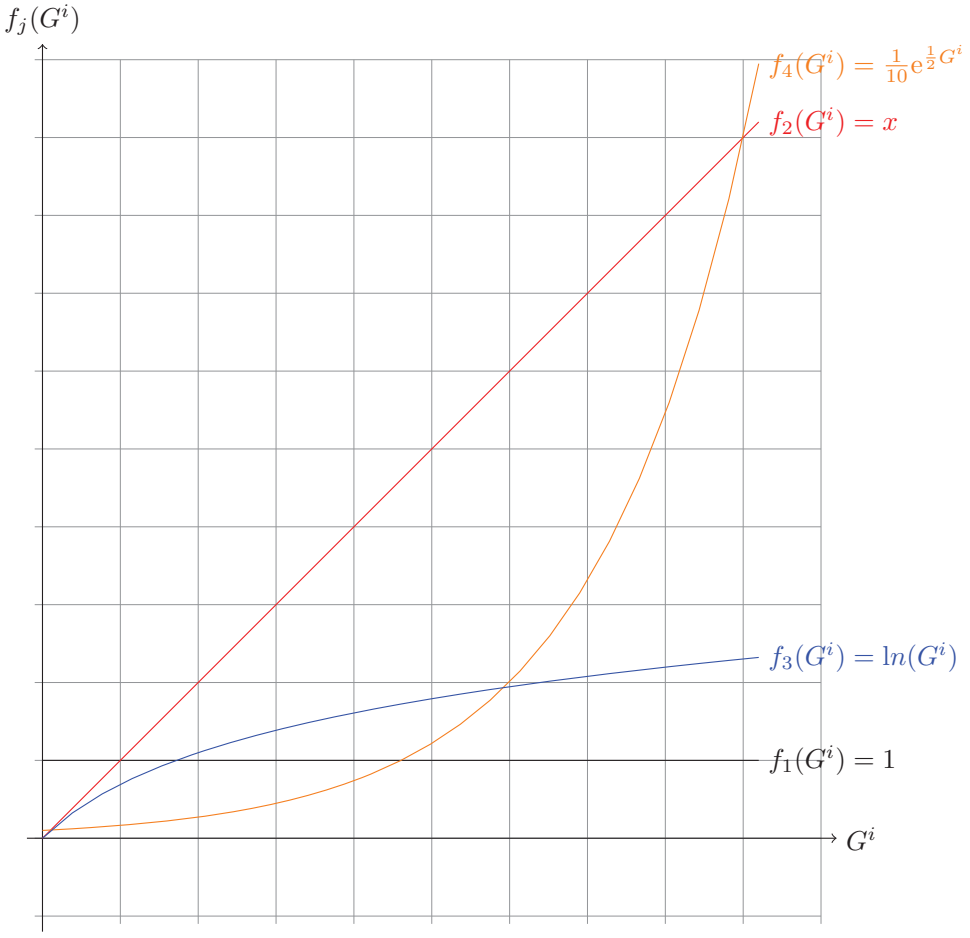


Figure 3.1. Weight assignment functions for snapshot G^i

changing pattern over time. Therefore, we employ a linear weight assignment scheme for this model. As shown in Fig.7.1, $f_2(G^i)$ is a linear function.

(M 3) **Log model**. This model is used for the case where the snapshots have a logarithm changing pattern over time. Therefore, we adopt a logarithm weight assignment scheme for this model. As shown in Fig.7.1, $f_3(G^i)$ is a logarithm function.

(M 4) **Exponential model**. This model is used for the case where the snapshots have an exponential changing pattern with time. Therefore, we use an exponential weight assignment scheme for this model. As shown in Fig. 7.1, $f_4(G^i)$ is an exponential function.

For Model j, the existence probability assigned to edge e is calculated as following:

$$Pr(e) = \frac{\sum_{i=0}^{t} e^i f_j(G^i)}{\sum_{i=0}^{t} f_j(G^i)} \tag{3.1}$$

where e^i represents the existence of edge e in the graph G^i.

Definition 2: An uncertain graph is represented by $\mathcal{G} = (V,\ E,\ \Sigma,\ l,\ p)$, where V is the vertex set, $E \subseteq V \times V$ is the set of edges, Σ is a set of labels, $l : V \to \Sigma$ is a function assigning labels to the vertices, and $p : E \to (0, 1]$ is the function assigning each edge $e \in E$ a probability $Pr(e)$ obtained from Equation (3.1).

Let I be a possible world instance which is a deterministic graph. As shown below, $\mathcal{G} \Rightarrow I$ denotes that I can be generated from $\mathcal{G}$, and the probability of such a derivation is $Pr(\mathcal{G} \Rightarrow I)$.

$$Pr(\mathcal{G} \Rightarrow I) = \prod_{e \in E(I)} p(e) \prod_{e \in E(I)) \setminus E(\mathcal{G})} (1 - p(e)) \tag{3.2}$$

Example 1: Fig. 3.2 shows an example uncertain graph $\mathcal{G}$. The number marked on each edge e denotes $Pr(e)$. For $\mathcal{G}$, there are $2^{|E|}$ possible worlds $I_i (1 \leq i \leq 2^{|E|})$. In this example, there are $2^{|3|} = 8$ possible worlds. Each deterministic graph can be viewed as a special uncertain graph in which the existence probability of every edge is 1.

The relationship between a pair of nodes can be determined by the number of their common neighbors to a considerable degree. Therefore, a reasonable model for describing

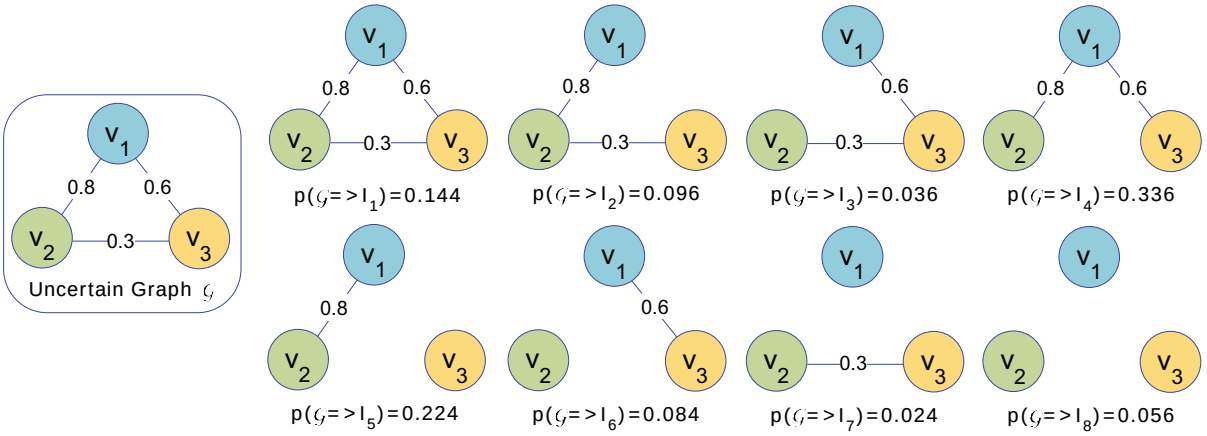


Figure 3.2. Derivation of possible worlds I_i for uncertain graph $\mathcal{G}$

common neighbors in uncertain graphs is expected. Let $N(v)$ be the neighbor set of vertex v. The *Jaccard index* in deterministic graphs which is defined as $\frac{N(v_i) \cap N(v_j)}{N(v_i) \cup N(v_j)}$ ranges from 0 (no overlap in the neighborhoods of v_i and v_j) to 1 (the neighborhoods of v_i and v_j are identical). However, for uncertain graphs, it is impossible to derive such an index, since the relationship between every pair of nodes is imprecise.

Definition 3: To measure the distance between two vertices u and v in an uncertain graph, an ***E**xpected **n**eighbor **distance***, $Endistance(u, v)$ is defined as the expectation of the *Jaccard index*.

$$Endistance(u, v) = Exp(\frac{N(v_i) \cap N(v_j)}{N(v_i) \cup N(v_j)}) \tag{3.3}$$

where the function $Exp(X)$ is the expectation of variable X.

From the above definitions, our investigated problem is defined as following.

Input: A dynamic graph $\mathbb{G} = (G^0, G^1, \ldots, G^t)$, model type j, sampling parameters ϵ and δ which guarantee the accuracy, and constant k which controls the number of the communities.

Output: k communities in $\mathbb{G}$ which have the top k highest probabilities.

3.3 Algorithm Framework and Theoretical Analysis

In this section, we present the framework of our algorithm. The first step is to construct an uncertain network based on dynamic snapshots; then, considering common neighbors, we introduce a method to measure relationships among nodes; third, an effective sampling scheme is introduced with some optimization strategies and complete theoretical analysis is presented to guarantee correctness and efficiency.

3.3.1 Construct an Uncertain Network

Algorithm 1: Constructing an Uncertain Graph

input : A set of snapshots in dynamic graph $\mathbb{G} = (G^0, G^1, \ldots, G^t)$, the model type j
output: Uncertain graph $\mathcal{G}$

```
1 for i = 1; i ≤ t; i++, E ∈ G do
2     for each edge e do
3         if e^i ∈ G^i then
4             Numerator += (f_j(e^i));
5         Denominator) += (f_j(e^i));
6     Pr(E) = Numerator/Denominator;
```

According to model type j, use weight assignment function $f_j(G^i)$ to assign weight to each $(G^0, G^1, \ldots, G^t) \in \mathbb{G}$. In the existing uncertain models, it is generally assumed that uncertainty exists in networks. The way they present uncertainty is to associate a random number between 0 to 1 to each node and/or each edge. However, except a real uncertain network PPI in Biology whose probability is determined by biology experiment, there is still not any other more reasonable model or method to present uncertainty in networks. Since the dynamic feature is one of the most important reasons resulting in uncertainty, we model uncertainty through several network snapshots coming from a dynamic network. Algorithm 1 presents the process of constructing an uncertain network.

Since the required data source to construct an uncertain network is a set of snapshots of a dynamic network, the topology is changing at any time. The nodes may disappear or

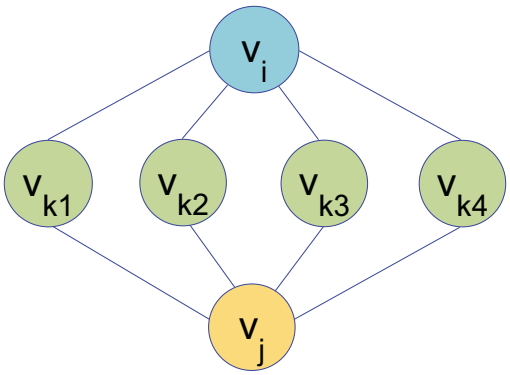


Figure 3.3. Common neighbors in a community

appear, we suppose the node set includes all the nodes ever appeared in a network. If node i disappears in one snapshot, we set the weights of all the edges connected to i to 0. The computational cost of Algorithm 1 is $O(t*n^2)$ where n is the number of the nodes in network and t is the number of snapshots.

3.3.2 Measuring Relationships Among Nodes in an Uncertain Network

The number of common neighbors is one of the most important measurements for relationships among nodes. On the one hand, common neighbors stand for direct relationships among nodes, since if there is an edge between node i and node j, they are also common neighbors of each other (each node neighbor set includes itself). On the other hand, the number of common neighbors can also describe indirect relationships within a community. However, in an uncertain network, the concept of common neighbors is difficult to define since the direct relationship between a pair of nodes is not clear. Researchers use the expectation weight of one edge to measure the direct connection between two nodes. Similarly, we use the expected number of common neighbors to represent a relationship.

In an uncertain graph $\mathcal{G}$, the expected number of common neighbors between node v_i and node v_j can be calculated by the expectation of the number of distinct 2-hop paths between them.

In a deterministic graph, for node v_i and node v_j, the number of common neighbors

equals to the number of distinct 2-hop paths between them. As shown in Fig. 3.3, there are four nodes (v_{k1}, v_{k2}, v_{k3}, v_{k4}) between v_i and v_j. Obviously, there are also four distinct 2-hop paths between them correspondingly. Apparently the number of distinct 2-hop paths and the number of common neighbors is a one-one correspondence. Then, we can have a deterministic graph. For v_i and v_j, a new distinct 2-hop path means adding a new node v_k as a connector between them, and v_k belongs to both $N(v_i)$ and $N(v_j)$, where v_k is the common neighbor of v_i and v_j.

Obviously, in a deterministic graph, the number of common neighbors between two nodes corresponds to the number of 2-hop distinct paths between them. Since the expected number of common neighbors cannot be calculated directly, we use the number of 2-hop distinct paths to represent it. In an uncertain graph $\mathcal{G}$, a 2-hop path is a one existing in some of the possible worlds generated from $\mathcal{G}$. We cannot derive whether there is a 2-hop path or not; however, we can obtain the expected existence possibility of a path according to its existence situation in each possible world.

Lemma 1. *In uncertain graph $\mathcal{G}$, the expected number of union neighbors between node v_i and node v_j can be calculated as follows.*

$$Exp(|N(v_i) \cup N(v_j)|) = Exp(|N(v_i)|) + Exp(|N(v_i)|) - Exp(|N(v_i) \cap N(v_j)|) \tag{3.4}$$

Proof. : This is a simple application of the set theory. □

Theorem 1. *In uncertain graph $\mathcal{G}$, the expectation of the Jaccard index $\frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|}$ can be calculated by $\frac{ExpDCount_{Path2}(v_i, v_j)}{Exp(|N(v_i) \cup N(v_j)|)}$, where the numerator part $ExpDCount_{Path2}(v_i, v_j)$ is the number of distinct 2-hop paths between node v_i and node v_j, and the fraction denominator part $Exp(|N(v_i) \cup N(v_j)|)$ is the expected size of the union set of the two neighbor sets.*

Proof. : Theorem 1 is obviously true according to Lemma 1. □

3.3.3 Sampling Possible Worlds

As mentioned, to derive $Endistance(u, v)$ in an uncertain graph, we need to enumerate all the possible worlds to calculate the expected number of 2-hop distinct paths, the expected number of common neighbor and the expected size of the union set of the two neighbor sets.

The computation overhead of enumerating is very expensive. To solve this problem, we utilize a random sampling method to acquire some possible worlds of an uncertain graph. The following lemma demonstrates how to calculate an edge's existence probability.

Theorem 2. .

Fact 1. *For a one-edge subgraph of an uncertain graph, the expected existence probability of this edge is equal to the edge's existence probability:*

$$Exp(e') = p(e') \tag{3.5}$$

Fact 2. *For a subgraph with more than one edge, its expected existence probability need to be calculated based on the possible worlds.*

Proof. : If P is the probability distribution of uncertain graph $\mathcal{G}$, then we have Equation (3.6), where I_i is the ith instance of all the 2^m possible worlds, and the existence probability of I_i is shown in (3.7).

$$Exp(\mathcal{G}) = \sum_{i=1}^{i=2^m} p(I_i) \tag{3.6}$$

$$p(I_i) = \prod_{e=(u,v)\in E(I_i)} p(e) \prod_{e=(u,v)\in (E(\mathcal{G})\setminus E(I_i))} (1 - p(e)) \tag{3.7}$$

According to Equation (3.7), for edge e' not showing up in a possible world I_i, the existence probability of e' in I_i should be 0.

$$p(I') = \sum_{I'\in I_i, i=1}^{i=2^m} p(I_i) = \sum_{I'\in I_i, i=1}^{i=2^m} \prod_{e\in E(I_i)} p(e) \prod_{e\in (E(\mathcal{G})\setminus E(I_i))} (1 - p(e)) \tag{3.8}$$

Only when $E(\mathcal{G}) \setminus E(I_i)$ is empty, we have

$$p(I') = \prod_{e \in E(I')} p(e) \tag{3.9}$$

In fact, Equation (3.9) shows the existence probability of a deterministic graph which has all the edges in I'. As shown in Equation (3.6), the existence probability of subgraph I' is the summation of the existence probability of each possible world which includes I'. According to Equation (3.2), we obtain Equation (3.6). Since the existence probability summation of all the possible worlds equals to 1 [181], this fact is true. □

Fact 2 told us that we can not get the exact expected existence probability without enumerate all the possible worlds.

To enumerate all the possible worlds generated from an uncertain graph $\mathcal{G}$ is a #P-complete problem [181]. According to this fact, we cannot enumerate all the possible worlds to calculate $Endistance(u, v)$ in an uncertain graph. We need to adopt some other more effective techniques. In this chapter, we apply a sampling method to estimate $Endistance(u, v)$.

Now we introduce how to perform sampling of the possible worlds which follow a bernoulli distribution. Each edge in an uncertain graph either exists in a possible world with probability 1 or not shows up at all. Consider $\mathcal{G} = (V, E, \Sigma, l, p)$ with n nodes. ϵ and δ are accuracy parameters where ϵ ($\epsilon \geq 0$) and δ ($0 \leq \delta \leq 1$) denote the upper bound and relative error respectively, and both of them can be arbitrarily small. The parameter r denotes the number of possible worlds. Let I^i, $1 \leq i \leq r$, be a set of sampled graphs under distribution P where all $\{I^i\}_p \in Imp(\mathcal{G})$, where $Imp(\mathcal{G})$ is a generated implication subspace.

The Chernoff bound gives exponentially decreasing bounds on tail distributions of sums of independent random variables [37]. We employ the Chernoff bound to reduce the number of the sampled possible worlds while guaranteeing accuracy.

Lemma 2. *Given a pair of vertices (u, v), set X_i to be equal to 1 if there exists at least one 2-hop path from u to v in graph I^i, and 0 otherwise. $p_2(u, v)$ is the existence probability of a 2-hop path between u and v. According to the Chernoff bound, we get $P(|\frac{1}{r}\sum_{i=1}^{r} X_i - p_2(u, v)| \geq$*

$\epsilon p_2(u,v)) \leq 2exp(\frac{r \cdot p_2(u,v)\epsilon^2}{3})$. *If the number of sampled possible worlds* $r \geq \frac{3}{\epsilon^2 p_2(u,v))} \ln(\frac{2}{\delta})$, *we have*

$$P(|\frac{1}{r}\sum_{i=1}^{r} X_i - p_2(u,v)| \geq \epsilon p_2(u,v)) \leq \delta \tag{3.10}$$

The Hoeffding's inequality provides a method to bound the upper bound of the probability of the sum of random variables deviating from its expected value [78]. We employ the Hoeffding's inequality to further reduce the number of the sampled possible worlds while guaranteeing accuracy.

Lemma 3. *Let* d_i *denote the number of 2-hop paths between* u *and* v *corresponding to a possible world* I^i. *The random variables* d_i *are bounded in the range [0,n]. Exp*(d) *is the summation of estimated expectation value according to the sampling subspace. Based on Hoeffding's inequality,* $P(|\sum_{i=1}^{r} d_i - Exp(d)| \geq \epsilon) \leq 2exp(\frac{2\epsilon^2}{r(n-1)^2})$. *If* $r \geq \frac{(n-1)^2}{2\epsilon^2} \ln(\frac{2}{\delta})$, *we have*

$$P(|\sum_{i=1}^{r} d_i - Exp(d)| \geq \epsilon) \leq \delta \tag{3.11}$$

Theorem 3. *Let* $p(u,v)$ *be the probability distribution function respond to the distance of path between* u *and* v *in uncertain graph* $\mathcal{G}$. *Let* $\widehat{p_2(u,v)}$ *be the independent bernoulli distribution random variables as the estimator of* $p_2(u,v)$ *come from sampling subspace in* $\{G^i\}_p$. *It is easily to prove* $\widehat{p_2(u,v)}$ *is an unbiased estimator of* $p_2(u,v)$ [119]. *If we sample at least* $r = max(\frac{(n-1)^2}{p}2\epsilon^2, \frac{6}{\epsilon^2}) \ln(\frac{2}{\delta})$ *possible worlds, we can guarantee that Equation* (3.10) *and Equation* (3.11).

Proof. : Theorem 3 is the simple application of Lemma 2 and Lemma 3. And we change the bound from $r \geq \frac{3}{\epsilon^2 p(u,v)} \ln(\frac{2}{\delta})$ to $r \geq \frac{6}{\epsilon^2} \ln(\frac{2}{\delta})$ since the only attention we need to pay to is the existence probability larger than $\frac{1}{2}$. □

It is easy to prove (omitted) that our sampling estimator is unbiased, which means the expectation of sampling is equal to the result of sampling space. The antithetic variable which is an unbiased estimator for sampling can result in better efficiency and sampling

variance. The sampling error for parameter $\theta \approx \bar{X}$ depends on the Mean Squared Error (MSE)=$Var(\bar{X})$=$Var(X)/n$, then the sampling could be more efficient when $Var(X)$ is reduced [119].

Theorem 4. *Using antithetic variables can reduce variance in our sampling process.*

Proof. : Suppose In_1 and In_2 are independent and identically distributed random variables with mean θ, then

$$Var(\frac{In_1 + In_2}{2}) = \frac{1}{4}(Var(In_1) + Var(In_1) + 2Cov(In_1, In_2)) \tag{3.12}$$

If we make $Cov(In_1, In_2) \leq 0$, then the variance is reduced. For many sampling methods, a θ estimator is $In_1 = g(U_1, \ldots, U_n)$ for some monotonously increasing function g. Considering the antithetic estimator $In_2 = g(1 - U_1, \ldots, 1 - U_n)$, the combined estimator would be $(In_1 + In_2)/2$. We can get $Cov(h(U_1, \ldots, U_n)$, where $h(1 - U_1, \ldots, 1 - U_n)) \leq 0$ [119].

Considering the sampling process in our algorithm, it is easy to see that the more edges we enumerate the higher probability we can obtain a 2-hop path. Let abstract function g be an $|E|$-dimension random variable. Each variable indicates whether edge e exists. If the edge e exists, then the function g will increase, so this function is monotonously increasing apparently. □

We employ matrix techniques to identify 2-hop path between each pair of nodes. We use a matrix to represent the connectivity information for a network as shown below.

If the multiply two copies of such a matrix, each 1 in the result matrix means there is at least one 2-hop path between the corresponding pair of nodes(Equ. 3.13).

Since a big network may have millions of nodes, it is hard to carry out matrix calculations in one pass. We divide a big matrix into different parts and apply the calculation respectively. Then, the final result comes from the combination of all the parts. In this step, we can significantly reduce the computation cost of finding 2-hop paths in a large network. In this

way, we only need to take care of the pairs of nodes whose corresponding entry in the result matrix is not 0.

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 1 & 0 & 1 \\ 1 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 1 & \cdots & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 1 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 1 & 1 & \cdots & 1 & 0 & 1 \\ 1 & 1 & 0 & \cdots & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 & \cdots & 1 & 0 & 1 \\ 1 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & 1 & \cdots & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 1 & 1 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 1 & 1 & \cdots & 1 & 0 & 1 \\ 1 & 1 & 0 & \cdots & 1 & 1 & 1 \end{bmatrix} \tag{3.13}$$

Algorithm 2: Strassen Algorithm in our Problem

input : Matrix M for graph $\mathcal{GM}$
output: Result of matrix M'

```
for i = 1; i! = m; i++ do
    M'_ij = 0;
    for j = 1; j! = m; j++ do
        M'_ij = M'_ij + M_ik M_kj;
return M';
```

We can even apply the Strassen algorithm 2 [130] to further reduce the computation cost. The standard matrix multiplication takes approximately $2N^3$ (where $N = 2^n$) arithmetic operations including additions and multiplications, and the asymptotic complexity is $O(N^3)$. The asymptotic complexity of the Strassen algorithm is $O([7 + o(1)]) = O(N^{\lg_2^7 + o(1)}) \approx O(N^2.8074))$. Based on the result matrix, we can apply our sampling algorithm to derive a weighted graph upon which community detection, clustering or other social problem algorithms can be run.

The computational cost of Algorithm 3 is $O(r * n^2)$, where n is the size of network's node set $|n = V(G_w)|$ and r is the number of possible worlds which are enumerated.

Algorithm 3: Sampling Algorithm

input : Sampling parameters ϵ and δ, uncertain graph $\mathcal{G}$
output: Weighted graph G_w

```
According to ε and δ, calculate r;
for i = 1; i ≤ r; i++ do
    for j = 1; j! = n; j++ do
        for j = 1; j! = n; j++ do
            if M'_jk != 0 then
                Calculate Endistance(j, k) ∈ G to construct G_w;
return G_w;
```

Table 3.1. Dataset for influence initialization

Data	Nodes	Edges	Diameter
Amazon0302 (A02)	262111	1234877	29
Amazon0312 (A12)	400727	3200440	18
Amazon0505 (A05)	410236	3356824	21
Amazon0601 (A01)	403394	3387388	21

3.4 Experimental Study

In this section,we evaluate our algorithms on the aspects of quality and efficiency. All the experiments were performed on a desktop computer with Inter(R) Core(TM)2 Quad CPU 2.83GHz and 4GB RAM. We implemented all the algorithms based on BGL which is a sub library of Standard Template Library (STL)[2]. One typical dataset from SNAP (Stanford Large Network Dataset Collection) [3] was used. Table 3.1 and Table 3.2 show the information of our two datasets. Each row in the tables represent a network snapshot.

In Fig. 3.4, we compute the average degree of each snapshot in the second dataset. Compare to four different uncertainty generation models (EAD(M1) to EAD(M4)), we can easily get the expectation of each snapshot, which is very different from the general average degree. This demonstrate that uncertainty is a very important feature which can effect the knowledge inherent the network.

We evaluate how our proposed models can depict the dynamic evolvement of a network

[2] `http://www.boost.org`
[3] `http://snap.stanford.edu/data/`

Table 3.2. Gnutella dataset

Data	Nodes	Edges	Diameter
p2p-Gnutella04 (p04)	10876	39994	9
p2p-Gnutella05 (p05)	8846	31839	9
p2p-Gnutella06 (p06)	8717	31525	9
p2p-Gnutella08 (p08)	6301	20777	9
p2p-Gnutella09 (p09)	8114	26031	9
p2p-Gnutella24 (p24)	26518	65369	10
p2p-Gnutella25 (p25)	22687	54705	11
p2p-Gnutella30 (p30)	36682	88328	10
p2p-Gnutella31 (p31)	62586	147892	11

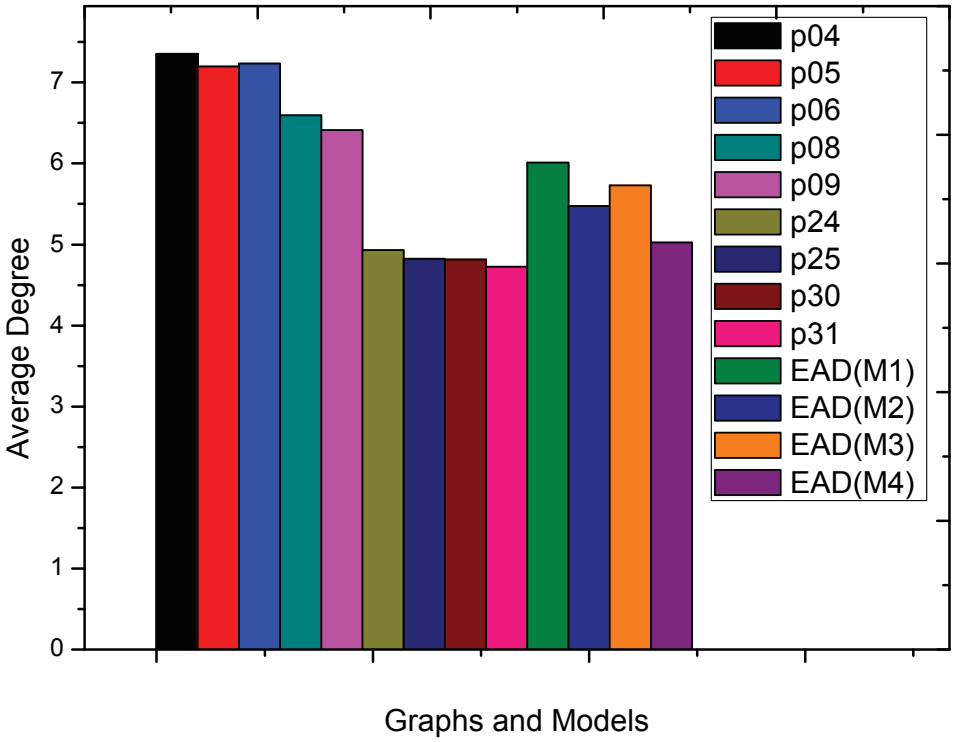


Figure 3.4. Average degree VS expect degree

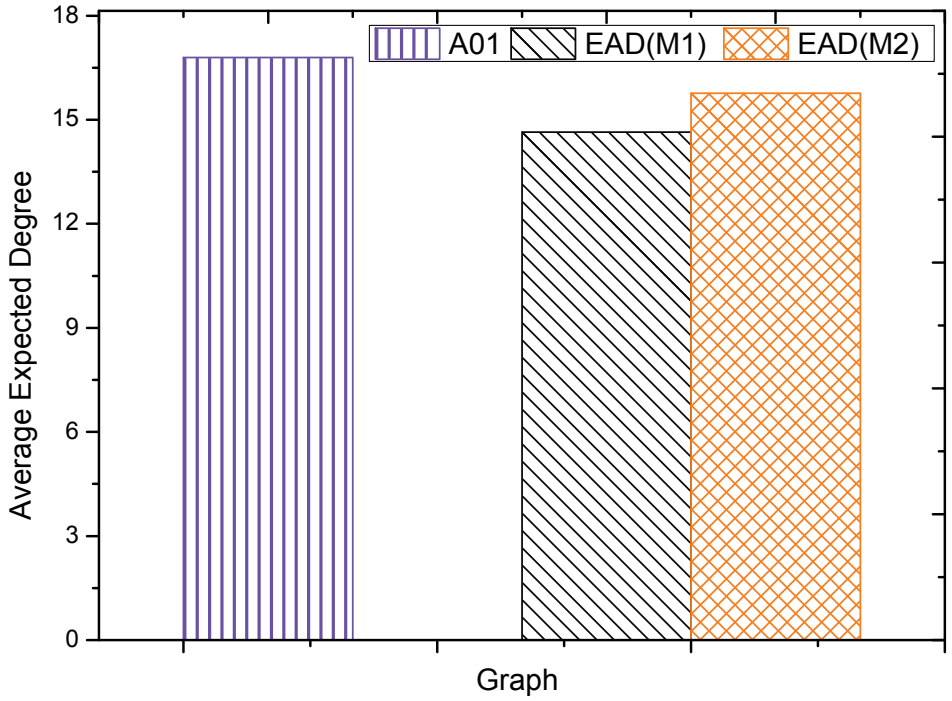
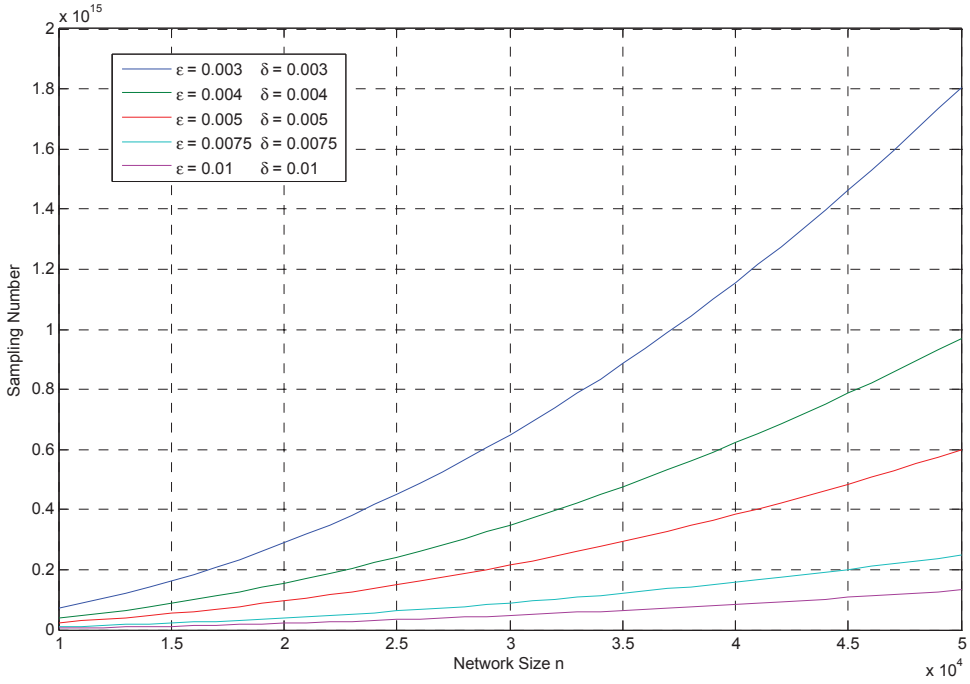


Figure 3.5. Effectiveness of models evaluation

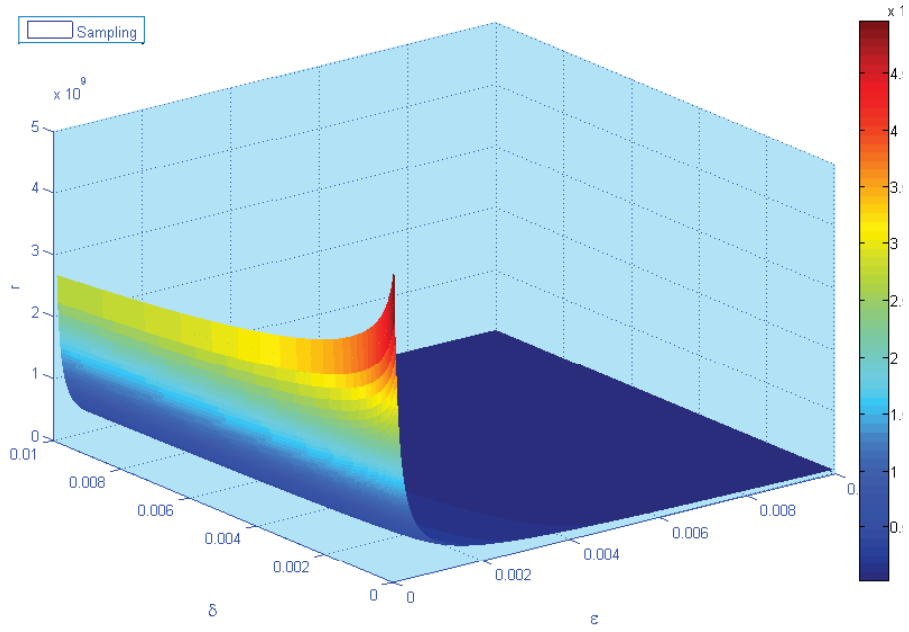
based on history network information. We use the first three network snapshots in Table 7.3 as history information, and employ our M1 and M2 models to generate corresponding uncertain networks. Fig. 3.4 shows the average degrees of the $4th$ network snapshot in Table 7.3, the uncertain network generated by M1 and the uncertain network generated by M2. Apparently, the generated uncertain networks by M1 and M2 are almost in accordance with the $4th$ network snapshot. Because the $1st$ input network snapshot is very different from the other two, there is a little difference between the generated uncertain networks and the $4th$ network snapshot. M1 focuses on the overall history, while M2 considers more about the most recent situation. This is why the uncertain network generated by M2 is more proximate to the $4th$ network snapshot.

In Fig. 3.6(a) the different size of networks are selected and the network in Fig. 3.6(b) is fixed. We configure parameters (ϵ , δ) on different size of networks and compare the sampling number of possible worlds to show the stability of our algorithm in Fig. 3.6(a).

Next, we evaluate the quality of our sampling algorithm in terms of correctness and



(a)



(b)

Figure 3.6. Effect on parameters (ϵ , δ)

the size of the sampling space. Base on Theorem 3, ϵ reflects the upper bound of the difference between the sampling result and the expected result, and δ denotes the relative error of our sampling result. Fig. 3.6(b) illustrates the different settings of ϵ and δ, and the resultant sampling number. In possible worlds model, the whole sampling space is 2^{5000} which even cannot be accepted directly by a typical computer. However, even if the user has an extremely strict requirement of correctness, for example $\epsilon = 0.005$ and $\delta = 0.005$, the sampling number is $5.98 * 10^{12}$. Eventually, if the sampling number is only $1.4 * 10^{10}$, it still can be guaranteed that δ is less than 0.08 and ϵ is less than 0.05.

3.5 Summary

The importance of uncertainty in networks has been recognized in many application areas, such as social networks, wireless networks and PPI networks. In this chapter, we present a framework for generating uncertain networks based on historical network snapshots. Two uncertainty construction models are presented to capture uncertainty from dynamic snapshots, and sampling techniques are also employed to improve the efficiency of the algorithm. To describe the relationship in uncertain networks in a more practical way, 2-hop expectation distance are adopted to approximate the expected number of common neighbors. Both the theoretical analysis and our experiments demonstrate the effectiveness and efficiency of our proposed methods.

Chapter 4

DYNAMIC PROBING FOR INFLUENCE

4.1 Introduction

Social influence is the phenomenon that one's emotions, opinions, or behaviors can induce his or her friends to behave in a similar way. This phenomenon has become a popular topic recently due to its unlimited potential in business and marketing [64, 75, 125, 23, 149, 145, 144, 45, 147, 62]. For instance, The Alibaba Group, one of China's e-commerce giants, reported that the value of its online transactions during China's unofficial "Singles' Day" holiday had hit $9.3 billion. On the same day in the previous year, the total purchases through Alibaba was $5.75 billion[1]. While in E-commerce environments (also known as one type of Online Social Networks (*OSNs*)), strong virtual relationships existing among users construct the basic structure in modern social life. Substantial commercial opportunities are coming with lots of social influence based applications. From this aspect, how to maximize the influence on potential customers is one of the most important keys to open the door of the unlimited business opportunities, which drives the extensive investigation on the influence maximization in social networks.

Influence maximization is based on trust among individuals' within close social circle, such as families, friends, and co-workers. Word-of-mouth or viral marketing differentiates itself from other marketing strategies and is widely supported by lots of research and industrial applications. Besides the strategy of distributing promoted samples, maximization of influence on the social network could also be applied to other similar services not only limited by trial devices but also include marketing for releasing new music (Spotify.com), cloud computing resource (AWS from Amazon.com), and even financial services (www.usaa.com/) *etc.* Companies, such as Spotify, Amazon, or USAA, they mainly promote their new services

[1] http://www.nytimes.com/

through distributing emails with a special offer to solicit new customers. An important issue of these social networks is that they are dynamic and community-based. Applying our algorithm could help them to find out who are the optimal target groups that the emails or advertisements should be geared towards.

As one of the most popular research topics in social networks, many existing literatures focusing on the influence maximization in social networks can be found. For example, [88, 95] proposed several models to simulate the influence diffusion process. However, influence maximization in social networks is still an developing problem. Due to the complexity and diversity of the phenomena, researchers are still facing a lot of challenges concerning how to analyze and utilize the influence in *OSNs*. First, OSNs are dynamic networks. Therefore, the change of network topology directly affects the diffusion of influence. In addition, almost all of the classical models, such as IC and LT together with their many derived varieties involving Monte Carlo simulation, which are incredibly resource intensive. The basic idea of both IC and LT is to find a subset $S^* \in V$ such that $|S^*| = k$ and $\delta(S^*) = \max\{\delta(S)||S| = k, S \in V\}$, where the function $\delta(\dot{})$ is the information propagation strategy. Since the influence maximization problem is unfortunately *NP-hard*, it is impossible to find the most influential nodes in a network. It is even more difficult to pursue the optimal node sets that can maximize the influence in a dynamic social network. Besides all the other challenges, updating a network to reflect its dynamic nature with time is extremely resource consuming in large social networks. Therefore, in this chapter, we are interested in proposing an efficient integrated solution to select the most influential nodes in dynamic social networks considering the challenges and features of OSNs.

After a comprehensive background research from real OSNs, we summarize the following facts:

1. First of all, communication is one of the essential and distinctive features of social networks. Because the nature of a dynamic network concretely comes from each node in each community, and the final influence effect could also be evaluated by nodes in each community.

2. In a whole network, dynamics do not exist in every corner. It mainly comes from several parts of the network instead. The parts commonly reflect in the unit of a community.

3. Detecting significant communities which reflect the change of a network and performing selective updates could save a plenty of computation overhead.

Fig.4.1 shows an example of the influence diffusion in a dynamic social network. The top figure presents the network at time $t = 0$; the middle figure describes the changing of the network at time $t = 1$; and the bottom figure is the network topology at the end $t = 2$. From left to right, the network is divided into three communities. Through the time flow from top to bottom, we could get that only the two communities (the left and the right in the dashed line frames) have changed their topology. But the nodes in the middle part remain unchanged. From this example, we notice that it would be more efficient if we could identify the two changed communities but ignore the middle community during updating. Therefore, probing the most active communities to approximate the global evolution of the network would be a very effective solution. Most existing research surprisingly ignores the advantages of the community feature. On the other hand, the complexity of solving the influence maximization problem rapidly increases with the size of the network. Therefore, finding influential local nodes in each relatively smaller community could be much more efficient.

It's worth pointing out that the objective of our work is to track the network's global dynamics as well as reducing the cost brought by frequently updating the whole network. We utilize the "community" instead of "node" as a unit to probe the change of the network because the community is the basic and natural structure in large networks, which is a better choice compared with the node. Even though the updating unit is the community, the changing of nodes and links among nodes in the communities are more commonly the updating targets. For each iteration, based on our theoretical analysis, when b communities are selected to be actually updated, the nodes and the links among nodes in the selected communities are going to be updated. The reason we do not take a node as the unit to

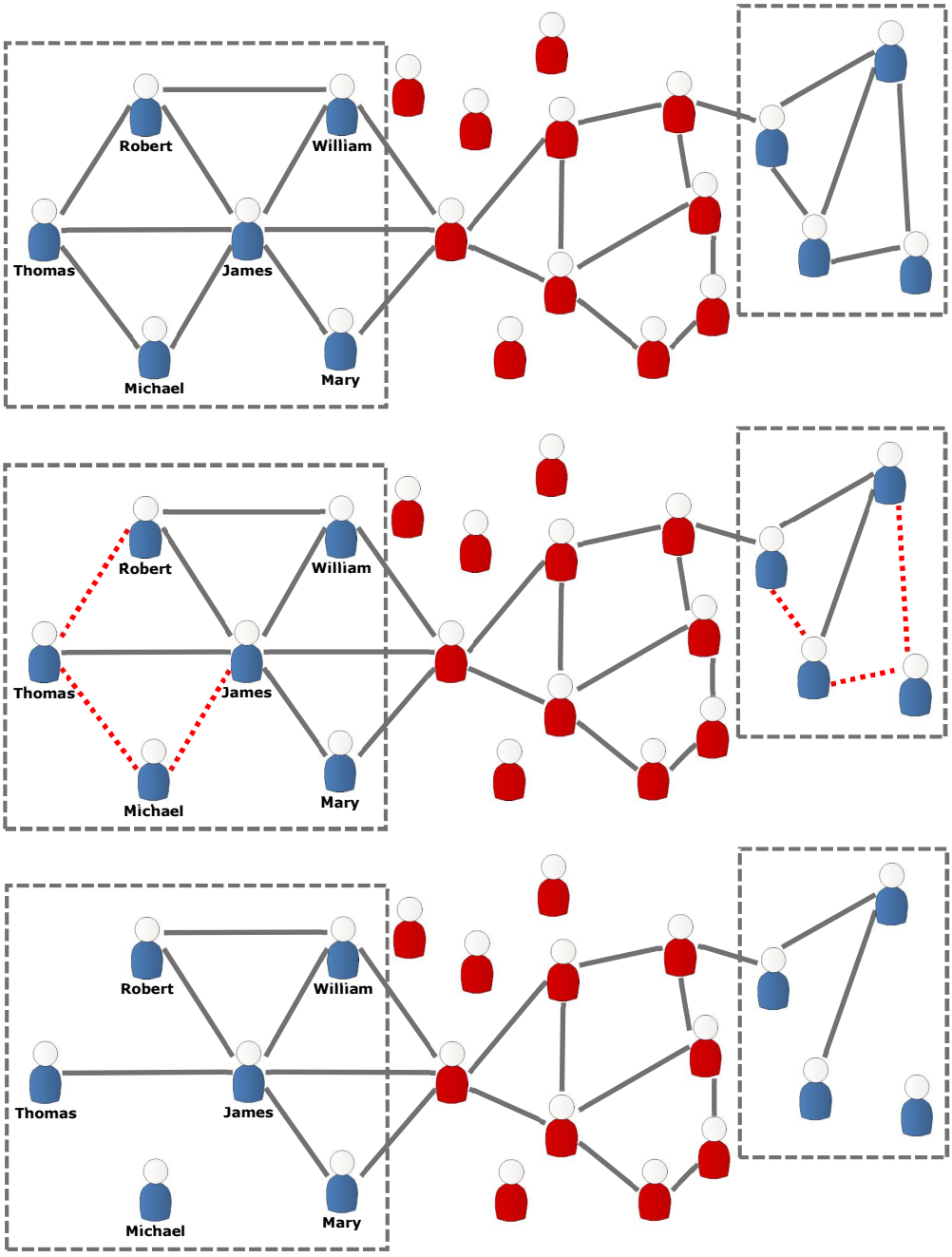


Figure 4.1. Probing community for dynamic network

update the network is: from an overall perspective, even the changing of only one node in a network will only result in several relationships changing. Frequently updating the network node by node will bring in more redundant costs because of updating their neighbors. On the other hand, a community will cover several nodes with closer relationships. The observations in previous results [180] show that most of the dynamics in large networks have some kind of local effect [120] confirming the advantage of communities over nodes. The local update of communities could update the dynamic changing within specific areas.

Again, considering the node "Michael" in the left frame, from time $t = 0$ to $t = 2$, "Michael" disappears from the original topology. In this case, we could consider it as node "Michael" has been excluded from the network at time $t = 0$, which is the opposite process of a new node joining. Our algorithm considers the changing of each node within the selected communities. Both new nodes joining and leaving are considered in the algorithm. Thus, our algorithm considers the community as a unit to probe the dynamic of networks from a global sense but the dynamics of nodes is the actual cell being updated.

Although it is not always the case that one influential node in a community corresponds to an influential node in the whole network, apparently an influential node in one community has a stronger influence based on its degree and neighbors' density compared with normal nodes in the whole network. The remaining challenge is how to fill the gap between the local influential nodes in a community and the global influential nodes in a social network. In [150], the authors try to reduce the global computational cost of influence maximization algorithms by splitting the computation cost to many smaller communities. Inspired by [150], we take full advantage of the divide-and-conquer strategy with many differences and improvements. First, the network topology considered in our work is dynamic. Second, the output of our solution could be the maximal influential node set with a specific time stamp instead of only one influential node set in a static social network. From this point, our solution is a superset of [150]. Third, in [150], the considered diffusion speed depends on a static network topology which is not a general scenario in social networks. Finally, we propose innovative approaches other than using the traditional straightforward method.

Our contributions are summarized as follows:

1. We formulate and model the influence maximization problem in a dynamic social network. The model explores the potential of employing a community-based algorithm to achieve the influence maximization in dynamic OSNs.

2. We propose an effective algorithm by probing only part of communities instead of the whole network in each time stamp to save computation cost. Meanwhile, we provide theoretical analysis for the error bound of the influence diffusion under the defined model.

3. Based on the network structure derived from the dynamic environment, we develop a community-based dynamic programming algorithm together with multiple optimization techniques of partition and combination processes to mine the influential nodes.

4. Last but not the least, our experimental results on several real data sets show that in both synthetic and real large networks, our solution clearly improves practicality and accuracy against several existing algorithms.

The rest of the chapter is organized as follows. Section 4.2 presents the preliminaries and problem definition. Then we introduce our proposed model with analysis and the algorithm in Section 4.3. The evaluation results based on synthetic and real data sets are shown in Section 4.4. Section 4.5 concludes the chapter.

4.2 Preliminaries and Problem Definition

We first introduce the probing technique for a dynamic network. The motivation behind it is that it is almost unrealistic to observe the whole dynamic network at any moment, but it is much easier to monitor the changes of some parts of the network. Initially, the entire network can be represented as $G = (V, E, W)$, where V is the node set, $E = V \times V$ is the edge set, and W is the weight set for edges representing the influence probability among

users for a given diffusion process. Let $\mathbb{C}$ be the set of initially detected communities, C be one community belongs to set $\mathbb{C}$, and C^i be the i^{th} community.

Since the dynamism of networks result in the change of links in each community, we only consider the node set for a community. Assume there are n_C communities in total, then the union of all C^i from C^1 to C^{n_C} constitute the whole network. For discrete time sequences $t = \{0, 1, \ldots, n\}$, let G_t be the network topology for the corresponding time stamp. Then G_0 is the actual observed one, and the network topology can only be observed by probing in the rest of time series. Assume budget b equal the number of communities that can be probed in one time stamp. If $b = n_C$, then the whole network will be updated in this time stamp. Otherwise, only b communities will be updated in the network to approximate the actual network topology.

To make the illustration more clear, we will briefly review the classical Independent Cascade Model for the influence maximization [88]. The diffusion process under the *Independent Cascade* (IC) model works in discrete time. Initially, all the seeds in set S are activated, while all the other nodes are inactive. In the next time step, any active node u in the previous step is given a single chance to activate any of its inactive neighbors with an independent probability $w(u, v) \in W$. Once a node is activated, it does not change its status anymore. The influence process continues until the number of activated nodes stops increasing.

Let $\delta(S)$ be an influence function representing the final number of activated nodes in a network when the initial activated node set is S. The objective is to find the most influential seed set S with size up to k such that they can maximize $\delta(S)$ in Eq.4.1.

$$S^* = \arg \max_{S \subseteq V, |S| \leq k} \delta(S) \tag{4.1}$$

Since the Linear Threshold (LT) model has been proved that could be unified with proper parameter initialization to IC [137], we introduce IC, and all our proposed algorithms that are equally applicable to LT.

Preliminary Problem: Given the topology of a dynamic network G at time $t = 0$ and

Table 4.1. Notation summary

Notation	Description
$\mathcal{G} = (V, E, W)$	A weighted directed graph
V	The node set
E	The edge set
W	The weight set for edges
k	# of influential nodes to be mined
S	The set of initial activated nodes
S^*	The set of final activated nodes
$\delta(\cdot)$	The influence maximization function
C^i	The ith community
n_C	The number of communities
b	The budget to probe communities
$w(u, v)$	The probability on edge (u, v)
$\widehat{E}(S)$	The influence function of S
ϵ	The parameter of probing quality
$\varpi(C)$	The bound of probing quality
τ_C	The percentage parameter for probing

the changes that can be observed by probing, supposing b is the budget allowed to probe communities, the problem is how to accurately approximate the "real" network topology at time t via selectively observing b communities and updating the remaining unselected communities based on the historical topology information.

Once the updated network topology is obtained, the next target is to maximize the influence in the network.

Ultimate Problem: Given a dynamic social network G and the probing budget b, while probing pieces of the network at each time stamp to approximate the change of the whole network, we aim to find a set S with size k such that $\delta(S)$ is maximized under the specified classical (e.g. IC or LT) influence diffusion model.

4.3 Model and Algorithm

Based on the problem defined in Section III, the input of our problem is a dynamic network G. The expected outcome is to approximate the real network change by probing the change of b communities in the network, and then find up to k seeds which can maximize the influence considering the dynamics of the social network. Below, we first present an overview of our partition method based on [178] which prepares communities for our following probing algorithm. We propose a community partition algorithm in order to further improve the efficiency and effectiveness of our solution. The influence diffusion among nodes has been taken into consideration in the community partition process to make it more applicable to our defined model. The partition algorithm includes three steps. First, we assign unique community labels from 1 to $N = |V|$ to each node. Second, we compute the influenced neighbor set for each node based on the IC model. Third, labels are propagated through the network.

4.3.1 Probing Dynamic Networks

A simple strategy for community-based network update is to randomly choose communities with equal probability and then probe the changes within each community. In this

case, communities that have not been changed may be selected and communities that have been changed may not be selected. We may lose accuracy from in-accurate approximation. A better solution is to accurately find communities that are expected to bring the biggest change in the network, so that probing the updating of the selected communities can be used to closely estimate the updating of the network. Assume $\widehat{E}(S)$ evaluates the influence spread from seed set S in the observed network G after probing the community C. Let S_m and S'_m be the maximized influence seed set obtained before and after community C has been probed, respectively. Since S'_m depends on C, the function could also be written as a function of C, *i.e.*, $S_m(C)'$. The larger $(\widehat{E}(S_m) - \widehat{E}(S'_m))$ is, the more meaningful for probing a specific C. We expect that the occurrence probability of such difference is no more than ϵ. Accordingly, let $\varpi(C)$ be the bound of the difference $(\widehat{E}(S_m) - \widehat{E}(S'_m))$. The objective of our probing algorithm is to optimize the approximation of global topology changes by only probing parts of communities. And for each community C, the maximum bound $\varpi(C)$ satisfying

$$P[|\widehat{E}(S_m) - \widehat{E}(S'_m)| \geq \varpi(C)] \leq \epsilon \tag{4.2}$$

As we mentioned, computing the expected influence spread for a given seed set is intractable even in a relatively small network. Thus, we employ the Azuma-Hoeffding inequality which provides the method of bounding differences. This method is commonly used in the analysis of randomized algorithms to estimate ϵ instead of directly calculating the influence spread.

A heuristic function $\Delta(C)$ based on the classical result of Chen [31] is developed:

$$\Delta_{\Pi}(C) = \quad 1 + d_{in}(C) - s_{in}(C) - s_{out}(C) \tag{4.3}$$

$$-s_{out}(C)[d_{in}(C) - s_{in}(C)] \tag{4.4}$$

where Π denotes the currently selected seed set; $d_{in}(C)$ is the total in-degree of nodes in community C; $s_{in}(C)$ and $s_{out}(C)$ are the number of nodes that have already been chosen as

seeds in Π among the predecessors and successors from community C, respectively. Based on this heuristic function, we could calculate the marginal improvement $\widehat{\Delta_{\Pi}}(C)$ of each community C in the observed network $\widehat{G}$. Note that, for a community C and a specific node u (which could be chosen or not chosen as a seed node), we consider C as chosen if and only if more than τ_C percentage of nodes in C have been chosen as seeds. Although the network is dynamic, the summation of in-degree of nodes in each community should be relatively stable, thus the expectation is $E[d_{in}^{t+1}(C)|d_{in}^{t}(C)] = d_{in}^{t}(C)$.

The assumption that in-degree of nodes in each community is relatively stable is based on our observation and several previous related literatures. Zhuang, Sun *et al.* [180] adopted similar assumption for the stainability of a individual node. Our assumption is a relaxed version of the assumption in [180], we consider the in-degree of nodes in a whole community is relatively stable, which could also reduce the effect of nodes' individual changes. In each time slot, for a specific node, the in-degree is the number of direct link point to that node. In social networks especially online social networks (*e.g.* Facebook, Twitter, *etc.*), most of the links are generated from friendships and interactive communications. Once a link between two nodes has been build, keeping the link has no extra cost. Furthermore, it is not common to see the situation that many links directed to one node are removed at the same time in reality. In some cases, relationships change among the nodes within the same community may not affect the in-degree of the community. For example, the communication between A and B changes to A and C (assume user A, B and C are in the same community). Thus, even though the in-degree of a single node may change, the stability actually will be increased while rising the horizon from a node to a community. Previous work in [90], Kumar *et. al* analyzed users' behavior from the community-level in [114], they describe the pattern of users' behavior in a social networks, and [114] reported the stability of community structure in social networks. According to their research, the in-degree is one microcosmic aspect of a community, which should naturally follow the same properties. From empirical datasets [104] and observations obtained from [114] also demonstrates that networks keep relative stability in real social networks, which confirm our assumption again. The results

show the ties among people who are more likely to share opinions (*e.g.* same race, gender, or socioeconomic class) decay at a slower rate than ties among persons who are likely to have different opinions. For the link of influence, the ties are much more stable in the social scenario. Therefore, we believe that the stability of the in-degree in a community of a social network is a reasonable assumption.

Suppose the latest update of the network is at time t_δ, and $|d_{in}^{t+1} - d_{in}^{t}| < t_\delta$. Applying the Azuma-Hoeffding inequality for martingale, we have

$$P(d_{in}^{t+t_\delta}(C) - d_{in}^{t}(C) \geq z) \leq e^{\frac{-z^2}{2t_\delta}} \tag{4.5}$$

Because d_{in} is sub-martingale, symmetrically

$$P(d_{in}^{t+t_\delta}(C) - d_{in}^{t}(C) \leq -z) \leq e^{\frac{-z^2}{2t_\delta}} \tag{4.6}$$

With Eq.4.5 and Eq.4.6, by applying the union bound, we have

$$P(|d_{in}^{t+t_\delta}(C) - d_{in}^{t}(C)| \leq -z) \leq e^{\frac{-z^2}{2t_\delta}} \tag{4.7}$$

Let the probability in Eq.4.7 to be bounded by ϵ. We have $e^{\frac{-z^2}{2t_\delta}} = \epsilon$, then $z = \sqrt{-2t_\delta \ln \epsilon}$. If we probe $C \in S_m$, and find that $\widehat{d_{in}}(C)$ is still higher than $\arg\max_{C' \notin S_m} d_{in}(C')$, then the performance gap will be 0; otherwise C will not be included by $S'_m(C)$ and $\dot{C} = \arg\max_{C' \notin S_m} d_{in}(C')$ will be included. Thus, the difference before and after probing C would be $\min\{0, \widehat{d_{in}}(\dot{C}) - d_{in}(C)\}$ if $C \in S_m$, and $\min\{0, d_{in}(C) - \widehat{d_{in}}(\ddot{C})\}$ where $\ddot{C} = \arg\min_{C'' \in S_m} \widehat{d_{in}}(C'')$ if $C \notin S_m$.

Thus, the value of function $\varpi(C)$ is

$$\begin{cases} \min\{0, \widehat{d_{in}}(C) - z - \min_{\dot{C} \in S} d_{in}(\dot{C})\} & \text{if} \quad C \notin S_m \\ \min\{0, \max_{\ddot{C} \notin S} \widehat{d_{in}}(\ddot{C}) - d_{in}(C) + z\} & \text{if} \quad C \in S_m \end{cases} \tag{4.8}$$

Based on the estimation of $\varpi(C)$, Algorithm 4 is proposed to update a dynamic network by probing the change of several important communities within the network. In Algorithm 4, the initialization has been shown in line 1. The probing process is an iteratively repeated process as shown from lines 2 to 21, where Lines 4 to 15 show one iteration. Lines 17 to 21 update and print the result set S_m^t of time t, which will be discussed in more detail in the next section. The overall complexity of Algorithm 4 is $t * (O(bn_C) + O(IM))$ where $O(IM)$ is the cost of influence maximization.

4.3.2 Influence Maximization in Communities

We first briefly give the algorithm of community detection. Our community detection algorithm is based on the classical algorithm proposed in [117]. The main idea is assigning a unique community label from 1 to $|V|$, where $|V|$ is the number of nodes. Then the algorithm performs the label propagation among a node and its neighbors based on the IC model. At each iteration of the label propagation, the community detection algorithm assigns the community label for a node v based on the labels of its neighbors that are influenced by v. The algorithm stops until we go through the dynamic time slot.

One straightforward method can be used to find the top-k influential nodes is to obtain the top-k influential nodes in each community locally then find the global k nodes based on the local results. However, this approach results in high computation cost, most of which is mainly caused by the repeated computation of local communities. For example, we have n_C communities and if we find top-k nodes in each community by a naive algorithm, before we obtain the final result, we need to generate at least $n_C \times k$ candidates. But the final result only contains k nodes. That is, the calculation of the other $n_C \times k - k$ nodes is a waste if skipping any of them does not influence the final result. More local results also involves

Algorithm 4: Probing algorithm

Input: network G, budget b, time t, threshold ϵ
Output: Seed set S_m^t at $t \in T$

Initialize $\widehat{G} = G$; $\forall t_\delta = 0$;
for $t \in T$ **do**
  $\forall C \in V$, $t_\delta = t_\delta + 1$;
  for $i = 1$ *to* b **do**
    $S_m = k$ nodes with maximum $\hat{d}_{in}(C)$;
    $\hat{d}_{max} = \max_{C \notin S_m} \hat{d}_{in}(C)$;
    $\hat{d}_{min} = \min_{\dot{C} \in S_m} \hat{d}_{in}(\dot{C})$;
    for $C \in V$ **do**
      $z_C = \sqrt{-2t_\delta \ln \epsilon}$
      if $C \in S_m$ **then**
        $\min\{0, \hat{d}_{in}(C) - z - \min_{\dot{C} \in S} d_{in}(\dot{C})\}$
      else
        $\min\{0, \max_{\ddot{C} \notin S \hat{d}_{in}(\ddot{C})} - d_{in}(C) + z\}$
    $C^* = \arg\max_{C \in V}\{\varpi(C)\}$
    Probe C^* in G_t and update $\hat{G}$
  // Community-based algorithm in the next section
  for $i = 1$ *to* k **do**
    $C^* = \arg\max_{C \in V \S_m^t} \hat{\Delta_\Pi}(C)$
    $S_m^t = S_m^t \cup C \cap S_m^t$
    Update $\hat{\Delta}_\Pi(C)$
    return S_m^t

more operations in further filtrating.

To reduce the cost, we utilize the dynamic programming to find communities that could provide the global kth influential nodes. Wang *et. al.* [150] proposed an algorithm named *CGA* which provides a solution to do the influence maximization based on communities. However, their algorithm cannot be directly employed in dynamic networks. Furthermore, the community detection algorithm used in [150] prevents overlap of communities in the network, which is not practical in reality. Compared with *CGA*, our algorithm has the following improvements: (1) A more practical community detection algorithm which allows community overlap has been proposed. The detailed algorithm is ignored due to space limitation. The main idea is based on the algorithm proposed in [171] which divides a network according to the structure similarities. (2) Allowance of overlap means that some nodes can participate in different communities. We build a max-heap to store all the intermediate results, which guarantees the worst adjustment time cost is limited within $O(n \log n)$. Based on how much enhancement of influence derived from each node, the algorithm ranks nodes with time. (3) *CGA* utilizes the diffusion speed to measure the speed of influence. In practice, the speed of influence in a social network is based on the distance of connection among users, which should be measured by the number of minimum hops or the activated nodes on the connected path. Different from *CGA*, our probing algorithm takes the update of a whole network into account but ignores the calculation of the diffusion speed.

Dynamic programming gives us a way to obtain results through iterations by keeping intermediate results in a table. Let S^*_{i-1} be the set of influence nodes obtained in the $(i-1^{th})$ step through the influence maximization process. If we find the i^{th} influential node in community C^l, denote the maximal increase as $\triangle\delta_l$, then we have

$$\triangle\delta_l = \arg\max\{\delta_l(S^*_{i-1} \cap v_j), \delta_l(S^*_{i-1}) | v_j \in C^l\} \tag{4.9}$$

The notation v_j represents a new node added to the objective function which belongs to community C^l, and $\Delta\delta_l$ is the local influence increment brought by v_j's joining. It is worth

mentioning that the influence in Eq.4.9 is computed only in local community C^l rather than the whole network, which is pretty fast in practice. To evaluate which community is a better choice, the influence increment on one community is used as the measurement. The community which brings the largest increment will be the best candidate to find the ith influential node.

Let $S[l, i]$ denote the total influence of finding ith influential node in the first l communities, where $l \in [1, n_C]$, $i \in [1, k]$, $\delta[l, 0] = 0$, and $\delta[0, i] = 0$.

$$\delta[l, i] = \arg\max\{\delta[l-1, i], \delta[n_C, i-1] + \triangle\delta_l\} \tag{4.10}$$

Eq.4.10 is a typical dynamic programming process. When we find the i^{th} influential node, if the influence function $\delta(\cdot)$ has higher value in the first $l-1$ communities than in the l communities, we mine it from the first $l-1$ communities. Otherwise, we mine it from the previous l communities.

Algorithm 5 shows the process of finding the top k influential nodes in a network after probing the dynamic of our solution. Lines 1 and 2 show the initialization. In the loop (lines 3 to 17), according to the equation, lines 3 to 14 shows the process of maintaining a heap H and storing all the information and the influence incremental value of the tested nodes. After sorting the result in each community, line 17 outputs the final result. The complexity of Algorithm 5 is $O(|L|k^2)$. Since k generally is much smaller than nodes number n, Algorithm 5 is almost a linear algorithm except line 16. Our algorithm limits the calculation of influence maximization within each community, which could speed it up compared with global network computing.

4.4 Experimental Study

To evaluate the performance of our proposed approaches, we test our algorithms on synthetic data based on well-studied models as well as real social networks. To test the effects of network size, topology, and dynamics on our proposed solution, we generate some

Algorithm 5: Community-based dynamic algorithm

Input: network G, size of result k
Output: Top influential node set with size k

Initialize communities to L with $n_l = |L|$, and all $\delta[l, 0] = 0$;
$\delta[0, k] = 0$, heap $H = null$, result $R_{l,k} = null$;
for $i = 1$ *to* k **do**
    for $l < n_l$ **do**
        $\triangle\delta_l = \arg\max\{\delta_l(S^*_{i-1} \cap v_j), \delta_l(S^*_{i-1}) | v_j \in C^l\}$
        $\delta[l, i] = \arg\max\{\delta[l-1, i], \delta[n_C, i-1] + \triangle\delta_l\}$;
        if $\delta[l-1, i] \geq \delta[L, i-1] + \triangle\delta_l$ **then**
            $R_{l,i} = R_{l-1,i}$;
            Update the heap H for all tested nodes;
            // Mine the i^{th} influential node from first $l-1$ communities;
        else
            $R_{l,i} = l$;
            Update the heap H for all tested nodes;
            // Mine the i^{th} influential node from the mth community;
    for $i = 1$ *to* k **do**
        According to $R_{l,k}$, calculate the most influential nodes in each community by IM algorithm [137];
return k *influential nodes*

Table 4.2. Details of synthetic data

Network model	Nodes	Edges	Communities #
Syn-SmallWord(S)	2,000	13,657	15
Syn-SmallWord(L)	50,000	468,391	36
Syn-Kronecker(S)	2,000	21,062	17
Syn-Kronecker(L)	50,000	723,276	41

synthetic dynamic networks. Two different models - Small-world graph and Kronecker graph models are applied to generate networks. Besides generated networks, real social networks are also used to verify the performance of our proposed solution.

4.4.1 Data and Observations

Data from Synthetic Generated Social Networks

Small-world graphs: the small-world network model is a very classical model following the small-world features according to "small-world" [151]. This model is referred as *Syn-SmallWord*.

Kronecker graphs: this generative model proposed in [96] generates a network in a natural way. The networks grow from 5 initial nodes and then Kronecker idea is repeatedly apply to expand the network. This model is referred as *Syn-Kronecker*.

Based on the initial networks generated from the above models, we dynamically change each network based on the idea proposed in [9]. Since we have multiple synthetic networks in the experiments, the average summary of 10 networks' statistic features has been used instead. As shown in Table 7.2, we generate two scales (small (S) and large (L)) of networks for both models with time stamp length of 50 and 100, respectively.

Data from Real Social Networks

Epinions is a Who-trust-whom network, where nodes are users using their services and an edge from node u to another node v means u has influence on v (u is trusted by v). This network includes $75,879$ nodes and $508,837$ edges.

Slashdot is a technology-related news network which features user-submitted and editor-

evaluated contents. This network includes 77,360 nodes and 905,468 edges. Both *Epinions* and *Slashdot* are obtained from SNAP [2].

Twitter is one of the most notable micro-blogging service provider [3]. Twitter users can publish tweets (with a limited length of 140 characters). We use the dataset obtained from [79]. The subnetwork includes 112,044 nodes (users of Twitter), and 468,238 edges (following relationships) and 2,409,768 tweets posted by the selected users.

Inventor is a network of inventors (extracted from USPTO [4]) obtained from [132]. The edges in this network represent the co-inventing relationship. The *Inventor* network consists of 2,445,351 nodes and 5,841,940 edges.

Our work aims at probing partial communities in dynamic networks to maximize the influence from a global scale. All the above data sets are from real static networks. Based on those data sets, dynamic networks are derived from randomly select nodes and communities.

Particularly, the Amazon co-purchase network's which is a network with real dynamic observations that had been used to verify the effectiveness and efficiency of our algorithm.

Amazon Dynamic Networks is based on the *Customers Who Bought This Item Also Bought* feature of the Amazon website. The four data sets are from March to May in 2003. There is a directed edge from product i to product j if i is frequently co-purchased with j [95].

Shown in Fig.4.2 is the average degree of all the 8 real data sets we have used. Most research literatures assume that the probabilities or the weights on edges and the thresholds are given as the input. However, as pointed out in [49], learning those probabilities and thresholds has remained a non-trivial problem. Therefore, we use the learning algorithms proposed in [122] to achieve the balance between complexity and practicability to the raw input data for all the synthetic generated and real networks. For the Amazon data sets, as shown in Table 7.3, since they are a series of snapshots from the network, we generate the real influence spreading trend by comparing our model with the real learning algorithm proposed

[2]http://snap.stanford.edu/data/
[3]http://www.twitter.com
[4]http://www.uspto.gov/

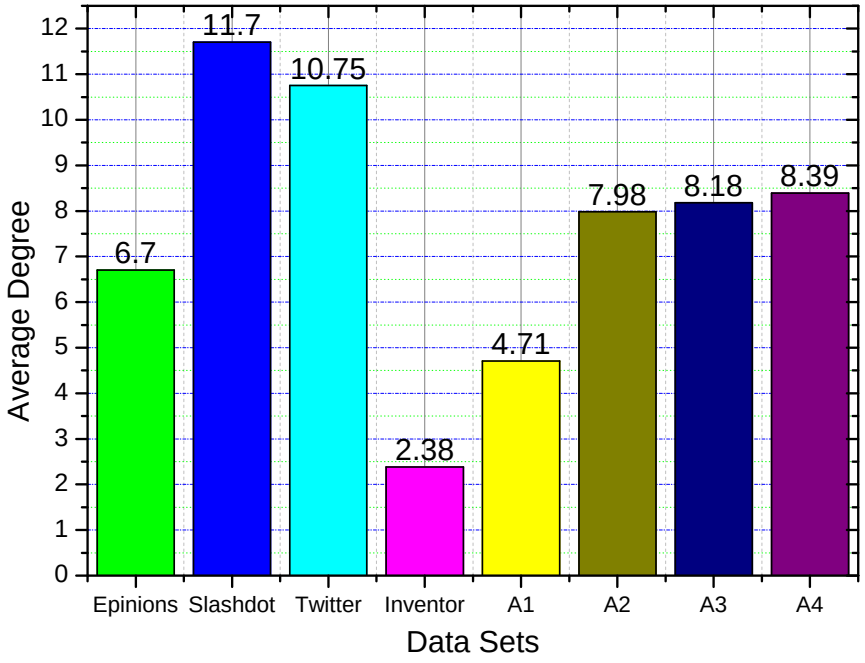


Figure 4.2. Average degree of the real data sets

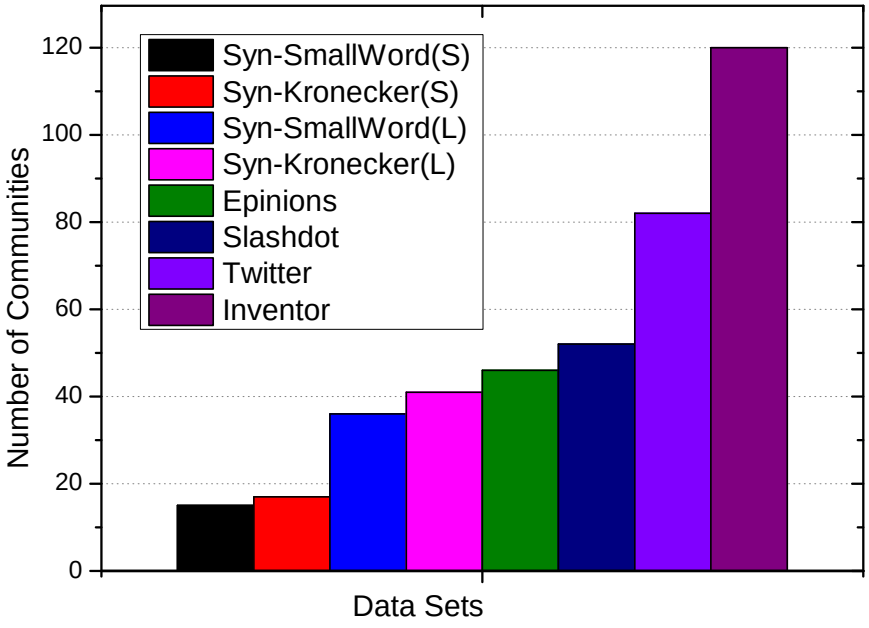


Figure 4.3. Number of communities in each network

Table 4.3. Amazon dynamic dataset

Data Set	Nodes	Edges	Diameter
Amazon0302 (A1)	262,111	1,234,877	29
Amazon0312 (A2)	400,727	3,200,440	18
Amazon0505 (A3)	410,236	3,356,824	21
Amazon0601 (A4)	403,394	3,387,388	21

in [50] which initially treats the data as the user log then solves the influence maximization.

The probability is learned from the networks in the previous position. That is, the probability of the second network comes from the first one, and the probability of the last network is generated from the first three network snapshots through the linear prediction.

4.4.2 Algorithm Evaluation

In both synthetic generated and real networks, we evaluate algorithms proposed in Section 4.3 based on the classical approach introduced in [178]. Several optimizations are used to partition a network. The community detection is a pre-process for all the data. Communities are detected by initially assigning each node with a unique label and adopting the label step by step for most of its current neighbors. In this case, iteratively, the most densely connected groups will be grouped into a community. The result is shown in Fig.4.3, from which we can see that as the network size increases, the number of communities increases accordingly.

All the codes are implemented in Python 2.7 based on the latest version of Snap.py [5], and all experiments are performed on a PC running Windows 10 with Intel(R) Core(TM) i3-2120 CPU 3.30GHz and 12GB memory.

We first evaluate the probing algorithm, then test the overall efficiency. Comparisons are conducted with the following algorithms.

- Random Nodes Probing (*RandNodes*). Randomly choose $b * |\overline{C}|$ nodes, where $|\overline{C}|$ is the average community size in the network, and then probe communities with uniform

[5]Python interface for SNAP

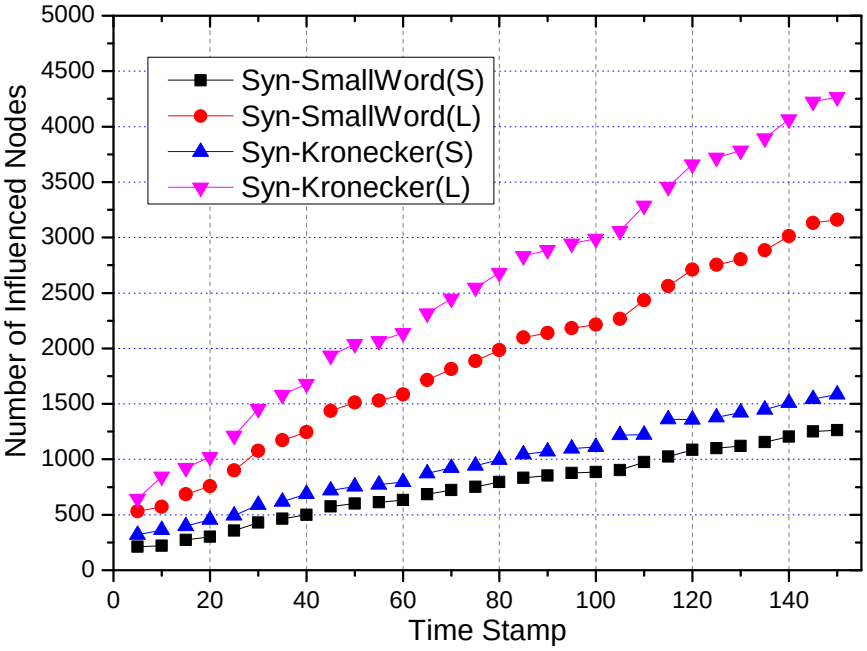


Figure 4.4. Results of the probing algorithm with $b = 2$

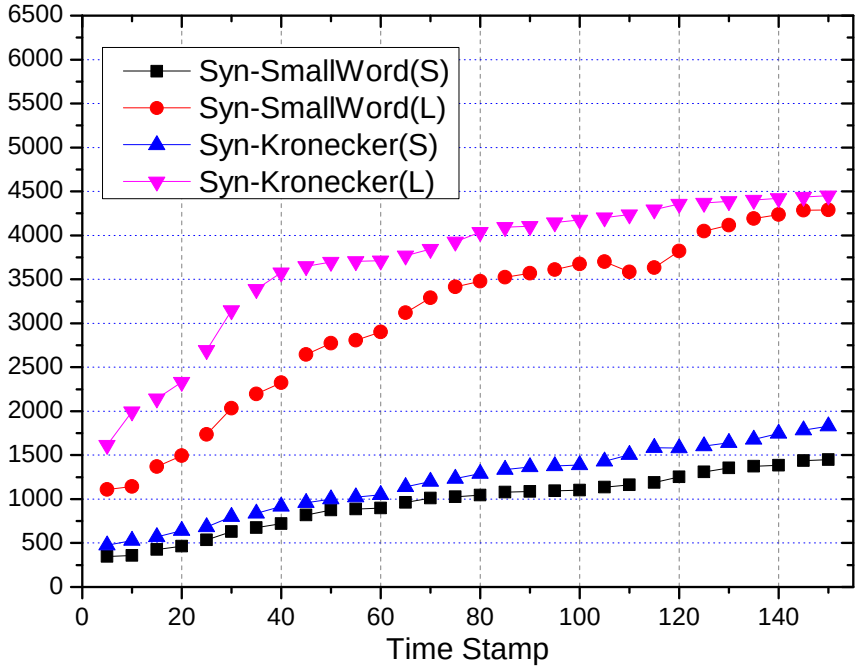


Figure 4.5. Results of the probing algorithm with $b = 10$

probability at each time stamp.

- Random Community Probing (*RandComm*). Randomly choose b communities and probe with uniform probability at each time stamp.

- Maximum Gap Probing (*MaxG*). This algorithm is proposed in [180]. Choose $b * |\overline{C}|$ nodes according to their algorithm.

- Optimized Communities Probing (*OptComm*). This is our algorithm introduced in Section 4.3. Probe b communities per time stamp.

- Global updating the whole network (*OptWhole*). This is the optimal benchmark of updating a whole network in each time stamp.

We compare all the algorithms from the aspects of effectiveness and efficiency. In order to be more reasonable, we calculate the influence expectation based on the uniform algorithm proposed in [30]. Later in this section, we will show the advantages of our integral solution.

We evaluate the performance of our probing algorithm on the synthetic networks by time stamp. Fig.4.4 and Fig.4.5 show the results of 4 different synthetic networks with different probing budget b. We can see that as b increases, the beginning increments of influenced nodes are sharper, then they go to a gentle slope. With more time stamps consumed, the algorithm could probe more accurate structure and property of a network, then more nodes are influenced through the algorithm.

We divide our data into two groups: normal size group (Epinions and Slashdot) and large size group (Twitter and Inventor). With budget $b = 5$, Fig.4.6 and Fig.4.7 show the number of influenced nodes and the running time, respectively.

Consider the number of the influenced nodes in 5 different algorithms, our *OptComm* is as good as *MaxG* and approaches to the performance of the optimal algorithm *OptWhole*. From the result, *MaxG* is a little bit better than our *OptComm*. That is because their algorithm considers a node as a unit. It finds the most active node and probe it. While, our algorithm considers a community as a unit to reduce computational cost which may miss the

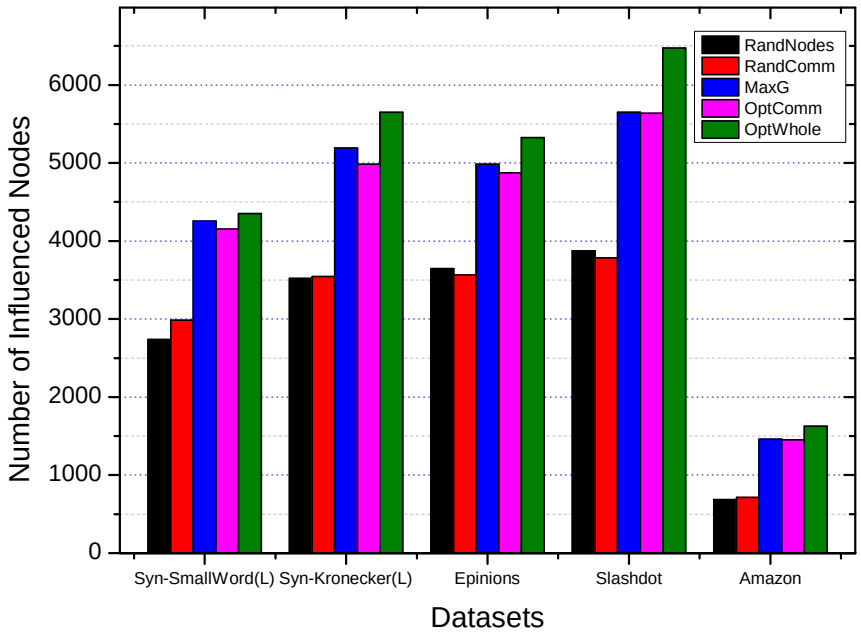


Figure 4.6. Number of influenced nodes in the probing algorithm

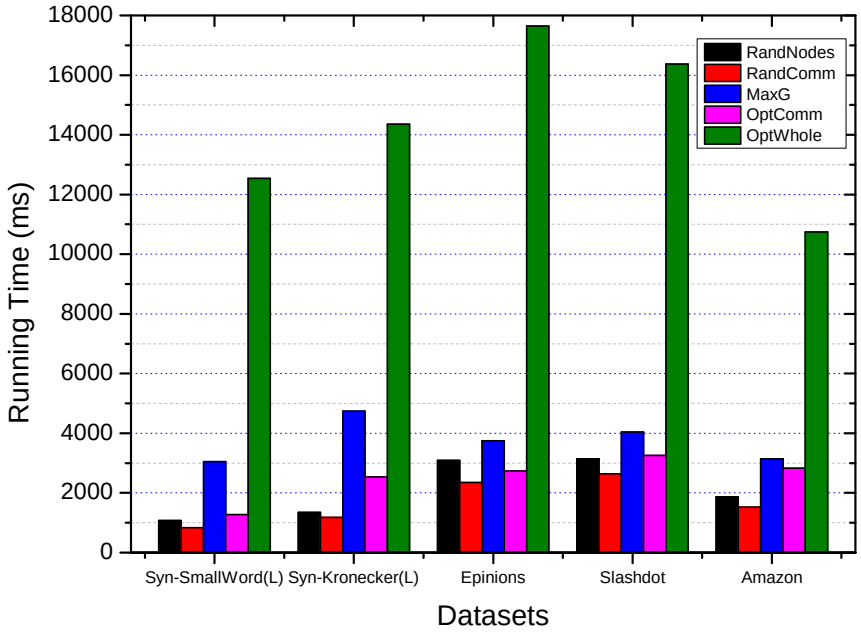


Figure 4.7. Running time of the probing algorithm

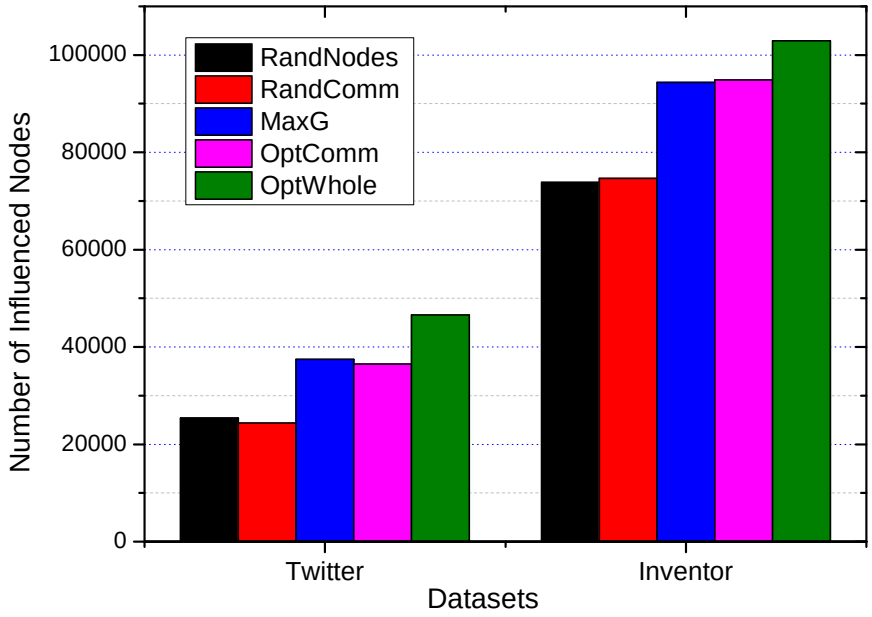


Figure 4.8. Number of influenced nodes in the probing algorithm

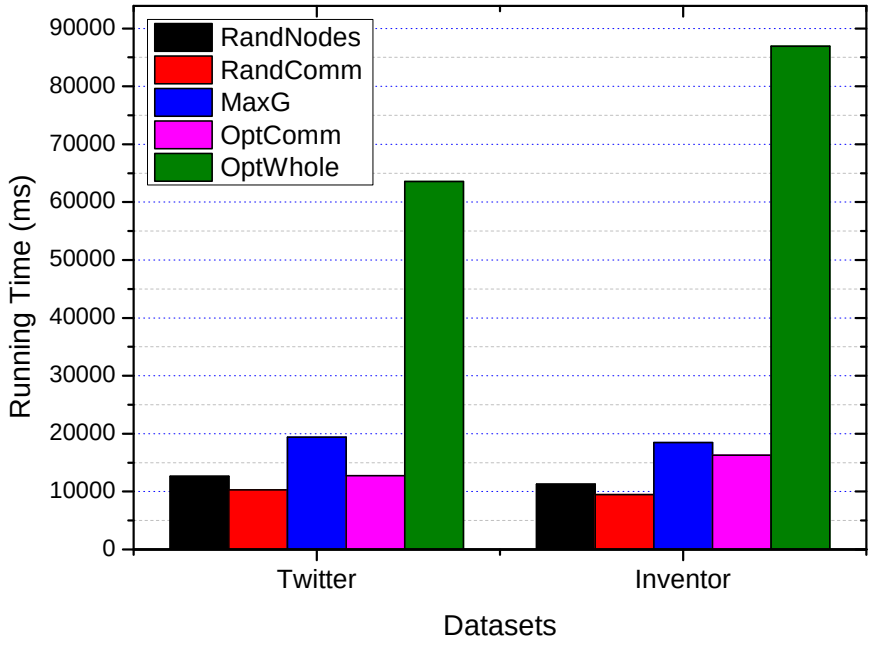


Figure 4.9. Running time of the probing algorithm

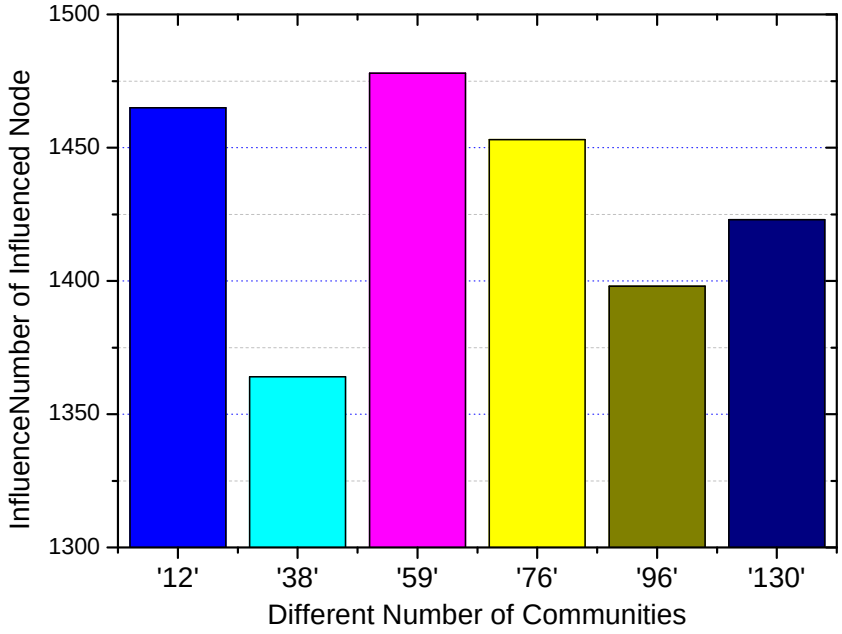


Figure 4.10. Number of influenced nodes with different number of communities

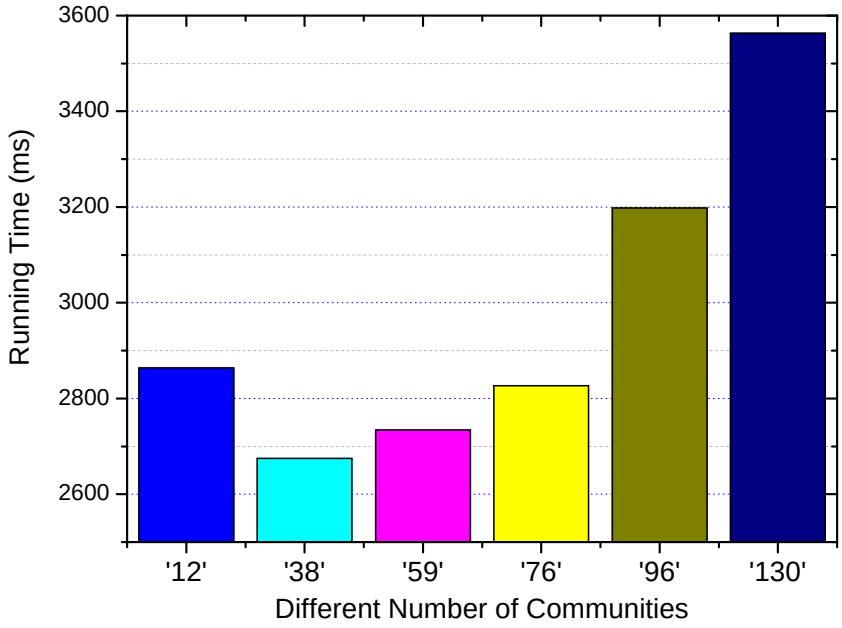


Figure 4.11. Running time comparison with different number of communities

most active node. *OptComm* takes the advantage of considering community as an unit which partly save computational cost of probing nodes within the same community. This strategy could calculate a more accurate influence within the same community especially when the community structure is sparse. This is the situation of *MaxG* and *OptComm* in the Inventor network. Inventor has 120 communities compare to 82 communities of Twitter. At the same time, as shown in Fig. 4.2, Inventor has the smallest average degree 2.38 compared with 10.75 of Twitter.

The average degree of Twitter (10.75) is much higher than Inventor (2.38). According to the results in Fig. 4.8 and Fig. 4.9, the running time of OptComm is much better than MaxG in Twitter compared to Inventor. We can easily find out that the difference of the two network that the number of influenced nodes in Twitter is much higher than Inventor. This is resulted of different sparseness of different networks. Therefore, with the same probing budget, for a sparse network, algorithm *OptComm* could update the network more accurate and achieve better performance.

The advantage of our *OptComm* is when we take a community as probing unit, although the community has $|C|$ nodes, the real number of nodes needed to be updated is fewer than $|C|$ because most nodes in a community have strong connections with each other, and most connections may have already been updated through previous operations. Even more distinct, as shown in Fig.4.7, benefitting from our community-based probing algorithm, our *OptComm* is much faster than *MaxG*, and even close to the random algorithms. The two random selection approaches are approximately the same, but the community based algorithm has a better performance which verifies that influence diffusion is community orientated. Both observations state that Inventor is a very sparse network, which further confirm the advantage of algorithm *OptComm*. *OptComm* is only a litter bit better than *MaxG* as shown in Fig.4.8.

We apply our algorithm to larger size networks. Results are shown in Fig.4.8 and Fig.4.9. Contrasting with Fig.4.6 and Fig.4.7, the larger the data size, the advantage of the community-based probing algorithm is more obvious. Especially, Fig.4.9 shows that the

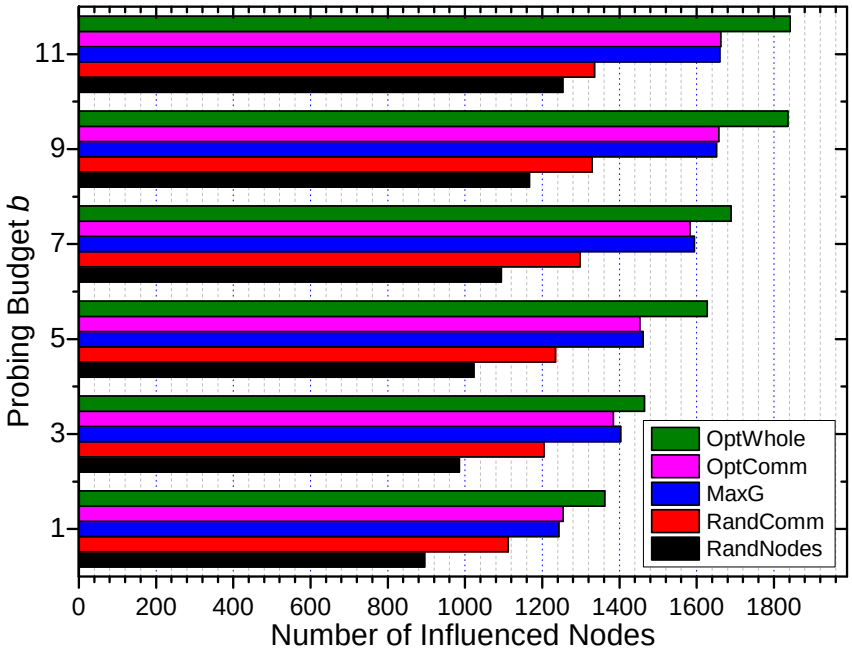


Figure 4.12. Effect of budget b

running time of our algorithm is only 2827ms which is 26.3% of the *OptWhole* and better than *MaxG*'s running time. Considering the overall performance of our solution, the running time as presented in Fig. 4.7, *OptComm* is much better than *MaxG*, and this result further verified our model and algorithm.

We separate the running time into two parts in order to demonstrate the advantage of our algorithm more clear. In the first step, our probing algorithm takes advantage of community probing which is comparable to node-based probing on both accuracy and efficiency. In the second step, based on the community based probing algorithm, we maximize the influence from local to global. The dynamic programming technique allow us to take full advantage of efficiency and achieve our contribution. Fig. 4.13 shows the total running time, from which we can see that our *OptComm* is significantly better than *MaxG*.

The budget b is the parameter that balance the degree of dynamic approximation and the computational cost. Larger budget b gives more accurate approximation results. If budget b is equal to the total number of communities in the network, the algorithm could obtain the real update to the dynamics of the whole network.

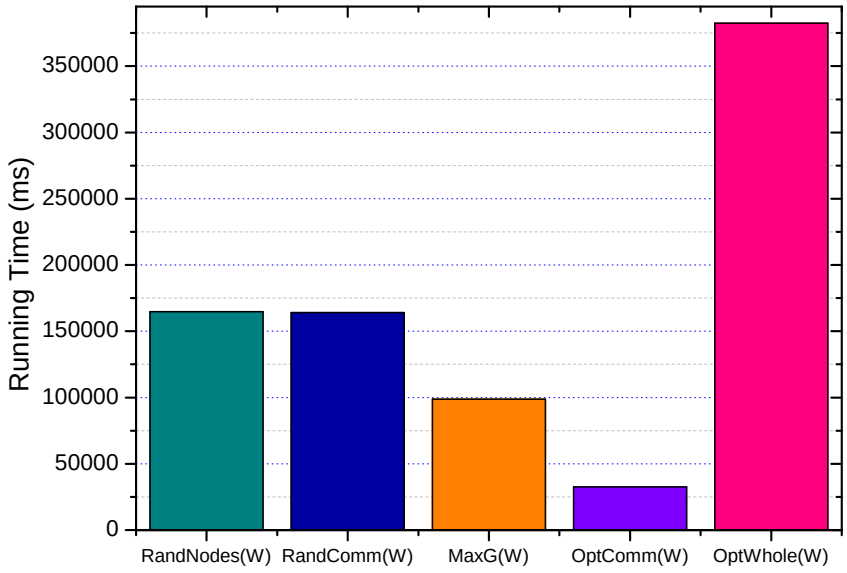


Figure 4.13. Overall running time comparison

As shown in Fig.4.12, more nodes are influenced as b increases. But as we increase the budget to a relatively large level such as 9 to 11, the more communities we probe, the marginal increment of influenced nodes is decreasing. This is because when the budget is large enough, more inactive communities will be tested which could not contribute much to the final result due to their inactivity. On the other hand, the diminishing of marginal utility once again verifies that it is not necessary to update the whole network with massive cost. At the same time, larger budget of probing results in higher computational cost. Our probing solution is targeted at saving the computational cost. It is not very meaningful to use an enormous budget. In the experiment section, we test the performance of different budget b in different data sets.

From the real data experiment result, budget b from 3% to 5% of the total communities could achieve the best balance between efficiency and computation overhead. Therefore, in our evaluation section, budget $b = 5$ is used in most of experiments, which is a typical value we have tested. Other budget also follows the similar trend.

According to our analysis and experiment results, the most important factor of the probing cost is the budget b. Different topology will affect the community structure, but

has no effect the probing cost significantly. To proof this, we changed the setting of the community detection algorithm to obtain different community detection results. Besides our default community detection result of 76 communities, we got another 5 different scale of community detection results as 12, 38, 59, 96, and 130, respectively. As shown in Fig. 4.10, as well as the increase number of communities, the number of influenced nodes is changing fluctuated.

In order to further investigate how the community was affecting our algorithm cost, we also compared the cost of different community scales as shown in Fig. 4.11. The real experiment result shows that when the number of communities is very small which implies that each community is very large, the cost of probing is increased while the benefit of community probing is decreased, but the trend is kept in a relative smooth way. In contrast, when the number of communities is increasing, the cost is also increasing since very small communities are generated in the network. Another reason for the increase of cost is when we have more communities, we need probing and compute more communities during both steps in our solution. From the result, we can determine that the most important factor on probing cost is how many nodes we need to probe in total but not the number of communities. As we have tested, with the same number of nodes, investigating nodes by community can boost the efficiency of the whole algorithm. Even though, from both Fig. 4.10 and Fig. 4.11, the change of experimental results are still kept in a relatively small range (for influenced nodes from 1300 to 1500 and for running time from 2500ms to 3500ms). This result states that the community structure does not have a strong effect on our algorithm. But other concerns such as probing techniques and dynamic programming algorithm do affect the final result.

In the end, we apply Algorithm 4 and Algorithm 5 with probing budget $b = 5$ and $k = 50$ as an integral solution to evaluate the performance towards the Amazon data. As shown in Fig.4.13, our running time is around 1/3 of *MaxG*'s and less than 10% of the *OptWhole*'s.

Overall, The proposed probing and dynamic programming algorithms as an integral solution shows its outstanding performance regardless of random heuristic strategy or up-

to-date techniques. In particular, the algorithms could be well employed to networks in a larger size, which indicates its high practicality and scalability.

4.5 Summary

In this work, we propose a practical dynamic network probing framework to explore the real changes of networks. The probing framework takes the community as a unit, updates network topology by only probing b communities instead of searching the entire network. Besides, we propose a divide-and-conquer strategy to apply the dynamic programming technique to community-based influence maximization. The comprehensive experiment results show that our model can achieve comparable influence diffusion performance compared to the node-based probing algorithm, while having a much better computation cost, efficiency and more applicable to large scale networks.

Chapter 5

BROADEN THE INFLUENCE PROPAGATION

5.1 Introduction

Each month, more than 1.3 billion users are active on Facebook, and 190 million unique visitors are active on Twitter. Furthermore, 48% of 18-34 year old Facebook users check their online personal web pages when they wake up, and 98% of 18-24 year old people are involved in at least one kind of social media[1]. Since customers are the most important foundation of business, Online Social Networks (OSNs) have become one of the most effective and efficient solutions for marketing and advertising. But there is still no specific answer for how to handle and utilize data from OSNs. The development of OSNs and the resultant of a huge volume of data bring both opportunities and computation challenges.

Influence maximization, as one of the most popular topics in OSNs, attracts a lot of interest recently. Several models have been proposed in literatures [88, 95] to model influence diffusion. However, because of the complexity and diversity of social phenomenon, many important features have been ignored, resulting the practical influence diffusion is still not well modeled. We are facing a lot of challenges such as timeliness, acceptance ratio and breadth while analyzing and maximizing influence in OSNs. *Timeliness* refers to the phenomena that the effect of influence would decay with time; *acceptance ratio* measures the percentage of influence which gets a response; and influence *breadth* aims at maximizing influence not only by having more users, but also by achieving a broader user distribution in reality.

In the viral marketing and media domain, it is very common that many limited-time promotions and immediacy news exist where the influence and spreading of them decay with time. During the process of advertisement promotion or marketing strategies, the fact

[1]http://www.statisticbrain.com/facebook-statistics/

that a message could be passed on to someone never means the message could be accepted by the receivers (acceptance means the receivers take actions or response to the message). Therefore, receiving and accepting would be two procedures of influence. From this point of view, taking the acceptance ratio into account for influence would make the model more practical than the traditional naive way. The expectation of the influence model traditionally formulated is considered as the depth of influence. Another important issue is how broad area the influence could be from the selected source seeds: the breadth of influence. Breadth relies not only on the number of influenced nodes, but also on the size of the area that could be covered by the influenced nodes. Surprisingly, although most researchers consider the path or routing of influence spreading based on network structure, as far as we know, there is not any existing work considering the range (breadth) of the influence yet. Therefore, the question appears: which one is more important for influence maximization? influence more users in depth [2] or breadth?

Let us take a conventional social network activity as an example to discuss influence diffusion in daily life. Assume there is one user on Facebook sharing a new song or movie. This action results in an influence diffusion process. That is, friends or followers of the action initiator will have similar behaviors - be influenced. Considering one instance, user *Mike* posts a new status "*I got a new iPhone 6 plus from Apple Store with student promotion. It is awesome!*" with pictures on Facebook. All of *Mike*'s friends and followers will get this information from their Facebook's news feed or related search results. For timeliness, the effect of this influence will be weakened as time goes on. For acceptance ratio, obviously not all the neighbors who see the post will forward it, although some of *Mike*'s friends might have already been influenced and begun to take next step to purchase an iPhone, but some of his friends might have simply ignored this post. We consider the receiving of that post as the first step of influence, and all the users having a friend relationship with *Mike* have a probability to receive this influence. But only the neighbors who comment, forward this status, or take

[2]depth might result in "rendezvous problem", which is a term from mathematics to state the overcrowded of seeds selection

response action regarding this post could be considered as accepting the influence, which is the second step of the influence. For the breadth of influence, one possibility is a lot of *Mike*'s friends are studying at the same department of the same university. If we evaluate the influence ability of *Mike* in the whole social network, he might not be as good as another user *Michael*, who has fewer friends studying in many different universities. Compared with *Mike*, *Michael* has a good chance to pass the influence much more broader than *Mike*. Thus, all the three aforementioned factors we mentioned above should be taken into account.

Additionally, how to evaluate influence in OSNs is still an open problem. Although several models have been proposed to evaluate, influence by analyzing the history logs [49] or learning users' behaviors [163], there is still few literatures considering the impact between users in a timeliness model with respect to the influence decaying process and the optimistic selection for a better acceptance ratio. Therefore, different from the most traditional influence models which only focus on the simple traditional influence expectation result or the efficiency of the algorithm [31, 52, 131], we deal with influence maximization from a much more practical and comprehensive perspective.

In this chapter, we address the problem of identifying the node set which maximizes influence in practical social networks. Our model incorporates influence decay function, opportunistic selection and broader maximization accommodating to three factors: timeliness, acceptance ratio and breadth. More specifically, our contributions are summarized as follows:

1. We formulate the problem of influence maximization with opportunistic selection in a timeliness model $ICOT$. The model incorporates the timeliness feature and considers the decaying of influence diffusion.

2. We propose opportunistic selection to deal with the acceptance ratio which represents the real reception of influence transmission in practice.

3. We show the NP-hardness of the problem together with the monotone and submodular properties of the object function. Our model is generalizable to other influence maximization problem by using a different influence diffusion model. The analysis result

shows that the classical models (e.g. IC) are special cases of our model.

4. Considering the coverage of influence diffusion, we take the first step to explore the relationship between the breadth and depth of influence and propose the model $BICOT$. Specifically, in the extended version of our model, we use the number of communities to measure the breadth of the influence, which is novel.

5. The experiment results on several real data sets show that our solution can significantly improve the practicability and accuracy against several baseline methods. Especially on the aspect of influence spreading range.

The rest of the chapter is organized as follows. Section 5.2 presents the preliminaries and problem definition, then we introduce our model with analysis and the algorithm in Section 5.3. The evaluation results based on real and synthetic data sets are shown in Section 5.4. Section 5.5 concludes the chapter.

5.2 Preliminaries and Problem Definition

Kempe et al. [88] formulated the influence maximization problem as a discrete optimization problem: given a network with a node influence probability (weight) on each edge, a node set with a fixed size is initially activated as seeds and these seeds begin to influence other nodes under a certain model. The objective is to find the optimal node set which could maximize the expected number of final active nodes. Formally, we can model a network as a directed graph $\mathcal{N} = (V, E, W)$ where V, E, W represents the vertices, edges, and weights, respectively. Let function $\delta(\cdot)$ be the expected number of active nodes at the end of the influence process. Out purpose is to identify a seed set S of size up to k which devote such S which can maximize $\delta(S)$. Denote such S as:

$$S^{\star} = \arg \max_{S \subseteq V, |S| \leq k} \delta_{IC}(S) \tag{5.1}$$

Table 5.1. Notations adopted in sections

Notation	Description
$\mathcal{G}$	A weighted directed graph
V	The vertices set
E	The edge set
W	The weights set on edges
O	The opportunistic acceptance ratio set
k	The number of influential nodes to be mined
S	The set of influential nodes
τ	The influence decaying ratio
$d_\tau(t)$	The decrease ratio of influence at time t
$f_o(\cdot)$	The information diffusion ratio for current step
$\tilde{T_o}$	Threshold of opportunistic selection ratio
$\delta_{ICOT}(\cdot)$	The objective function for $ICOT$ model
$\delta_{BICOT}(\cdot)$	The objective function for $BICOT$ model
$P_C(v)$	The percentage of communities node v influenced
$i(v)$	The initialize PageRank score for node v
φ	Tradeoff parameter for depth and breadth

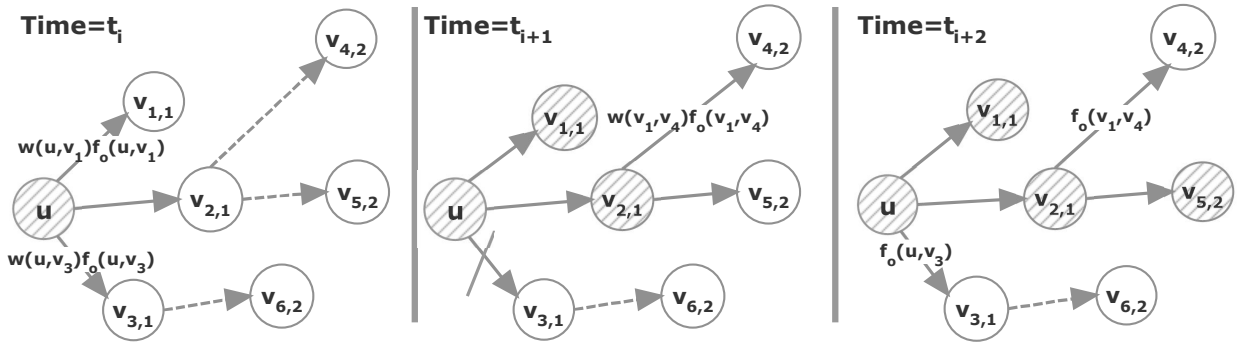


Figure 5.1. Models of social influence

The diffusion process under the *Independent Cascade* (IC) model works in discrete time $t_0, t_1, t_2, \ldots$. Initially, all the seeds in set S are activated at t_0, while all the other nodes are inactive. As the process continues to time t_i ($i > 0$), any active u in the prior time t_{i-1} is given a single chance to active any of its currently inactivate neighbors with independent probability $w(u, v) \in W$. Once a node is activated, it stays and will not change status any more. The stochastic process iteratively continues until no new activated node appears.

The general idea behind IC is to measure the influence ability by the number of activated nodes. It targets at finding the optimal seed set which can maximize the global influence in the network. As mentioned in Section 5.1, in practice, the influence diffusion process has to face opportunistic selection and time decay. Thus, function $\delta(\cdot)$ should also be improved to adapt to the reality.

We first extend the IC model to a dynamic network with time decay and opportunistic selection, then we propose a utility function to measure influence breadth.

Formally, we introduce our $ICOT$ ($\underline{IC}$ model with $\underline{O}$pportunistic selection and $\underline{T}$ime decay) model. We define $\delta_{ICOT} : 2^V \rightarrow \mathcal{R}$ as the objective function such that $\delta_{ICOT}(S)$ with $S \subseteq V$ is the final expected number of activated nodes under $ICOT$ model.

$$S^{\dagger} = \arg \max_{S \subseteq V, |S| \leq k} \delta_{ICOT}(S, o, \tau) \tag{5.2}$$

where o is the opportunistic acceptance ratio set controlling the acceptance of influence, and

τ is the influence decaying ratio controlling the decaying process as time goes on.

The influence maximization problem with opportunistic selection under the *ICOT* model is the problem of finding the optimal seed set S with at most k seeds such that the expected number of activated nodes is maximized.

The extended version of *ICOT* is *BICOT* (***B****roadly* influence maximization problem under the *ICOT* model). Different from *IC* which only maximizes the influence expectation in depth, *BICOT* considers both depth and breadth of influence. We will provide more properties and details of this model in the next section.

$$S^{\ddagger} = \arg \max_{S \subseteq V, |S| \leq k} \delta_{BICOT}(S, o, \tau, \varphi) \tag{5.3}$$

where φ is the parameter leveraging depth and breadth of influence.
As a summary, the two proposed models could be formulized as follows. Let M be the influence model. Our purpose is to find the optimal node set such that:

$$S^{\S} = \arg \max_{S \subseteq V, |S| \leq k} \delta_{M}(\cdot) \tag{5.4}$$

Problem Statement:

Input: Directed graph G, parameters (τ and $\tilde{T}_o$ for *ICOT* or α, β, ϵ, τ, $\tilde{T}_o$, and φ for *BICOT*), influence model type M (*ICOT* or *BICOT*).

Output: Optimal seed set $S^{\S}$ which maximizes influence in G under M.

5.3 Model Analysis and Algorithm

This section introduces the details and properties of the *ICOT* model and the *BICOT* model.

5.3.1 Model Analysis

We model a social network as a directed graph $\mathcal{G} = (V, E, W, O)$. We may learn the influence probability weight $w(u, v) \in W$ on each edge from practice initially. O denotes the set of opportunistic acceptance ratio functions where $f_o(u, v) \in O$ represents an independent probability indicating whether the target could accept the influence or not (in this chapter we use the same weight $w(u, v)$ as an example, $f_o(u, v)$ could also be learned according to further information related to real data). $d_\tau(t)$ is a decaying function representing the decrease of influence, where t is the beginning time when only the selected seeds turn active, $t_{current}$ is the current time, and τ is the decaying coefficient.

$$d_\tau(t) = \frac{t_{current} - t}{\tau} \tag{5.5}$$

In $ICOT$, due to time decay and influence decrease, for each step of influence diffusion, an opportunistic acceptance function $f_o(\cdot)$ is designed to model the latest step of the information diffusion with continues time decaying.

$$f_o(u, v) = w(u, v)^{d_\tau(t)} \tag{5.6}$$

The acceptance ratio between nodes u and v denoted by $f_o(u, v)$ is an independent probability different from $w(u, v)$. In $ICOT$, the probability that u's influence reaches v is measured by $w(u, v)$, the opportunity whether v accepts this influence or not is decided by both $w(u, v)$ and $f_o(u, v)$. Furthermore, the final objective function is also improved to $\delta_{ICOT}(\cdot)$ which includes the weight all the active nodes try to influence their neighbors at the end (all the neighbors of the active nodes in the last step) with acceptance ratio greater or equal to threshold $\tilde{T}_o$. Those nodes will also be marked as activated according to our case study in Section 5.1.

Fig. 5.1 shows an example of influence diffusion under the $ICOT$ model. The shaded circle represents an activated node, a blank circle represents an inactivated node, solid line represents an influence attempt with probability $w(u, v)f_o(u, v)$, and a dash line changes to

a solid line only when the start node becomes active. Node v_{a,t_d} denotes the status of v_a in the diffusion time slot t_d. As shown in the example, at the beginning time t_i, only node u is *active* and all the links from u to its neighbors indicate the chance (attempt) of influence (solid line) from u to other nodes (e.g. v_1, v_2, and v_3). If v_1, v_2, and v_3 could be influenced (received ($w(u,v)$) and accepted ($f_o(u,v)$) the influence) successfully, their status will change to *activate* and they continue to influence others in the next step as shown by the dashed link from them. At time t_{i+1}, nodes v_1 and v_2 are influenced successfully by u, but node v_3 is not. Because link (u, v_3) is the only link between u and v_3, and v_3 does not receive the influence from u by $w(u, v_3)$ successfully. u will not try to influence v_3 by $w(u, v_3)$ anymore but will attempt to influence v_3 by $f_o(u, v_3)$ again at the end of the diffusion.

Several possibilities could be considered in mapping the decay and opportunistic selection into *ICOT* in practice. As mentioned above, user *Mike*'s promotion on Facebook for his new iPhone 6 will diffuse to all his followers, but whether and when they can be influenced and when and whether they would continue to pass this information to others are uncertain events. The decay and the opportunistic receiving selection phenomenon are very common in our daily life. Therefore, the model considers influence from both the receiving and accepting aspects is critical to capture the natural characteristics of influence diffusion in practice.

Theorem 1: The Influence Maximization Problem under the *ICOT* model is NP-hard.

Proof: The original influence maximization problem for the *IC* model is NP-hard. The *IC* model is a special case of the *ICOT* model with opportunistic acceptance ratio being constant 1 (without the effect of decaying function), and the threshold of opportunistic selection for the final step being constant 0. This leads to the hardness result of Theorem 1.

There are two choices: designing a heuristic algorithm which has no theoretical performance guarantee or an approximation algorithm with nice approximation ratio which can guarantee the solution results. Since influence maximization has been widely employed in

OSNs, a solution results in real cost. Thus, a better accuracy leads to a better profit for a company entity. In this chapter, we try to find a solution with a theoretical guarantee and incorporate various optimization strategies to improve efficiency.

Given function $\delta(\cdot) : 2^V \rightarrow \mathcal{R}$, the function is *monotone* iff $\delta(S_1) \leq \delta(S_2)$ whenever $S_1 \subseteq S_2$. Also, function $\delta(\cdot)$ is *submodular* iff $\delta(S_1 + x) - \delta(S_1) \geq \delta(S_2 + x) - \delta(S_2)$ whenever $S_1 \subseteq S_2 \subset V$ and $x \in V \setminus S_2$ where V is the set of the vertices.

As shown in [88], IC model is *monotone* and *submodular* which allows us to develop a hill-climbing-style greedy algorithm to achieve $(1 - 1/e - \epsilon)$ approximation ratio. Since the IC model is a special case of our $ICOT$ model, the objective function of $ICOT$ can also satisfy both monotonicity and submodularity. ■

Theorem 2: Influence function $\delta_{ICOT}(\cdot)$ is *monotone* and *submodular* under the $ICOT$ model.

Proof: We use the "*possible worlds*" semantic to prove the theorem. As shown in Fig. 5.2, the top graph $< v_1, u, v_2 >$ is a small fragment of the whole network (we use $\mathcal{G}$ to denote this uncertain graph fragment) and the four graph instances are possible world semantics generated from $\mathcal{G}$. For each possible world instance, based on the weight on each edge, each instance with different generation probability could be presented as a corresponding determined graph. All the possible world instances are generated by a cascade process. We could directly assume that before the cascade process starts, the outcomes for all the opportunistic selection and time decaying process have already been determined. For each possible world W_x, the existing probability is

$$P(\mathcal{G} \Rightarrow W_x) = \prod_{e \in E(W_x)} p(e) \prod_{e \in E(\mathcal{G})) \setminus E(W_x)} (1 - p(e)) \tag{5.7}$$

Specifically, each cascade step could be viewed as an individual coin-flip event with probability $f_o(u, v)$ which determines if u will influence v at the corresponding time t successfully or not. Since all coin-flip events are independent, a determined set of the coin-flip

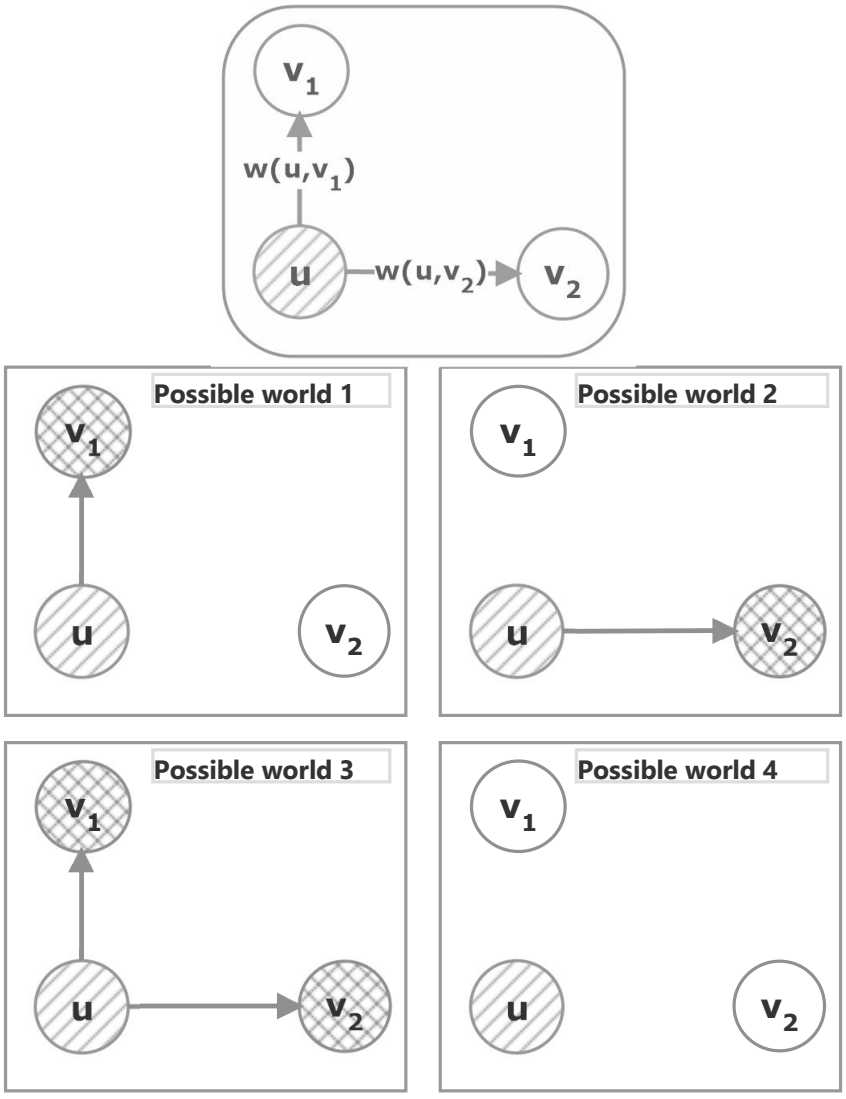


Figure 5.2. An instance of possible world sematic

events could be mapped to a *possible world* W_x. Assume there is an edge (u, v) in W_x, under the traditional IC model, without opportunistic selection and time decaying, u could directly reach v via one hop with probability 1. In the $ICOT$ model, to be more practical and accurate, u has to pass through opportunistic selection and decaying process when it tries to influence v. Since the time decaying process will not stop unless the distance between two nodes approaches to 0, it would be a limited process for opportunistic selection. On the other hand, node v is reachable from a seed set S if and only if there exists at least one path from S to v consisting of all active links (each node on the link is active). Let S_1 and S_2 be two arbitrary sets such that $S_1 \subseteq S_2 \subseteq V$. Since $\delta_{ICOT}(S)$ is the number of the nodes reachable from S in possible world W_x, if there is any node reachable from S_1, the active path will also be included in S_1's super set S_2. We can get the monotonicity of $\delta_{ICOT}(S)$.

For submodularity, based on Eq. 5.7, let all the probabilities related to our opportunistic selection and decaying process equal to 1. Different from IC, to take the decaying and delaying phenomenon into account, $ICOT$ tries to influence all the neighbors of activated nodes by $f_o(\cdot)$ for the last time (as accepting step) even no new activated node appears. Consider one instance of the accepting step of influence diffusion, the relationship between the number of neighbors in the last step and the number of nodes could be activated is just linear. If let the acceptance function $f_o(\cdot)$ equal to 0 at this point, IC and $ICOT$ could be unified. Considering node u reachable from $S_2 \cup \{w\}$ (w is another active node not in S_2) but not reachable from S_2, which means u is not reachable from S_1 either. Thus, w has to be the source of the active path to u, and u should be reachable from $S_1 \cup \{w\}$. For the margin increase for both S_1 and S_2, we have

$$\delta_{ICOT}(S_1 \cup \{w\}) - \delta_{ICOT}(S_1) \geq \delta_{ICOT}(S_2 \cup \{w\}) - \delta_{ICOT}(S_2) \tag{5.8}$$

Then consider the opportunistic selection and time decaying process, we have

$$\delta_{ICOT}(S) = \sum_{\mathcal{G} \Rightarrow W_x} Pr(W_x) \delta^{W_x}_{ICOT}(S) \tag{5.9}$$

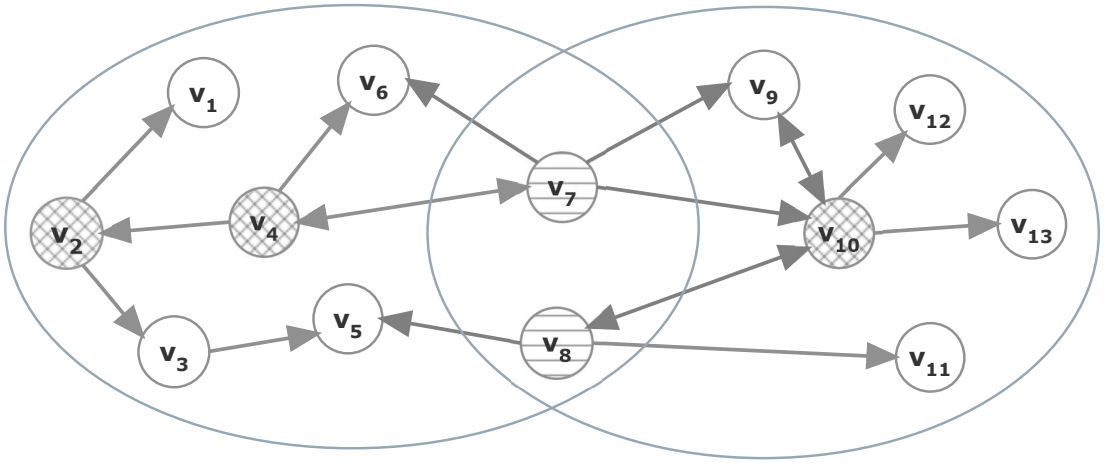


Figure 5.3. An example of social influence

Since $\delta_{ICOT}(S)$ is a nonnegative linear combination of $\delta_{ICOT}^{W_x}(S)$ which are monotone and submodular functions, $\delta_{ICOT}(S)$ keeps the same property, that is, submodular. ■

Based on the result of Nemhauser et al. [112], function $\delta(\cdot)$ suggests an approximate greedy algorithm with factor $1 - 1/e$. However, the hardness of computing $\delta(\cdot)$ for the IC model is $\#P$-hard[30]. If we apply the proof result to the $ICOT$ model, for a large scale network, even if a greedy approximate algorithm is applied by using Monte-Carlo simulations, the computation cost is still unacceptable. Considering the influence breadth, we apply a community detection algorithm [110] in the network to find different communities with overlap, then calculate the best influential k nodes taking both individual influence and global influence into account by applying a dynamic programming algorithm.

Our goal of influence maximization is to influence more nodes and larger area. In this case, besides the objective function $\delta_{ICOT}(\cdot)$, we take a further step to make influence diffusion as broad as possible.

Fig. 5.3 shows an example of the breadth of influence. The two circles represent two communities, and the influence is diffused according to the directed links. Assume we measure the influence by the number of outgoing links. Node v_{10} has the most outgoing links, and it should be selected in the next step based on the current measurement. Suppose

that the algorithm has selected the best $k-1$ influential nodes including v_{10}. If v_2, v_4, and v_8 provide the same influence increase, and v_2, v_4, and v_8 all have 3 outgoing links, since v_8 connects two different communities, v_8 has significant advantages than the other two, considering the breadth of influence.

Next, we discuss the *BICOT* model. Suppose network $\mathcal{G}$ has m communities $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$. The more communities the influence could cover, the broader influence this model could achieve. We borrow the similar idea in [105] mining structural hole spanners in a network. Different from structural hole spanners which only consider the minimal value of user's importance scores in different communities, we try to find the nodes that maximize the influence globally and affect as more communities as possible. Formally, let N_c be the number of communities the algorithm could cover under *ICOT*.

Intuitively, we expect the node's individual influence in its community to be similar to its influence in the whole network. Although the gap between local community and global influential node sets exists, as the monotone we proved, the influence diffusion is built on unit node activities from local to global. The social network is strong community-based organization, and the influential node set in local from a very large extent represents the global result. We try to find the best k influential seeds in each community first, then by comparing the difference between local and global, we iteratively fill the gap by further optimization algorithms. Let $P_C(v)$ be the number of communities node v influenced divided by the number of all communities, and $S \subseteq \mathcal{C}$ denotes the subset containing more than one community, then a utility function $Q(\cdot)$ is defined for each node to measure its contribution in maximizing the influence breadth. Let $A(v, S)$ be the structural score of v in S.

$$Q(v, C_i) = \max_{e_{u,v} \in E, S \subseteq \mathcal{C} \wedge C_i \in S} \{P_C(v) Q(v, C_i),\ \alpha_i Q(v) + \beta_S A(v, S)\} \tag{5.10}$$

$$A(v, S) = \min_{C_i \in S}\{Q(v, C_i)\} \tag{5.11}$$

In Eq. 5.10, α_i and β_S are two tunable parameters. The contribution function $Q(\cdot)$ is computed as the combination of the importance score of v's friends and the structural score of v itself. Since $Q(\cdot)$ is the influence measurement of individual node, we use the famous PageRank [113] to initialize score $i(v)$ for each node v in each community, then continue the iteration until the converge based on the two reinforce Eq. 5.10 and Eq.5.11 stable. Same as [105], for all the node v not belongs to community C_i, we set their influential score to 0, that is:

$$Q(v, C_i) = \quad i(v), v \in C_i \tag{5.12}$$

$$Q(v, C_i) = \quad 0, v \notin C_i \tag{5.13}$$

Theorem 3: For α_i and β_S, the function scores of $Q(v, C_i)$ and $A(v, S)$ exist for any graph if and only if,

$$\max_{C_i \in S}\{\alpha_i + \beta_S\} \leq P_C(v) \tag{5.14}$$

Proof: Suppose community $C_i \in \mathcal{C}$ and $C_i \in S$ such that $\alpha_i + \beta_S > P_C(v)$. Considering nodes v_1 and v_2 which connected to each other with the PageRank score $i(v_1) = i(v_2) = 1$, where $v_1 \in \cap_{C_j \in S} C_j$ and $v_2 \in C_i$. We have $Q(v_1, C_i) = P_C(v_1)$. Then by Eq. 5.11, $A(v_1, S) = \min_{C_i \in S}\{Q(v_1, C_i)\} = P_C(v_1)$. According to Eq. 5.10, $P_C(v_1)Q(v_2, C_i) \geq \alpha_i Q(v_1, C_i) + \beta_S A(v_1, S) = P_C(v_1)(\alpha_i + \beta_S) > P_C(v_1)$, which means product of two positive fraction is larger than one of the fractions, which is impossible.

For the if direction, $\{\alpha_i + \beta_S\} \leq P_C(v)$. Suppose in the first iteration $Q^0(v, C_i)P_C(v) \leq P_C(v)$ and k-th iteration later $Q^k(v, C_i)P_C(v) \leq i(v)P_C(v) \leq P_C(v)$. In the $(k+1)$-th

iteration, for each $C_i \in S$, we have $Q^{k+1}(v, C_i)P_C(v) \leq \alpha_i Q^k(u, C_i) + \beta_S A^k(u, S) \leq P_C(v_1)$.

■

We narrow the bound of the result in [105] α and β from $\{\alpha_i + \beta_S\} \leq 1$ to $\{\alpha_i + \beta_S\} \leq P_C(v)$. We also improve the performance of the $ICOT$ model by incorporating the number of communities which can be globally covered by one node.

Algorithm 6: Iteration algorithm

Input: Graph G, α_i, β_S, and convergence threshold ϵ
Output: Function convergence result $Q(v, C_i), A(v, S)$

Initialize $Q(v, C_i)$ according to Eq. 5.12
while $\max |Q'(v, C_i) - Q(v, C_i)| \geq \epsilon$ **do**
  for $v \in V$ **do**
    for $C_i \in \mathcal{G}$ **do**
      $t(v, C_i) = \max_{C_i \in S}\{\beta_S A(v, S) + \alpha_i Q(v)\}$
    if $u \in N(v)$ & $t(u, \cdot) \neq t'(u, \cdot)$ **then**
      /* t' is the previous value of t which monitors the change of v's neighbors */
      for $v \in V$ **do**
        for $C_i \in \mathcal{G}$ **do**
          $Q'(v, C_i) = \max_{C_i \in S}\{P_C(v)Q(v, C_i), \max\{t(v, C_i)\}\}$
        for $C_i \in \mathcal{G}$ **do**
          $A'(v, S) = \min_{C_i \in S}\{Q(v, C_i)\}$
  Update $Q = Q'$ and $A = A'$

As shown in Algorithm 6, through finite iterations we can get a rank of all the nodes based on their own ability to influence others within their communities. By the configuration of parameters α and β, we can control the balance of influence depth and influence breadth. Let $r(v, C_i)$ be the rank of node v in community C_i, and $Rank(v, C_i)$ be the rank of node v in the network.

$$Rank(v) = \frac{\sum \frac{r(v,C_i)}{|C_i|}}{\text{Number of communities involving } v} \times 100\% \tag{5.15}$$

By Eq. 5.15, we assign a percentage value $Rank(v)$ with a control parameter φ to each node

v, and calculate the influence spreading process on each edge by $\varphi Rank(v)w(\cdot)$. Thus, we can conclude our *BICOT* shown in Eq. 5.3.

5.3.2 Algorithm

The difference between *ICOT* and *BICOT* is whether taking breadth as a measurement for influence. Besides breadth, we adopt heuristic strategies in [31] in terms of a dynamic programming algorithm for both models. First, we detect communities in a network allowing overlap between different communities. Second, Algorithm 6 is applied to get the rank of each node. Through parameter φ, we control the balance of breadth and depth. Then, consider the updated weight of each node. We incorporate the strategies in [31] to model to find the seed set.

In [31], Chen et al. designed a heuristic strategy which builds a tree-like structure for influence. Then influence spreading path is maximized through a greedy algorithm. We use the same idea, but our model considers the opportunistic selection and influence ability decrease over time. When calculating and finding the seeds which have the largest incremental result in *ICOT* and *BICOT*, if the margin increases less than or equal to $\tilde{T}_o$, we regard this path as disconnected. The algorithm for *BICOT* is shown as follows:

Algorithm 7: Algorithm for model *BICOT*

Input: Graph G, α_i, β_S, ϵ, φ, τ and $\tilde{T}_o$
Output: Seed set for maximizing influence $S^{\dagger}$

1 Do community detection by Algorithm 1 from;
2 Algorithm by 6(α_i, β_S, ϵ) to get the value of $Rank(\cdot)$ by Eq. 5.15 for each node;
3 By parameter φ with Eq. 5.15 to control the tradeoff between influence breadth and depth;
4 Calculate the influence maximization seed set based on the *BICOT* model with parameters τ and $\tilde{T}_o$;

For model *ICOT*, we only consider the opportunistic selection and time delay, reducing the step for calculating the influence breadth for each node (Lines 2, and 3 in Algorithm 7). Then the seed finding process does not need to be incorporated with Eq. 5.15. The detailed

Table 5.2. Dataset for description

Data	Nodes	Edges	Diameter
Amazon0302 (A1)	262111	1234877	29
Amazon0312 (A2)	400727	3200440	18
Amazon0505 (A3)	410236	3356824	21
Amazon0601 (A4)	403394	3387388	21

algorithm is ignored due to space limitation.

5.4 Experimental Study

5.4.1 Data and Observations

We perform the experiments forwards the following data sets.

Epinions [3] is a Who-trust-whom network, where nodes are members of the web site and a directed edge from user u to v means u has the influence to v (v trusts u). The network includes 75,879 nodes and 508,837 edges.

Twitter [4] is one of most notable micro-blogging services. Twitters can publish tweets. We use the dataset obtained from [79]. The subnetwork includes 112,044 nodes (users of Twitter), and 468,238 edges (following relationships) and 2,409,768 tweets posted by them.

Inventor is a network of inventors, obtained from [132] extracted from USPTO[5]. The network consists of 2,445,351 nodes and 5,841,940 edges (co-inventing relationships).

Amazon Dynamic Networks

Table 7.3 is derived from the *Customers Who Bought This Item Also Bought* feature of the Amazon website. The four networks were from March to May in 2003. The connection is established in a network from i to j if product i is frequently co-purchased with product j [95].

Fig. 5.4 shows the average degree of all the seven data sets. The probability of each edge is learned from the networks in later time, which means the probabilities of the first

[3]http://www.epinions.com/
[4]http://www.twitter.com
[5]http://www.uspto.gov/

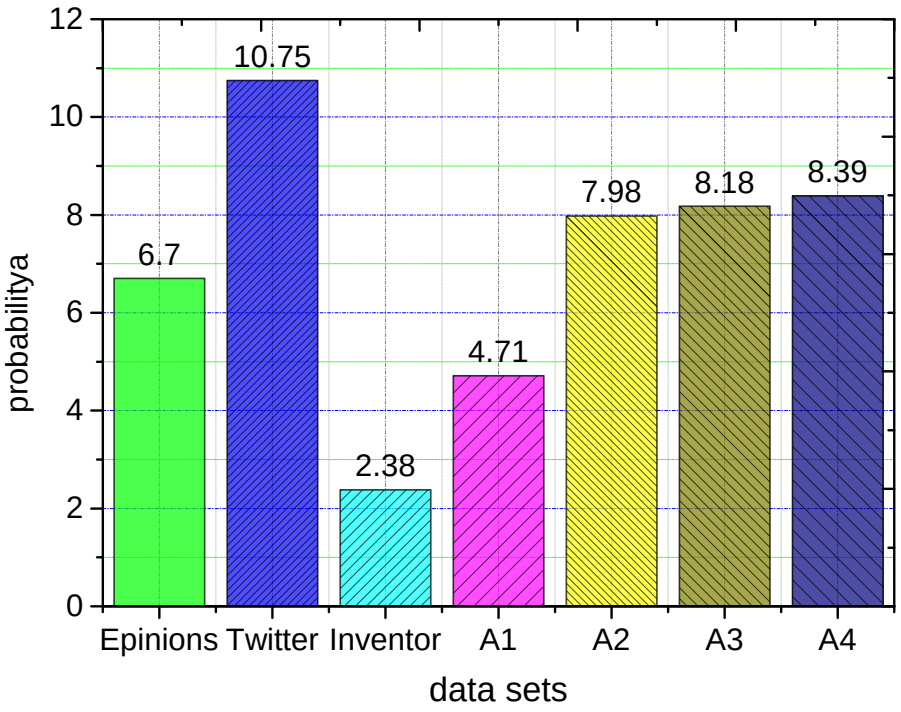


Figure 5.4. Average degree of data sets

network come from the second one, and the probabilities of the last network come from the first three networks based on the linear prediction. The probability distribution of the four networks from Amazon is shown in Fig. 5.5. As shown, the probability distribution of 4 Amazon networks is mainly in the range of 0.02-0.05. The reason for this range is the social characters of the relationship based on the co-purchased network. And this probability distribution also shows that the Amazon co-purchased are overall loose networks. Most research literature assumes that the probabilities or the weights on links and the thresholds are given. However, as pointed out by Goyal et al. [49], learning those probabilities and thresholds is a non trivial problem. Therefore, we use a learning algorithm on the raw input data [122] to get the balance between complexity and practicability. For the Amazon data set, since there are a series of snapshots of the networks, we generate the real influence spreading trend by comparing our model to the real learning algorithm [50] which initially treats the data as a user log then solves the influence maximization problem.

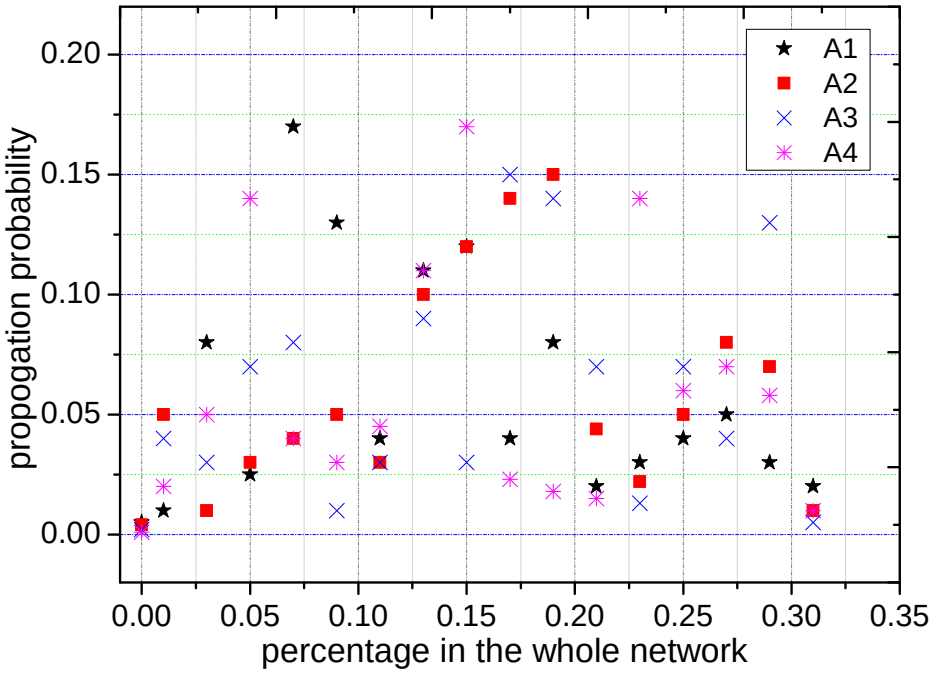


Figure 5.5. Probability distribution of 4 Amazon networks

5.4.2 Experiment Result

All the codes are implemented in C++, and all the experiments are performed on a PC running Ubuntu 14.04 LTS with Intel(R)2 Quad CPU 2.83GHz and 6GB memory.

We examine how the parameters affect influence spread in Algorithm 6. As shown in Fig. 5.6 and Fig. 5.7, the performance of Algorithm 6 is insensitive to the variation of α and β. Consider the difference between two networks Epinions and Twitters, the average degree is 6.7 for Epinions, and 4.17 for Twitter. Thus, from experiment result, the main factor affect parameter α and β is the sparsity of the network. Regardless of network Epinions or Twitter, when we increase parameter α from 0.05 to 0.45, and β from 0.2 to 0.6, the range of influenced proportion in both networks is less than 2%. And for both networks, we get the best performance when α is approaching to 0.2 and β is approaching to 0.35. The optimum points of α and β are based on experiment, but the insensitive of these two parameters give the algorithm more robustness and stability in practical.

We evaluated the number of the influenced nodes under different models. As shown in

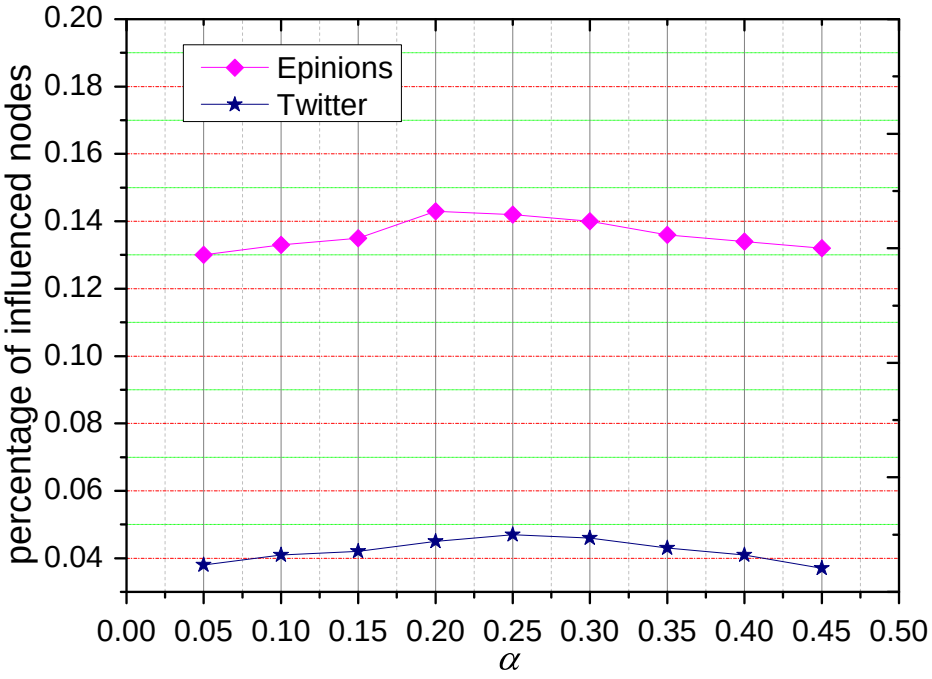


Figure 5.6. Effect of α for influence diffusion

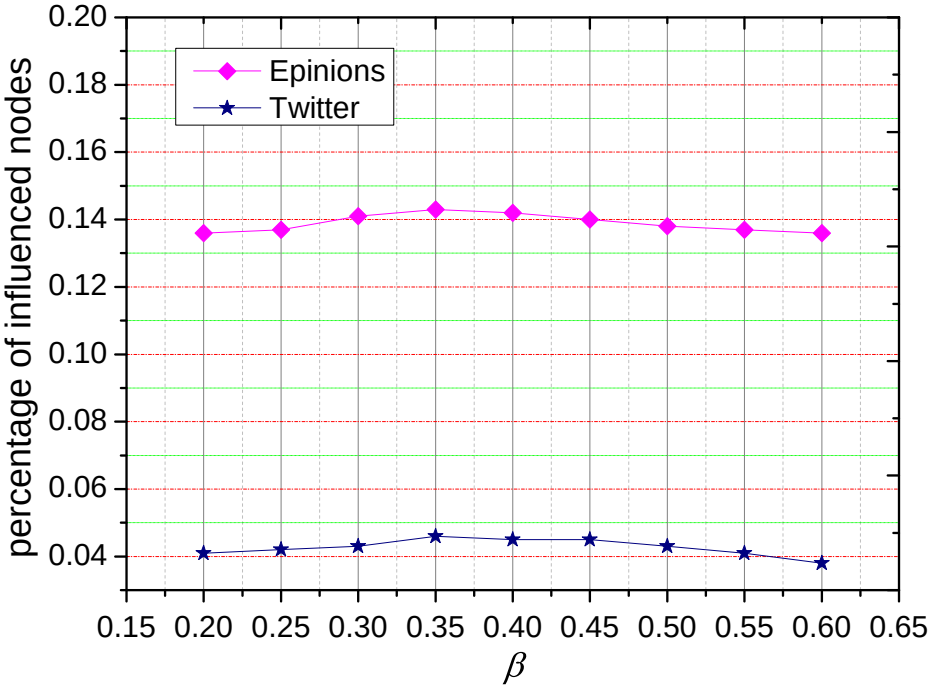


Figure 5.7. Effect of β for influence diffusion

Fig. 5.8, Fig. 5.9, and Fig. 5.10, we compared the traditional *IC* model [30] with our two models on the three static networks. We compared the performance of our algorithm with the classical *IC* model in Epinions, Twitter and Inventor, three different real networks. The vertical coordinates are the number of influences nodes which resulted from classical *IC* and our algorithm *ICOT* and *BICOT*. With the increase of the size of seeds set k, *ICOT* and *BICOT* could reach and continue to be at least 87.5% of the performance of *IC*.

Compare with the experiment results in Twitter, and Inventor, when seeds set k equals to 50, the number of influenced nodes in Epinions of *BICOT* and *IC* has the biggest difference. This phenomenon is because the inner topology of Epinions is closer to real life social connection based on Who-trust-whom but not like typical online social network which is more powerful and more compactly to spread information. The experimental results show that *BICOT* is more applicable to online social network with more connection but not the real life connection. Even though, *BICOT* could also achieve good enough performance (about 87.5%) compare with classical *IC* model. From the three plots, we can see that the proposed model on the static network shows very similar trend like the traditional *IC* model. Because our model includes the optimistic selection and time decaying process, it is hard to be comparable with other traditional models if only consider the number of influenced nodes. To model the real life influence more accurate, we also proposed a method to calculate the final influence expectation which include more nodes when the influence spread process ends. We set the default value of $\tau = 0.5$ giving the influence breadth and depth the same weights.

With the increase of the size of seeds set k, *ICOT* and *BICOT* could reach and continue to be at least 87.5% of the performance of *IC*. Compare with the experiment results in Twitter, and Inventor, when seeds set k equals to 50, influenced nodes in Epinions of *BICOT* and *IC* has the biggest difference. This phenomenon is because that Epinions is more close to real life social connection based on Who-trust-whom but not like typical online social network which is more powerful to spread information. By the experimental result, we could find out that *BICOT* is more applicable to online social network with more connection but not the real life connection. Even though, *BICOT* could also achieve good enough performance

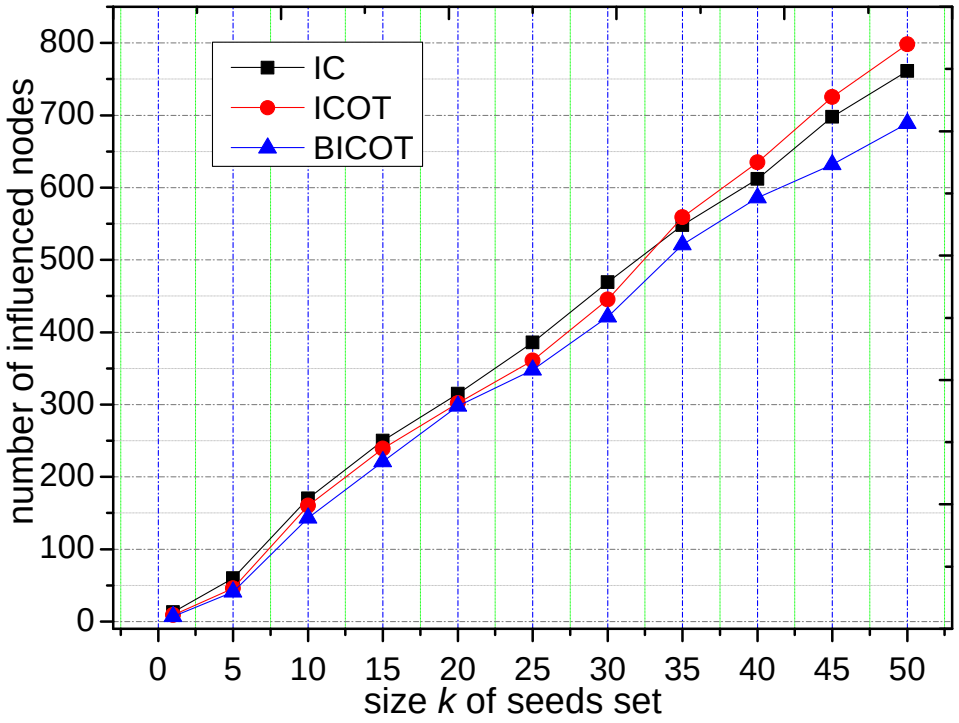


Figure 5.8. IC VS ICOT VS BICOT in Epinions

(about 87.5%) compare with classical *IC* model.

From the three plots, we can see that the proposed model on the static network shows very similar trend like the traditional *IC* model. Our models consider the optimistic selection and time decaying. We also proposed a method to calculate the final influence expectation which include more nodes when the influence spread process ends. We set the default value of $\tau = 0.5$ giving the influence breadth and depth the same weights.

To show our contributions in a convincing way, we compare our model with the up-to-date experiment based algorithm in [50] on the aspect of the real influence spread. We run our algorithm on the first Amazon co-purchase network, and run Goyal's algorithm called *CD* based on the four networks since their algorithm requires users' log. Meanwhile, we compare with traditional *IC* model towards on Amazon network 1 and network 4. As shown in Fig. 5.11, although all the curves follow similar trends, for a larger k, *CD* which is based on learning has slower increase which is more practical since it learns the knowledge from four data sets. Apparently, our models are more approximate to model *CD* which means

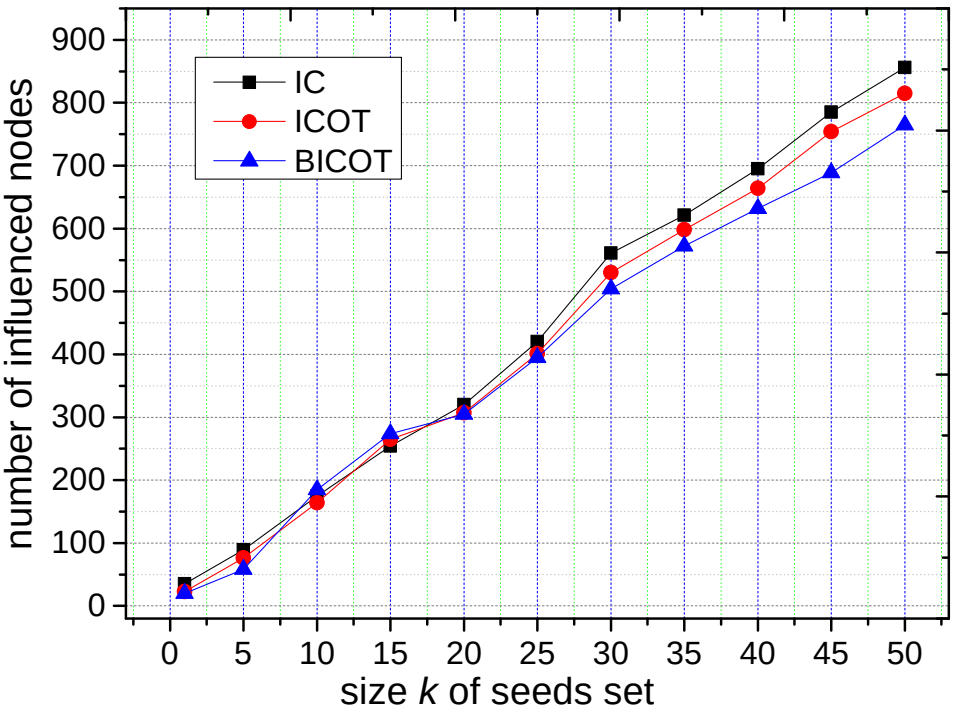


Figure 5.9. IC VS ICOT VS BICOT in Twitter

that our models are closer to the influence spreading in practice.

We compare the performance of our algorithm *ICOT* and *BICOT* with other classical algorithms in Fig. 5.11. Quantitatively, if k equals to 50, our algorithm *ICOT* and *BICOT* could influence 662 and 659 nodes respectively in the network. Compared to 665 influenced nodes by algorithm *CD*, our algorithms could approach more than 99% similarity in the number of influenced nodes.

If only from the aspect of influence depth, our algorithm is not better than classical *IC*, but our algorithm actually provides a way to control the balance between influence depth and breadth. As we'll show in the next experiment that the overall performance of *BICOT* is much better than the classical approaches.

Contrast to Fig. 5.11, Fig. 5.12 shows the number of the communities covered by each algorithms. Clearly, our *BICOT* covers much more communities than the *IC* and *CD*. The advantage of our model is as well as we have a similar result of influence maximization follow the real diffusion, community-based algorithm gives a much better efficiency to the influ-

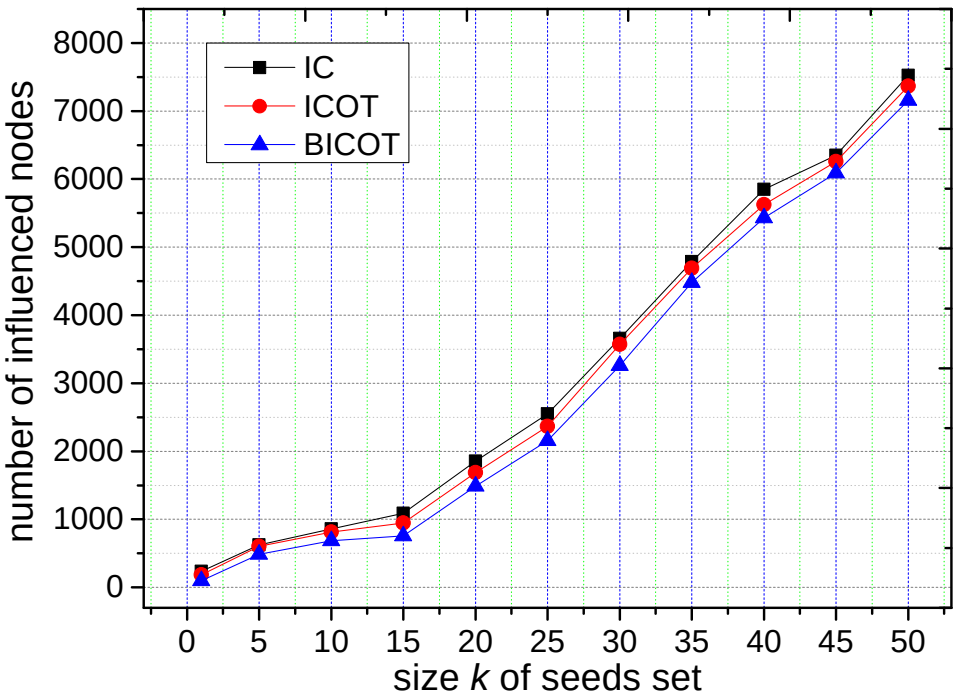


Figure 5.10. IC VS ICOT VS BICOT in Inventor

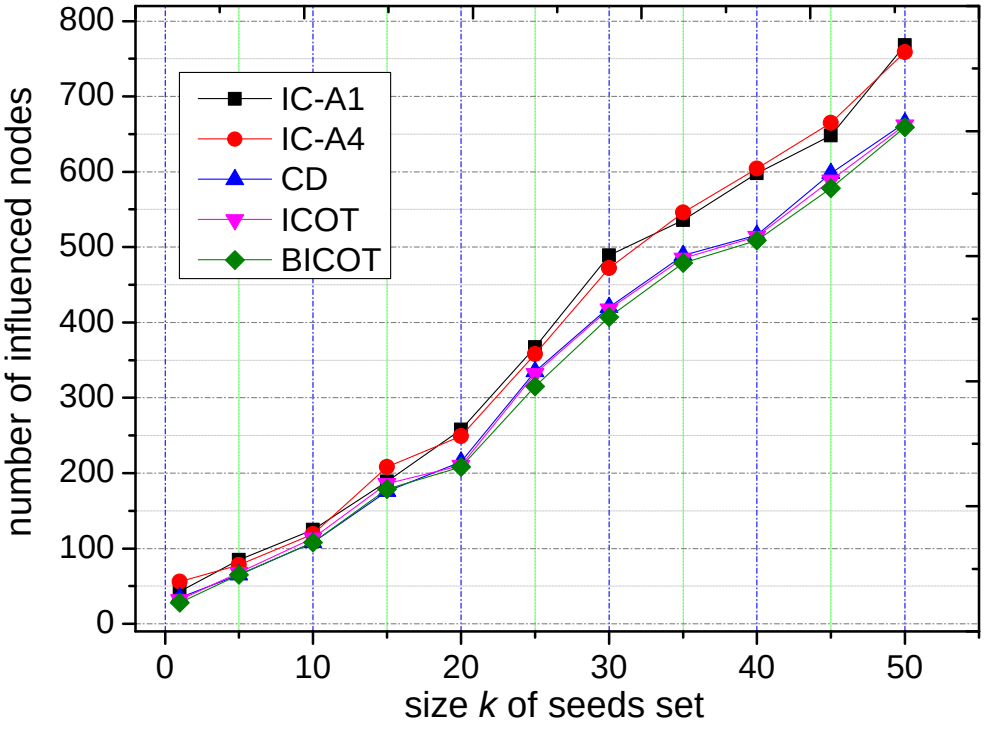


Figure 5.11. Influence spread by different algorithms

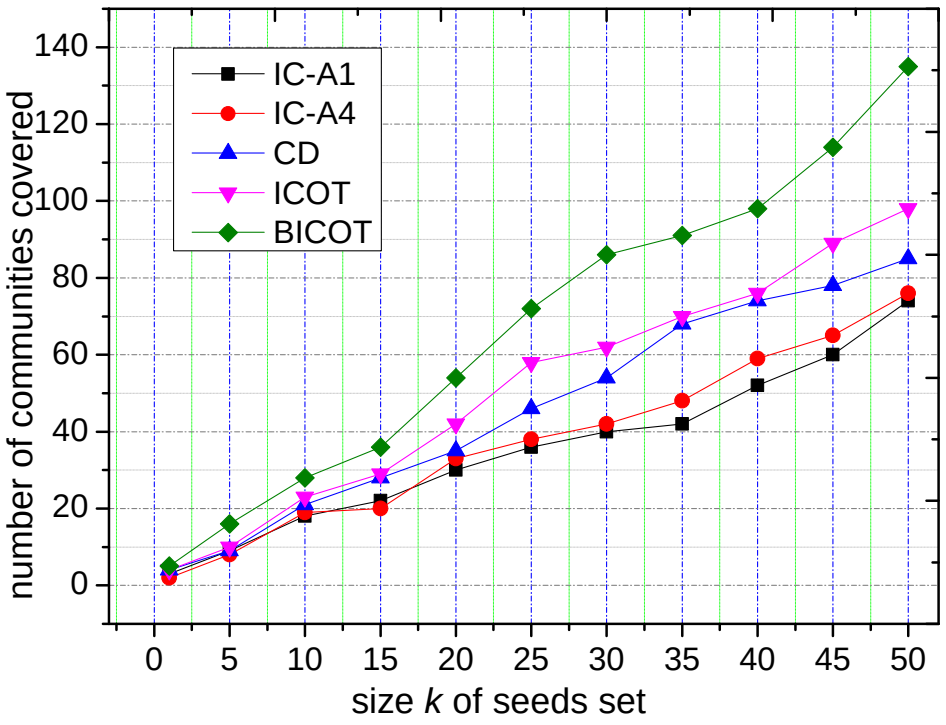


Figure 5.12. Communities covered by different algorithms

ence maximization problem. Further more, our model cover more communities indicating a broader influence diffusion.

To evaluate the relationship between influence depth and breadth, we change parameter φ from 0.1 which cares more about influence depth to 0.9 which emphasizes more on the breadth.

Fig. 5.13 shows the influence spread for different φ. We can see that as φ increases, the influence is decreased. This decrease is because of the definition of our objective function, we care more about breadth than depth. With the same parameter setting, we can derive from Fig. 5.14 that although the influence spread has been reduced, the number of the communities covered by our algorithm is increased.

In more detail, Fig. 5.13 is the number of influenced nodes and Fig. 5.14 is the number of communities covered by the influence for both Epinions and Twitter. In the situation that φ equals to 0.5, which is the balance point for both influence depth and breadth, the number of influenced nodes is 689 in Epinions and 796 in Twitter, while the communities

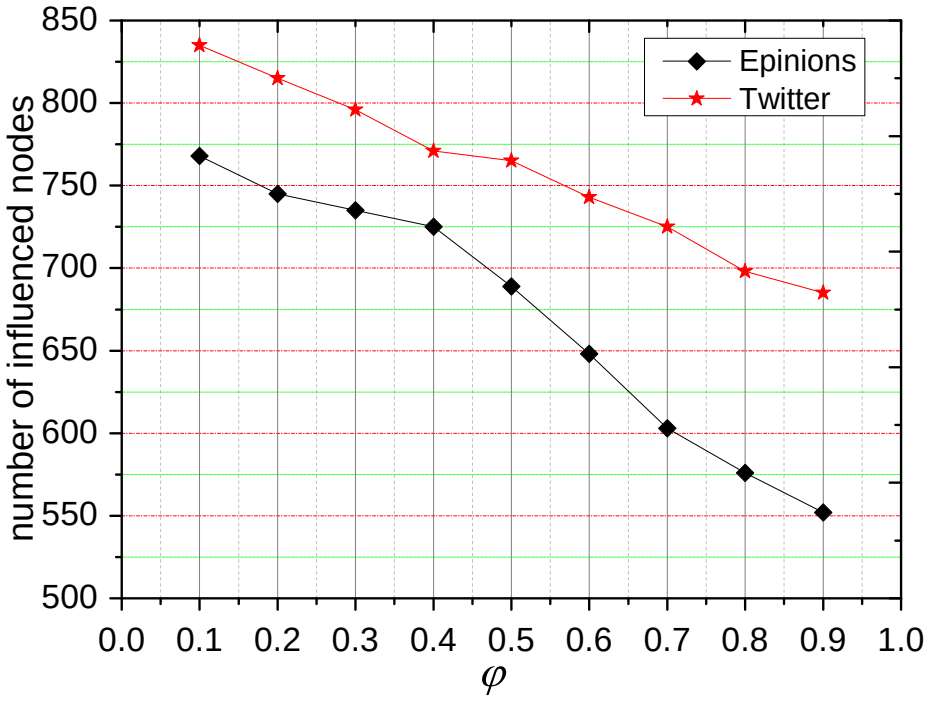


Figure 5.13. Influence performances for different φ of $BICOT$

covered by the influence is 78 and 96 respectively. If we just increase φ from 0.5 to 0.7, in Epinions, we could find that the number of communities covered by the influence increase to 89, representing 14.1% of growth, despite that 76 influenced nodes are gone. Such increase in the number of covered communities demonstrates that the breadth of the influence increased significantly. We can get a similar result in Twitter, which increased about 13.5% of growth in the breadth of influence.

We do consider the comparison between the traditional models and ours, Fig. 5.11 and Fig. 5.12 contrast the performance between our models $ICOT$ and $BICOT$ and classical models include IC, CD, and $ICOT$. The reason we do not compare to other algorithm is that in traditional approaches such as IC, CD, *etc.*, the influence breadth is not considered. There is not a parameter φ control the trade-off between breadth and depth in traditional models.

The parameter φ is the controller to influence depth and breadth in algorithm $BICOT$. The vertical coordinate in each of Fig. 5.11 and Fig. 5.12 is the number of influenced nodes.

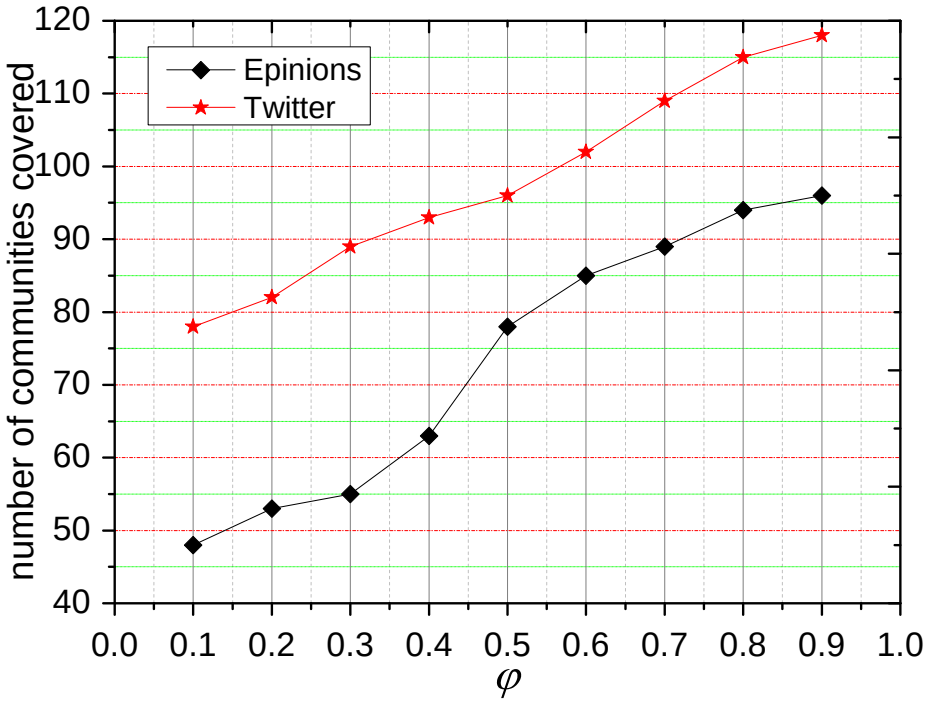


Figure 5.14. Communities covered for different φ of $BICOT$

With the increase of the size of seeds set k, which is the number of originally activated nodes, more nodes are influenced in the whole network. This is the situation similar like distribute trial product samples in a shopping mall. The more free trial samples (could be small bag of shampoo), the better advertisement effect could achieve. But if consider the breadth of influence, in Fig. 5.12 we could get that algorithm $BICOT$ achieve much better breadth performance as covering 135 communities compare to CD 85 communities, and IC just around 75 communities coverage. Under the circumstances of keeping the same depth in influence with CD, $BICOT$ could reach a result of 58.9% higher than CD in the breadth of influence.

In brief, empirical studies on different large real-world social networks show that our model demonstrates that high depth influence does not necessarily imply broad information diffusion. Our model, together with its solutions, not only provides better practicality but also gives a regulatory mechanism for influence maximization as well as outperforms most of the existing classical algorithms.

5.5 Summary

In this work, based on the observations from real data and application, we propose model *ICOT* which incorporates both diffusion decay and opportunistic acceptance selection for dynamic networks. In addition, we develop model *BICOT* to control the balance between influence depth and breadth. We take the first step to explore the potential of broad influence maximization. Through comprehensive experiments results, we show that our model can achieve a comparable influence diffusion result like the learning-based algorithm which has a more strict input requirement, and our models have a broader influence coverage.

Chapter 6

PRIVACY RESERVED INFLUENCE MAXIMIZATION

6.1 Introduction

Finding the most influential roles within a network, and letting these influential roles promote one product to the market or spread one view point into the society could be considered as the problem of influence maximization. Targeting at maximizing the propagation of a new item or an opinion through social networks by word-of-mouth effect with specific initial budgets is the basic assumption of influence maximization. As a very classical optimization problem problem, influence maximization, has attracted considerable attentions from different research areas including network analytic, data mining, and sociology.

However, the sensed data with the location information from wearable devices, smart mobile phones, and many other resources in cyber physical world [47, 23, 72, 142, 175, 73, 169, 71], which is still a new area for our researchers to explore, has not been flipped and understood well. So far, the research in connection with the sensed information from cyber physical world has not been invested well in the research of influence maximization. With the development of modern mobile devices including smart phones, touchable mobile computing devices, and wearable smart devices *etc.*, a huge amount of sensed data is generated. And the combination of sensed data and online social data will become the next "blue ocean". Huge opportunities from both cyber physical world and online social network are coming for us to employ. For example, Google reported that in January 2016, the amount of revenue of the Android operating system has reportedly reached $31 billion and this number is still increasing day and night. Utilizing both cyber physical sensed information and online information could generate and develop a lot of applications for smart car, smart city, and smart planet [1]. Apparently, combining both sensed cyber physical data and online data

[1] https://www.whitehouse.gov/the-press-office/2015/09/14/

could greatly enhance the model's ability of influence maximization in real life and solving the problem practically. In the meanwhile, few researches succeed in exploring the potential of influence maximization by taking into consideration of both online data and data from physical world. However, the integration of both cyber-physical sensed information and online social information is not a straightforward method and includes many challenges including but not limited to following:

- Sensed data is very complicated and therefore it is much more difficult to collect from cyber physical world. For example, the friendship connection tend to be geographically related in the real world, but the geographical similarity is not presenting directly but hiding invisible.

- The sensed cyber physical network and online social network are different systems, there is not a consistent one-to-one corresponding relationship between each other. Much more efforts are required to integrate the information from both systems. Especially in some special cases where there is an instance in the sensed cyber physical network while there is not available one in online social net work. In this case, the formulation and the integration process require much more attentions to deal with the result.

- The sensed data from cyber physical world is quite private. The privacy issue during the analysis of sensed data should be put on the agenda. The identity and location information of users should be protected well in the application we developed. However, the balance between utility and privacy is still a very challenge question in influence maximization applications.

- Maximizing the influence in an online network has never been a trivial problem. The difficulty of maximizing influence and the challenge of integration would still affect the performance of the framework. Furthermore, how to reduce the computational cost is another challenge.

In this work, we combine both the sensed location data from cyber physical world and online social network together to construct a comprehensive heteroecious network. Within the heteroecious network, we tried to further explorer the potential of sensed data to solve the classical influence maximization problem. To start with, we integrate the sensed cyber-physical data to supply the online social network with more real activity in cyber world. This strategy could allow the cyber social activity be considered while the inactive users might be ignored in online social networks. This combination of both sensed cyber physical knowledge and online social knowledge could also help the cyber physical world make their decision in an optimal strategy with the concern of the online activities. Besides, our objective is not only proposing a framework to help the applications maximize the influence, but also consider the users' privacy as an important preserved issue. In this sense, we understand the importance of privacy for any public user in both online or offline physical world. Our model and resolution especially deals with the privacy issue of user's location information.

To cope with each challenge above, we have the following contributions:

- We first unveil the hidden connection from sensed location data in cyber-physical world by considering the geographical similarity, and merging the connection information into the existed online social network.

- A random walk procedure is employed to construct the heterogeneous network. A framework combining both cyber-physical and online social network data together is proposed. The advantages of both online and sensed cyber world are taken by the framework.

- To protect user's sensitive location information, we employ the classical differential privacy mechanism. The differential privacy could preserve the location privacy and other sensitive link privacy with a guaranteed ratio.

- We remodel the problem of influence maximization under the heterogeneous framework developed. To maximize the influence for further extended applications, we propose several optimized strategies to improve the performance of the algorithm.

– Two large scale real life datasets are tested in the experiment section. The real experiment result shows that the power of engaging sensed location information into online social network could significantly improve the performance of influence maximization.

The rest of this chapter is organized as follows. Section 6.2 presents the preliminaries and illustrates the theoretical analysis. Real life networks and synthetic network are evaluated in Section 6.3. Section 6.4 concludes our work.

6.2 System Model and Algorithm

We introduce the model of the heteromerous network and the problem definition in the following section. We first describe the sensed location record pattern in the sensed cyber physical network; then we introduce a framework could integrate both sensed cyber physical network and online social network; in the end of this section, we give the preliminary of sensed location privacy and protection mechanism illustration.

6.2.1 Model of Sensed Cyber Physical Location Pattern

In order to better understand the hidden information in the sensed cyber physical location trajectories, we propose four different geographically patterns according to empirical observations, and formula design a function to formulate the potential relationship according to the sensed location records. Fig. 6.1 is the four different typical cases in sensed cyber physical location records. In all four cases, two user a and user b are enrolled. We tried to model their sensed location records to different behavior patterns as following:

Pattern 1: In the first pair of users, if user a and user b have a matched common location A in their sensed location segments, they might have some similarity with a certain probability. Let $FCL_{\alpha_l}(a, b)$ be the function represent the frequency of common stop pattern with the threshold α_l.

Pattern 2: The second pair of users, as shown in the same figure, if user a and user b share similar sensed trajectory records from Location A to Location B, they might have a

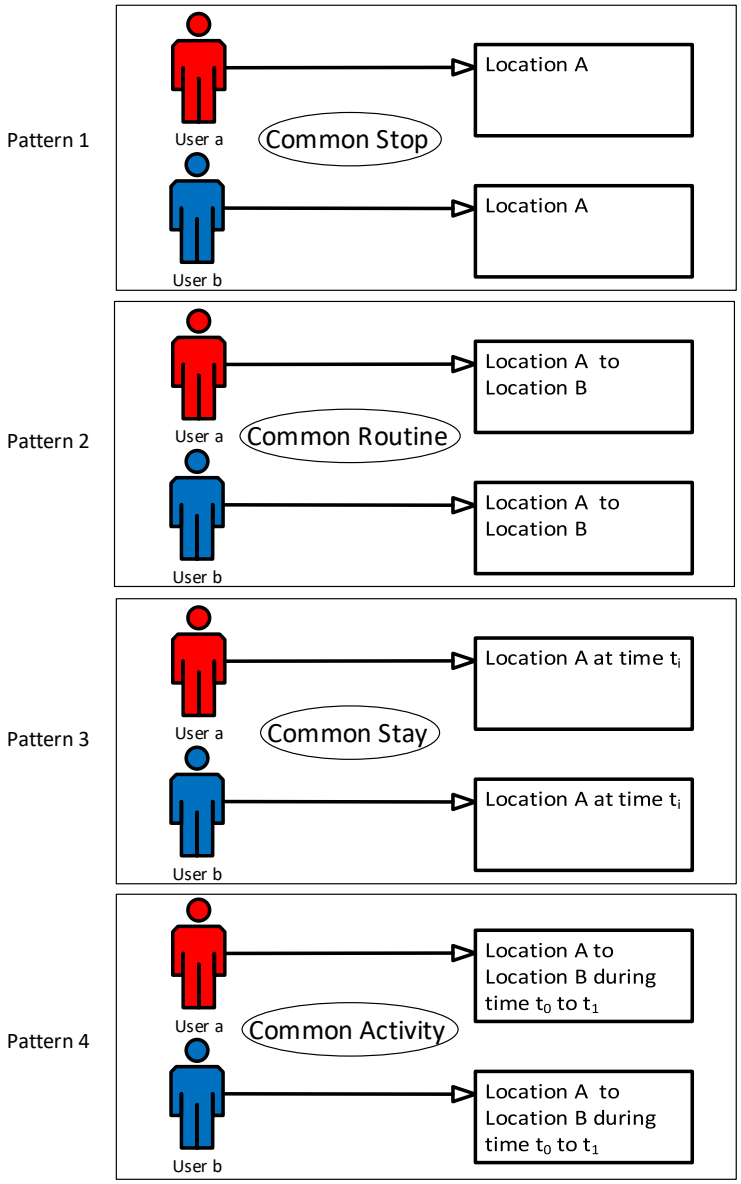


Figure 6.1. Four Different Cyber-Physical GPS Patterns

higher probability to be friend in real life. $FCR_{\alpha_r}(a,b)$ is the number of how many common routines exist with threshold α_r.

Pattern 3: As shown in the third pair of users, if two users share same sensed location record with same time point t_i, they may met each other at the time point. The probability that they are connect might be higher. $FCS_{\alpha_s}(a,b)$ is the number of times for common stay pattern appeared.

Pattern 4: In the last situation, the user pair at the bottom in Fig. 6.1, two users share the same sensed location records with a specific period from t_0 to t_1. This case would be a very high possibility to build a connection between the two users a and b, therefore, the existence probability of their connection would be much higher. The function $FCA_{\alpha_a}(a,b)$ for pattern 4 represent the common activity.

Based on the observation we illustrated and the frequency functions we developed above, we use the sensed location information from cyber physical world to build the cyber physical network.

Definition: Sensed cyber physical relationship is denoted by the CPF ratio r_{CPF} that larger than a threshold α, where the r_{CPF} is defined as

$$\begin{aligned} r_{CPF} = \quad & \beta_l FCL_{\alpha_l}(a,b) + \beta_r FCR_{\alpha_r}(a,b) \\ & + \beta_s FCS_{\alpha_s}(a,b) + \beta_a FCA_{\alpha_a}(a,b) \end{aligned} \tag{6.1}$$

In Equation 6.1, the parameter $\alpha_l, \alpha_r, \alpha_s, \alpha_a$ are the four thresholds corresponding to different four pattern function, and parameter $\beta_l, \beta_r, \beta_s, \beta_a$ are the adjusted weight to control the impact of each pattern for a sensed physical connection. The value of $\alpha_l, \alpha_r, \alpha_s, \alpha_a$ is set to 2 by default. This means that if the user a and user b who has any kinds of common behavior above appear more than twice, we consider the user a and b has the corresponding pattern (potential relationship) and their frequency count is 1. It's worth to mention that, all parameters are developed to model the user behavior pattern in a practical

way. It is still very challenge to quantitatively count the the behavior pattern and the relationships among the sensed cyber physical world directly and mapping back to the online social network. Intuitionally, the proposed four behavior patterns are based on observation. Along with the different coincidence of each pattern, we believe there is an incremental impact on a relationship. The users's behavior matching Pattern 4 would have a closer relationship compare to Pattern 3, Pattern 2, and Pattern 1. In the meanwhile, Pattern 3 is a stronger pattern compare to Pattern 2 in real life situation. Therefore, we manually assign a heuristic number 1, 2, 4, 8 for parameter $\beta_l, \beta_r, \beta_s, \beta_a$ to represent the contribution for each different behavior pattern. Unfortunately, all persuasive ethological result existed for the correlation among different behavior pattern we modeled have to be updated according to the domain knowledge. Based on the observation and learning process, heuristically, these four parameters could be adjusted according to different application background if our model is referenced.

6.2.2 Heterogenous Network Model

As shown in Fig. 6.2, the data collected from both sensed cyber physical world and online social world are integrated into the new heterogeneous network at the bottom. To be noticed, if two users belong to both sensed cyber physical world and online social world, we combine the same instance to one node the constructed heterogeneous network. But for the situation that one user does not exist in any one of the resource world, we will leave a user in the new network to carry the information from the original world the node come from.

To integrate the sensed cyber physical world and online social world together, we build the sensed cyber physical network $G_c(V_c, E_c, P_c)$ and online social network $G_o(V_o, E_o, W)$, where P_c represents the links of two nodes in the sensed cyber physical network according to the Equation 6.1. W represents the collected connection information from the online social network. V_c, V_o are the nodes and E_c, E_o are the links set of sensed cyber physical world and online social world, respectively. The objective is building the heterogenous network $H(V, E, P)$, where $V = V_c \cup V_o$, $E = E_c \cup E_o$. Finally, let P represent the set of each vertices

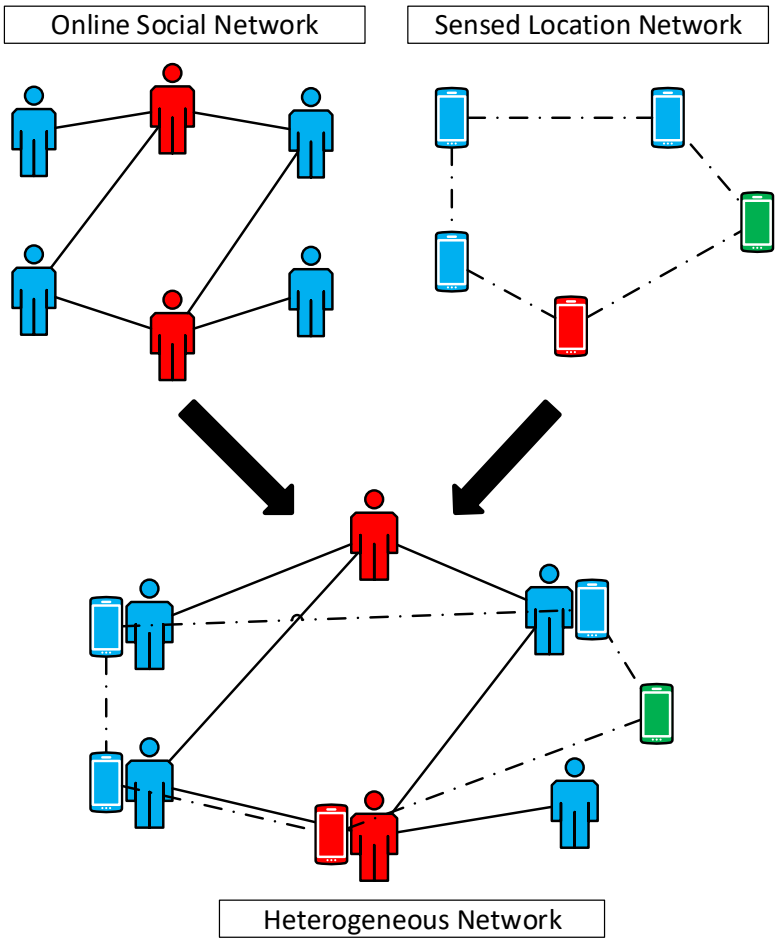


Figure 6.2. Illustration of Heterogeneous Network

pair u and v's relationship $pr(u, v)$ in the constructed heterogenous network.

Random walk is a very typical tool, which has been commonly used for graph analysis [115]. To incorporate random walk into network G_c and G_o, we develop a transition probability matrix to represent the transition probability on each link of both networks. To represent all possible transitions of random walk on H, the matrix size need to be $(|V_c \cup V_o|) \times (|V_c \cup V_o|)$. To satisfy the prerequisite that the sum of weights on outgoing links of each vertex equals to 1, we normalize the transition probability $Pr(H)$ of the heterogeneous network as following:

$$Pr(v, u) = \frac{pr(v, u)}{\sum_{w \in N(v) pr(v,w)}} \tag{6.2}$$

where u is a vertex connected to vertex v, and $N(v)$ denote all vertices connect to vertex v. Let $Pr(H)$ denote the normalized matrix which represents all the transition probability for the random walk. Particularly, consider the situation that many nodes existed in both network G_c and G_o, for each vertex appears, we create a link with transition probability 1 which let both two graph could be integrated as shown in Equation (3):

$$M_{(H)} = \begin{pmatrix} M_{(V)} & M_{(c)} \\ M_{(o)} & 0 \end{pmatrix} \tag{3}$$

where $M_{(V)}$ is matrix with size $|V| * |V|$. $M_{(V)}$ represents the transition probability in the heterogeneous network, $M(C)$ and $M(O)$ are the matrixes represent G_c and G_o. The random walk process is showing as follows:

$$Vec_N^{(t)} = (1 - \lambda) M_{(H)} Vec_N^{(t-1)} + \lambda M_{(H)}^{(0)} \tag{6.4}$$

where Vec_N is a vector denotes the edges' weight, and $Vec_N^{(t)}$ is the weight of each vertex at the t^{th} iteration. We could rewrite $Vec_N = [Vec_{(c)} Vec_{(o)}]^T$, according to Equation (4),

$$Vec_N^{(t)} = (1-\lambda)\begin{bmatrix} M_{(V)} \times Vec_{(V)}^{(t-1)} + M_{(c)} \times Vec_{(o)}^{(t-1)} \\ M_{(o)} \times Vec_{(V)}^{(t-1)} + 0 \end{bmatrix} + \lambda \begin{bmatrix} Vec_{(c)} \\ Vec_{(o)} \end{bmatrix} \quad (5)$$

We iteratively update Vec_N until the matrix become convergent, and the value on each link will be the relevance of all relationships in the heterogeneous network H.

6.2.3 Heterogeneous Construction

We apply random walk with restart process to combine both two sensed cyber physical world and online social world. As shown in the real data verification section later, this statistical approach is very stable and robust in our tested datasets. In the meanwhile, the method we proposed could also incorporate several heuristic strategies to optimize the heterogeneous network construction. For instance, if two users have a strong heterogeneous connection, the overall probability of those users having strong connection in cyber-physical world and online world could be high. And in another situation, two users who have large number of common friends in either online social network or sensed cyber physical network, the two users in the heterecious network would also tend to be connected because the sum of transition probability between these two nodes is higher. Another observation worth to mention is not all sensed cyber physical pattern could be directly applied without selection in practice. The main reason for the pattern selection is based on the real world practice. Some very high-frequency patterns we collected are actually common public locations such as airports, hospitals, or metrorail stations, *etc.* Meanwhile, the amount of very low-frequency patterns is also very large which should be deleted prior the random walk process in order to reduce the computational cost. For both two extreme situations, we use an entropy-based classical thresholds measure to filter the useless results [87]. We filter out the pattern with highest 10% high frequency and lowest 5% frequency according to the calculation result from [87] and [161].

Thus, we represent the random walk process in Algorithm 8.

Algorithm 8: Construction Heterogeneous Network H

input : $G_c(V_c, E_c)$, $G_o(V_o, E_o, W)$
output: $H(V, E, P)$

1 Generate FCL, FCR, FCS, and FCA according to our pattern models on $G_c(V_c, E_c)$;
2 Filter out biased patterns and shrink edges by apply result of entropy-based calculation ([87] and [161]);
3 According to 6.1 to generate $G_c(V_c, E_c, P_c)$;
4 Construct heterogeneous network $H(V, E, P)$ by with the set union of $G_c(V_c, E_c, P_c)$, $G_o(V_o, E_o)$;
5 Generate transition probability matrix and normalize each column following by Equation (2);
6 Iteratively update $Vec_N(t)$ until Equation (4) and (5) converges;
7 Return the heteromerous graph H with the edge weight based on the results of the last step;

6.2.4 Privacy Reserved Influence Maximization

Our ultimate objective is maximizing the influence in the heterogeneous network with privacy protection. Actually, through the integration of both sensed cyber physical world and online social world, the privacy of the user in the physical world is hidden behind the heterecious network since the new construct network will not tell whether the connection between two users are their common location, common activity, or their online friendship. However, the links within the heterogeneous network still represent the relationships among users. We believe identifying the most k influential users is not necessary to expose more sensitive information about users' relationships. Therefore, a differential privacy mechanism [111] is employed to further preserve the relationship privacy for influence maximization in the heterogeneous network.

The utility function of classical influence maximization is to find the best seed set $|S|$ with size k, which the set S could maximize the number of expected active nodes in the function $\delta_m S$. The influence prorogation process has an equivalent formulation as follows: (1) Flip a coin for each edge in H and remove it with probability $1-(P(e))$ until get resulting network H'. (2) Active the nodes in S, for any nodes in H' could reach from S [88]. The objective function $\delta_m S$ is the expected number of nodes in H'.

Let vertex utility of the network corresponding to an application parameter l be

$VU_{max}(G, \bar{G}, l)$. For any two users u and v, the weight between u and v is represented by $W[L_{uv}|\bar{G} \geq g(\epsilon)]$, where $g(\epsilon)$ denotes the prior probability of u and v being friends if they are both contained in a ϵ hop neighborhood, and $\epsilon = min\{h : W^h_{uv}(G) - VU_{max}(G, \bar{G}, h) > 0\}$. The utility function is monitoring the network variety between the original and perturbed networks. This result demonstrates a lower bound on link privacy. Therefore, by applying the above result, we update the utility function $\delta_m S$ while the same lower bound remains [111].

We incorporate a similar idea of IMM [136], which is the up-to-date influence maximization algorithm, adopting a sampling strategy to reduce computational costs. The main approach of IMM roughly consists of two phases: sampling and node selecting. To begin with, the first phase iteratively generates random reverse reachable set and puts them into a temporary storage set until reaching a specific condition; then the second phrase greedily selects the maximum coverage for each node to derive a k nodes set until the temporary set has been updated to the final result. We do not provide details of IMM due to space limitations anymore. Through the twofold privacy protection: first integrating the sensed cyber physical world and online social world by random walk, and hiding the sensitive user behavior patterns; then applying differential privacy technology to protect the connection privacy within the heteromerous network.

6.3 Experiment

In this section, we conduct real sensed cyber physical location network and online social network to evaluate the effectiveness and efficiencies of our proposed models. First, we show the proposed model accurately catching the features of sensed cyber physical world and online social network. We implemented all the codes with Python 2.7 with the support of the latest version of Snap.py [2], and all experiments are performed on a Desktop with operating system Windows 10. The PC is set up with 3.30GHz Intel(R) Core(TM) i3-2120 CPU and 12GB DDR3 memory.

[2]Python interface for Stanford Network Analysis Project(SNAP)

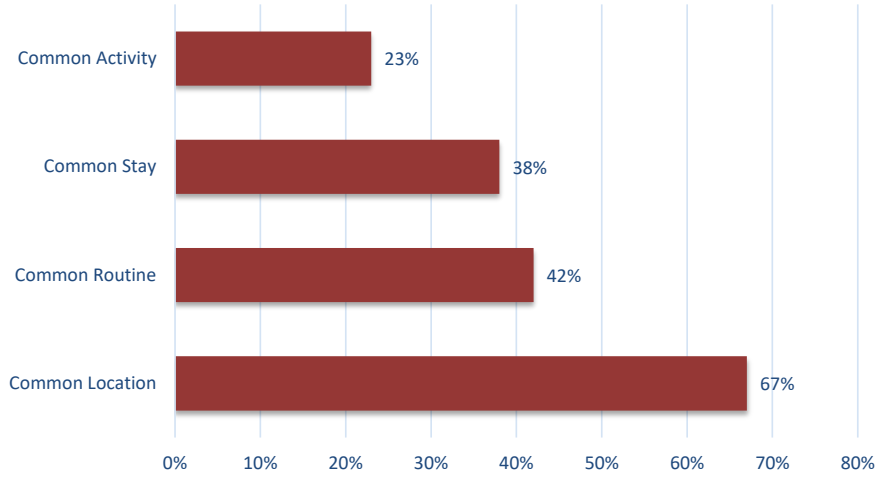


Figure 6.3. User's Behavior Patterns and Online Friendship

Two actual datasets named Brightkite and Gowalla from Stanford Network Analysis Project(SNAP) are investigated [38]. Brightkite and Gowalla are two popular LBS (location-based service) social network provider. The users' location information are sensed by the users' mobile device and the online social information are all available according to the users login with their Facebook accounts in the two datasets. Therefore, taking the advantage of our heterogeneous network construction, we could test the performance and effectiveness of our proposed framework. Consider the difficulty to detect users' movements that the distances of users movements are small while the scale of the physical world is large, we only employ part of the original Brightkite and Gowalla datasets to do the experiment. In the employed two subsets, the only conduct a densely distributed sub-area cyber physical data in a 400km×400km rectangle. We first adopt the very typical random point model to evaluate the users' position [86]. To simplify the calculation, we standardized all positions in the sensed cyber physical world and the time stamps for the logins to $[0, 1)^2$ and [0, 1), respectively. Table 6.1 summary the details of Brightkite and Gowalla.

As a whole, Brightkite only has 3/5 users compare to Gowalla, but the number of average login record in Brightkite is much high than Gowalla. In the following evaluation, we acquire several measurements to test the result from both two networks.

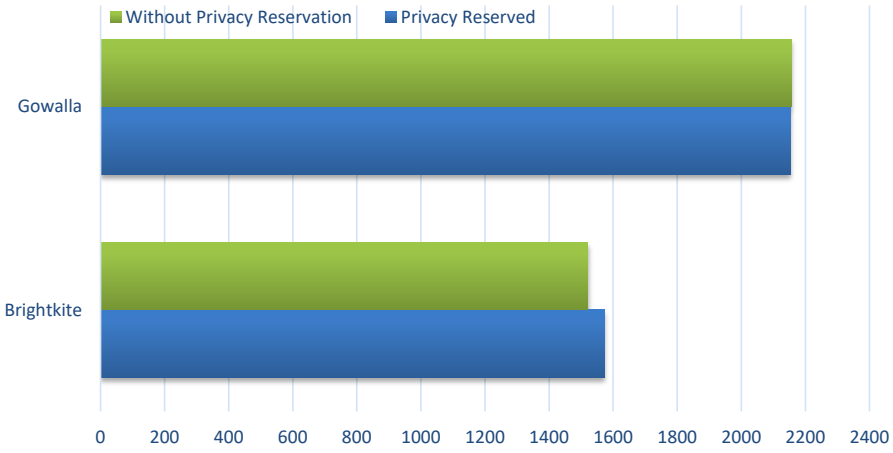


Figure 6.4. Privacy Preservation affect Influence Maximization

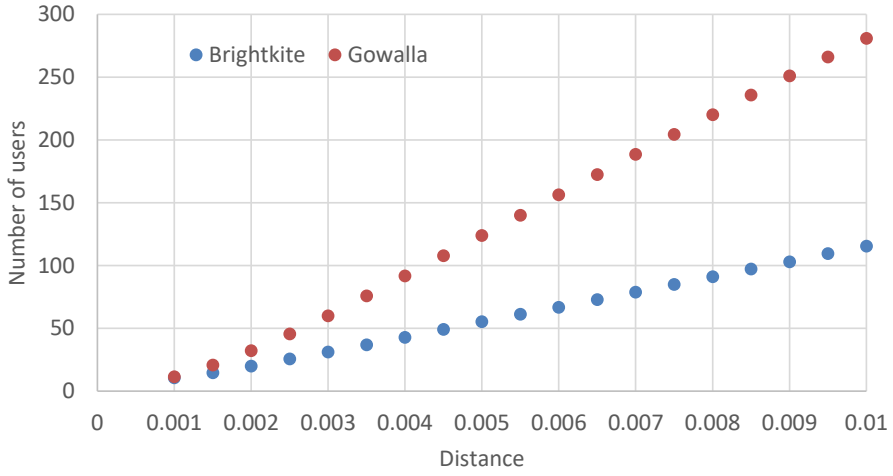


Figure 6.5. Number of Neighbors Distribution

Fig. 6.6 shows the distances distribution of login position records. The majority of the distances are very small for both Brightkite and Gowalla. For example, in Brightkite dataset, half of the distances are less than 0.004036, which is about 1.35km. In the meanwhile, for the Gowalla dataset, 25% of the distances are less than 1.92km. This observation indicates that the location information in the sensed cyber physical world is very stable. In other words, users location in the sensed physical world stays unchanged in most of time.

In Fig. 6.7, the second measurement as shown, demonstrates the distribution of friends numbers in the online social networks. The results show that there is a strong skewness in

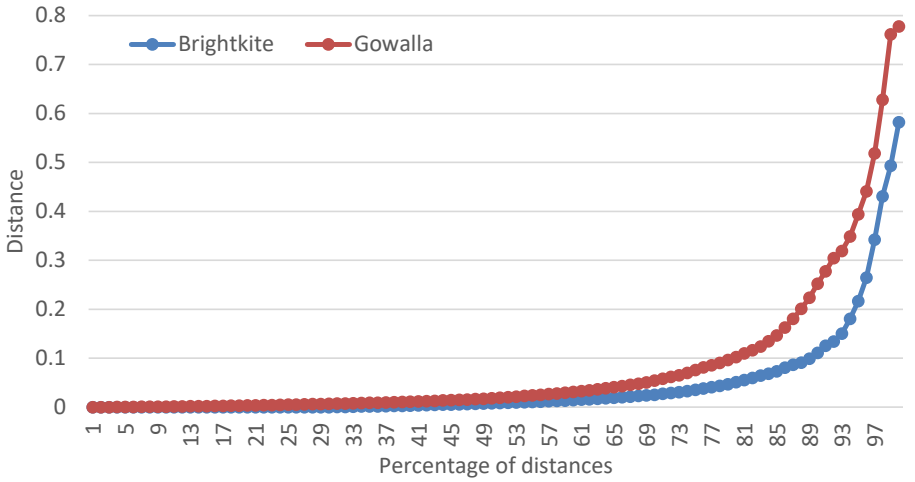


Figure 6.6. Distances Distribution

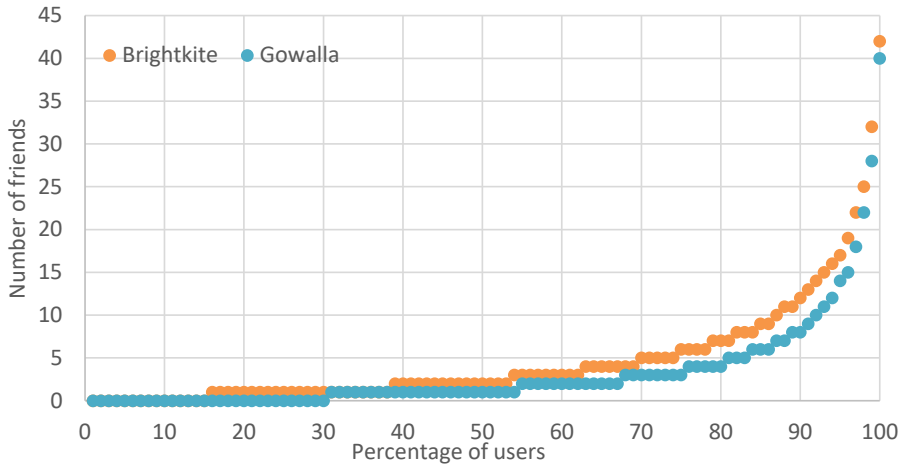


Figure 6.7. Number of Friend Distribution

this distribution for the two datasets. For instance, 80% users account for only about 25% of total friends in the online social network while the users with top 20% friends account for as high as about 75% of total friends in Brightkite. This result indicates that there are many variations existing in the online social networks for the influence prorogation of different users. This phenomenon also confirms that it is possible for a plenty of users get influenced in the end but only a few users get influenced initially. The combination of both sensed cyber physical world and online social network could take the advantage of both social features and cyber features into account.

Table 6.1. Network description

Network	Brightkite	Gowalla
User Number	3551	5231
Edge Number	9317	10134
Average Degree	5.248	3.875
Login times	121.278	56.797
Track period	2008.4 - 2010.10	2009.12 - 2010.10

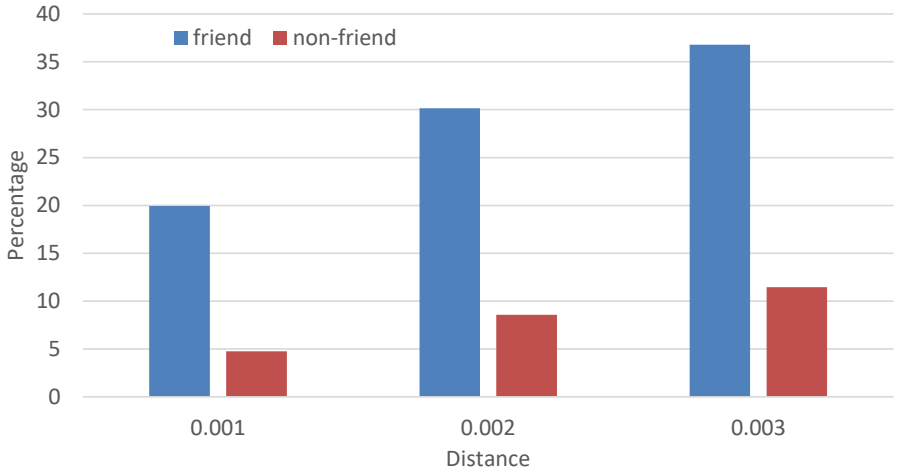


Figure 6.8. Percentage of Users in Brightkite

The results indicate that users location distribution may be very dense in the sensed cyber physical world. In Gowalla, for instance, more than 80 users are stay within the range of 0.005 (about 1.7 km) on average, which is a very small area. This result makes it much more convenient and practical to prorogate influence to real cyber world.

Considering the friends and non-friends relationship in Fig. 6.8 and Fig. 6.9, the result is the percentage of users within specific distances in both two network we evaluated. In this comparison, about 20% users keep a minimum distance less than 0.001 (about 0.4km) to their friends in Brightkite. However, for non-friends, this ratio is just 5% which is far less than the situation with friend relationship. In Gowalla, the two ratios of friends and non-friends percentage are 29% and 8%, respectively. This observation indicates that there is a very high interdependency of geographical positions for friends in the online social networks. In other words, if users are friends in the online social networks, their distances in the cyber-physical

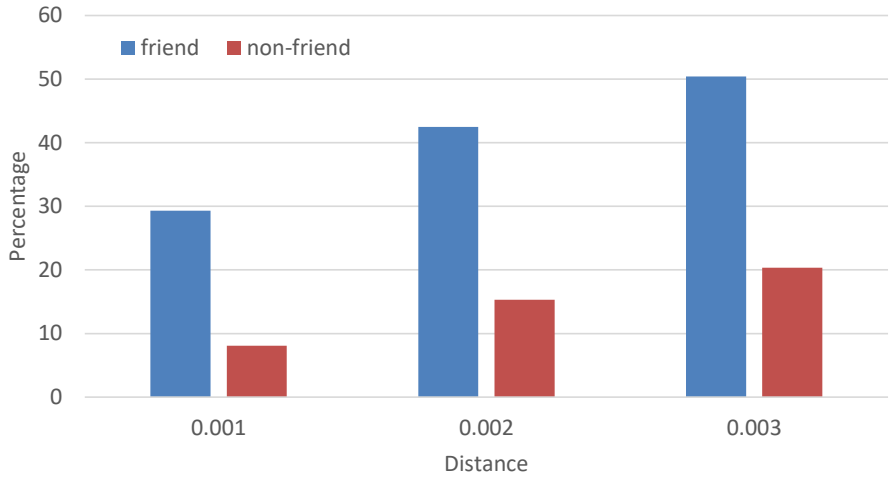


Figure 6.9. Percentage of Users in Gowalla

world may also tend to be very small. This result is also a verification of our assumption. Thus, the users' relationship in both sensed cyber physical world and online social network do have their inherent correlation.

Considering the number of neighbors in the sensed cyber physical world within different distances, Fig. 6.5 shows that in the cyber physical world, users who gathering together are still a very common situation. In the Gowalla, for instance, more than 80 users are within the range of 0.005 on average, which is about 1.7 km.

We consider the result of both friends and non-friends relationship in the two networks as shown in Fig.6.8 and 6.9. About 80% users keep the minimum distance more than 0.001 to their friends (about 0.4km). However, for non-friends, more than 95.2% users minimum distance is larger than 0.4km. There is a very high interdependency of geographical positions for friends in the online social networks. In other words, if users are friends in the online social networks, their distances in the physical world tend to be very small, vice versa. This phenomenon is also a verification of our assumption. Therefore, the relationships bewteen both cyber-physical world and online social network have their inherent connection.

Next, we test the result of our heterogeneous model by comparing both cyber-physical network and online social network. We first test the four user behavior patterns we proposed in sensed cyber physical network then compare the result of the online social friendship

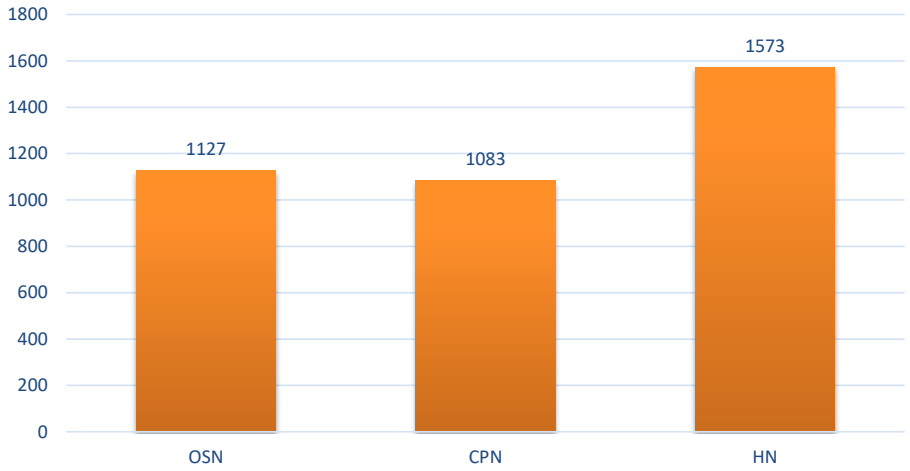


Figure 6.10. Number of Influence Nodes in Brightkite

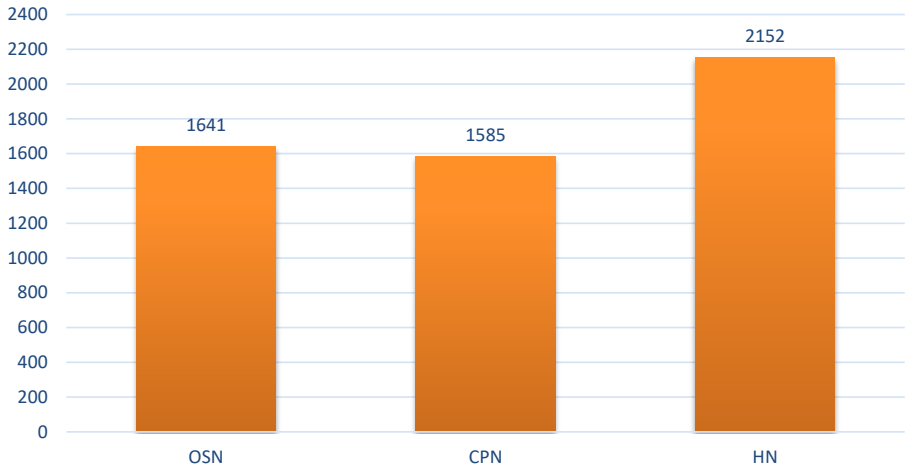


Figure 6.11. Number of Influence Nodes in Gowalla

connections in terms of percentage, separately.

As shown in Fig. 6.3, 67% of pattern 1 is overlapping with the online social friendship, which represents the consistency between sensed cyber physical world and online social world. Even for another 3 more complicated user behavior patterns, we proposed, the overlapping proportion between cyber physical world and online social network are still maintained a ratio at least 23%. As shown in Equation. 6.1, r_{CPF} is the combination of all 4 behavior patterns, the overlapping result of edges with r_{CPF} larger than 0 in cyber-physical and connections in online social world is more than 86%. Another obversion is for sensed cyber

physical network and online social networks, they have their own exclusive connections not includes in the other data. Therefore, applying the heterogeneous construction framework could take advantages of both the two kind of relationships for maximizing the influence in practice.

In both Brightkite and Gowalla, as Fig. 6.4 shown, we compare the number of expected influenced node with and without the additional privacy reserved mechanism. It is hard to tell a big difference between both algorithm, which states that add our privacy protection mechanism will not affect the result a lot, but as we analyzed in Section 6.2, the differential privacy techniques we employed could further protect the links privacy in the application.

At the end of this section, in Fig. 6.10 and 6.11, we demonstrate the result of the influence maximization with the privacy protection in cyber-physical network (CPN), online social network (OSN), and the heterogenous network (HN) we constructed. Obviously, the combination of both sensed cyber physical world and online world could significantly improve the performance of influence maximization in terms of the number of expected influenced nodes, and as we mentioned earlier, by employing the privacy protection mechanism, our approach could also protect the privacy issue cross both two world.

6.4 Summary

This chapter solves the influence maximization problem in a heterogeneous information network combing knowledge from both sensed cyber physical world and online social world. Four behavior patterns and corresponding formulated functions are proposed to model the users' behavior in sensed cyber physical world. We adopt the state-of-the-art influence maximization technique and differential privacy to achieve an efficient influence maximization with privacy protection. The real life data experiments verified that the framework works well for the problem of influence maximization and resolution is outperformed the up-to-date resolutions.

Chapter 7

INFLUENCE IN BIG SOCIAL DATA

7.1 Introduction

With the development of the internet, the increasing popularity of many online social network sites (Facebook, Google+ and Twitter, *etc.*) enable us to investigate large-scale social networks in a close view. However, we are facing challenges at all levels ranging from infrastructures to programming models for managing and mining large graphs.

Motivated by applications such as personalized recommendations, online advertising and microblog marketing, the study of influence diffusion and maximization attract more and more attentions. Domingos *et al.*[43] introduce the problem of identifying influential customs in the marketing campaign as a learning problem first. After that, Kempe *et al.* [88] studied the influence maximization problem and proposed two primary information diffusion models, namely as the independent cascade (IC) model and the linear threshold (IT) model.

In both of the models, the input is a network with nodes and weighted edges. Each node is either active or inactive. The possibility of one node becoming active increases monotonically with the number of its active neighbors. If one node becomes active, it will never be inactive again. This assumption is coming from the real life observation. If we only consider the influence diffusion process, let's look at the following example. One customer *Mary* just bought the latest iPhone 6S and posted one status on her Facebook page as "iPhone 6S is great, get one, you will never regret!". When her friend *Mike* on Facebook got this message and trusted her, then he could directly purchase one of his own. We naturally consider this process as *Mary* influencing *Mike*. As well as *Mike* is influenced (activated), his status will keep active (he has already purchased the device) and might continue to influence others through social media. Different users could have different levels of susceptibility, this characteristic was modeled as the probability of each edge between different users in

the network in our model. In the *IC* model, the beginning moment is denoted as time t_0, nodes with active status perform as "seeds" in the network. These nodes are considered contagious. A node u has one chance of influencing its inactive neighbor v with probability $p_{u,v}$, which represent the ability of the influence from u to v. If u succeeds in this attempt, node v becomes active at time t_1, otherwise, u will not try to influence node v anymore. This process will continues until no new node becomes active in the network. Similar to *IC*, there is also the same set of seed nodes in *LT* model. Whether a node v will be influenced depends on the sum of the weight of $\sum_1^{|N(v)|} p_{u_i,v}$, such that the sum of all the incoming weights to v is less or equal to 1. $\{u_1, u_2, \ldots, u_i\} \in N(v)$, where $\{u_1, u_2, \ldots, u_i\}$ are v's neighbors, and $N(v)$ is v's neighbor set. In each time stamp, node v selects a random threshold θ_v uniformly from $[0, 1]$. If the sum of weight from all the active neighbors of an inactive node v is more than θ_v, v becomes active at the next time stamp, otherwise, keep inactive. This process also repeats as well as *IC* to the end until no new node becomes active anymore.

Kempe *et al.* first formulated influence maximization as a discrete optimization problem in [88]. Considering a social network as a graph $G = (V, E, p)$, where V and E is the set of vertices and edges with size $|V|$ and $|E|$, and $p : E \to (0, 1]$ is the function assigning each edge $e \in E$ a probability $p(e)$. Choose an influence diffusion model (*IC* or *LT*) and an initial active seed set $S \subseteq V$, the expectation of the active node's number at the end of the process is the expected diffusion spread of S, denoted as $\delta_m S$. Then the *influence maximization* problem is defined as follows: A directed social graph $G = (V, E, p)$, find the best seed set S to maximize the $\delta_m S$.

However, in both *IC* and *IT* models, the evolution and influence are diffusing in unlimited time. The only termination condition is no new active node appears, but this assumption is not completely supported by the facts in the real social networks. Information always depends on timeliness, such as the advertisement of some products is limited just in a period of time, and the news is only meaningful during a particular period, influence diffusion in general is stopped before the original stop condition in *IC* and *LT*. Based on observation above, we give a time constrain τ to strengthen the classical models and the proposed model

"**T**ime Constraint ***IC*** Model(***TIC***)" and "**T**ime Constraint ***LT*** Model(***TLT***)", which restrict the output and let the algorithm aim at maximizing the influence within the threshold τ time.

On the other hand, as shown in Chen [32], the influence diffusion processing itself in IC and LT are "#P-Hard" problems, and the model IC and LT in their original paper are both proved to be "NP-Hard" problems. These facts told us that developing an exact algorithm to solve this problem is impossible if $NP \neq P$. And the simulation processing of the model itself is also very time consuming. Even though several heuristic algorithms have been developed recently, but it is hard to give a theoretical guarantee in heuristic algorithms. As a result, the existing works cannot handle the real large scale network efficient enough as we illustrated.

Additionally, "Big Data" is hitting at a future in which we could compute on a relatively transparent environment (such as a cluster) but not just single local machine has become another buzzword after Web 2.0. As probably the most notable big data frameworks *Hadoop* and *Spark* provide us a potential solution for large scale networks to do the influence maximization. Hadoop is an Apache project provided a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce [41] paradigm. Hadoop is available via the Apache open source license, which provided us an opportunity to develop a big data environment based on Hadoop for our influence maximization problem. As well as *Hadoop*, *Spark* is another open source Apache project. Different from the traditional map reduce, *Spark* approaches data processing mainly in memory instead of a hard drive space.

As shown in Fig. 7.1, the input of our problem is a social network with a huge number of nodes and edges between nodes. Each edge has a probability (weight) representing the influence between the nodes pair, by processing the influence maximization on both *Hadoop* and *Spark* environments, the output is a seed set S with size k, which can maximizing influence followed by our influence model.

In this chapter, we have the contribution below:

Figure 7.1. Influence maximization processing in cloud environment

- First, we introduce two new influence maximization models with time-constraint properties. We give the formal definition of the new models and the analysis of problem complexity.

- Followed by the analysis of problem complexity, we propose the theoretical proof result of the property monotony and submodular which give us the possibility of using an efficient greedy algorithm. We will also give the theoretical analysis of the approximation ratio of our algorithm.

- Thirdly, based on the new model and efficient algorithm we proposed, Hadoop-based cloud computing environment is used to deploy our experiment data set and algorithm. Consider the specific problem we also suggest new strategies to optimize the Hadoop-based algorithm.

- Last but not the least, by using both large scale simulation data and real social networks data, we implement the algorithm in a Hadoop-based cloud environment and evaluate the large scale data by several efficient distributed strategies.

The rest of this chapter is organized as follows. Section 7.2 presents the preliminaries and problem definition. Section 7.3 illustrates the algorithm and theoretical analysis. Evaluation

results based on real and synthetic data sets are shown in Section 7.4. Section 7.5 concludes our chapter.

7.2 Data Model and Problem Definition

In this section, the formal definition of the problem we solved and the corresponding analysis will be proposed.

Consider a time threshold $\tau \in \mathbb{Z}_+$, define $\delta_\tau : 2^V \to \mathbb{R}_+$ to be the set function such that $\delta_\tau(S)$ with $S \subseteq V$ is the expected number of the activated nodes number by the end of the time constraint τ under our model.

The time constraint influence maximization is the problem of finding the seed set S with at most $k = |S|$ seeds such that the expected number of activated nodes by time τ is maximized. Formally, $\delta(S^*) \geq \delta(S)$ for any set S of at most k nodes, find

$$S^* = argmax_{S \subseteq V, |S| \leq k} \delta_\tau(S) \tag{7.0}$$

More specifically, consider the evolution when influence is spreading in the IC and LT models:

Let decay factor $\lambda \geq 1$ in the influence evolution function. A large λ means a slow-decay effect. Then the decay evolution is the function $g(\lambda)$ equals to $(\frac{1}{2})^{\frac{t-t'}{\lambda}}$. When the value of this function below the minimum threshold $\dot{p}$, we stop to calculate the probability of that edge. In practice, this decay function could also be a linear, logarithm even exponential function which simulate the decay of relationship in the social network.

7.2.1 Time Constraint *IC* Model (*TIC*)

Based on the classical *IC* Model introduced in the Section 6.1, each node v will be influence by all $\{u_1, u_2, \ldots, u_i\}$ from v's neighbor set $N(v)$ according to the weight of $p_{u_i,v}$ on each neighbor u_i, such that the weight of any the incoming weights from u_i to v is less or equals to 1. In each iteration, the node u_i selects a random threshold $\theta(u_i)$ uniformly from

$[0, 1]$. If the weight $p_{u_i,v}$ between u_i to v is more than the threshold $\theta(u_i)$, then v becomes active at the next time stamp. Otherwise, u_i will not try to influence v anymore. Maximize the influence function $\delta(S^*)$ which is the expected number of influenced nodes at the end of the propagation is the objective. Different from the basic IC model, the probability $p_{u_i,v}$ on each edge in our model will decay followed by the decay function as the influence path from the original seed set S. And the process will terminate by the threshold τ.

7.2.2 Time Constraint *LT* Model (*TLT*)

Based on LT Model, each node v will be influenced by all $\{u_1, u_2, \ldots, u_i\}$ from v's neighbor set $N(v)$ according to the sum of the weight of $\sum_1^{|N(v)|} p_{u_i,v}$, such that the sum of all the incoming weights to v is less or equals to 1. The node v chooses a random threshold θ_v uniformly from $[0, 1]$ at each time stamp. If the sum of weight from all the active neighbors of an inactive node v is more than the threshold, then v becomes active at the next time stamp. Maximize the influence function. Similarly as the TIC model, the process terminates until one of the basic condition or the threshold τ satisfied.

7.3 Algorithm and Theoretical Result

According to the previous related work of Kempe [88], the following algorithm can solve the problem in a approximation ration as $(1 - 1/e)$.

Algorithm 9: Greedy Algorithm (GA)

input : T, k, δ_m
output: seed set S

$S \leftarrow \emptyset$;
while $|S| < k$ **do**
  $u \leftarrow argmax_{\omega \in V \setminus S}(\delta_m(S \cup \{\omega\}) - \delta_m(S))$;
  $S \leftarrow S \cup \{u\}$;

Different from IC and LT, we add the time constraint and the probability decay function to make the problem more suitable for social networks in practice.

Next, we will give the theoretical analysis of the problem property and the hardness of our problem. According to the approximate algorithm theory, if one algorithm problem can satisfy monotonicity and sublobularly, then we can directly get an efficient approximate algorithm with the approximate ratio as $(1-1/e)$. We first provide the proof of our problem of the property monotonicity and sublobularly.

Theorem 5. *The influence maximization model TIC a monotonic and submodular model.*

Proof: For the monotonicity, since the influence function of *TIC* is an increasing function, the conclusion is obvious.

For the sublobularly, let X denote one sample point in this space $\delta_X(A)$ is the total number of nodes activated by the process when A is the initially set $R(v, X)$ denote the set of all nodes that can be reached from v on a path consisting entirely of live edges. $\delta_X(A)$ is equal to the union $\cup_{v\in A} R(v, X)$.

(1) $\delta_X(S \cup \{v\}) - \delta_X(S)$ is the number of elements in $R(v, X)$ that are not already in the $\cup_{v\in S} R(v, X)$.

(2) $\delta_X(T \cup \{v\}) - \delta_X(T)$ is the number of elements in $R(v, X)$ that are not already in the $\cup_{v\in S} R(v, X)$.

$S \subseteq T \Rightarrow (1) > (2)$; $\delta(A) = \sum_{outcomes X} Prob[X]\delta_X(A)$. Since the basic *IC* follows the process as we mentioned, limited by the time threshold, the process will be terminated earlier, but it will get a similar result since it can be considered as a specific case of the basic IC model, end.

Theorem 6. *The influence maximization model TLT is monotone and submodularity.*

Proof: For the monotonicity, since the influence function of *TLT* is an increasing function, conclusion is obvious.

For the submodularity, let v have an influence weight $b_{v,w} \geq 0$ and $\sum_w b_{v,w} \leq 1$. Suppose v picks at most one of its incoming edges at random, selecting the edge from w with $b_{v,w}$ and no edge with $1 - \sum_w b_{v,w}$. Similarly as in the Theorem 5, Selected as live and Unselected as blocked. We prove the two distributions are the same:

(1)The distribution over active sets obtained by running the Linear Threshold process to completion starting from A.

(2)The distribution over sets reachable from A via live-edge paths, under the random selection of live edges defined above.

For directed and acyclic graph: Under the *TLT* Subset S_i of v_is neighbors is active, the probability is $\sum_{v \in S_i} b_{v_i,w}$. For graph G is not acyclic, A_t is the set of active nodes at the end of iteration t, A_0 is the initially set The probability node v become active in iteration t+1 is equal to the chance that the influence weights in $A_t \setminus A_{t-1}$ push it over its threshold $\frac{\sum_{v \in A_t \setminus A_{t-1}} b_{v,u}}{1-\sum_{u \in A_{t-1}} b_{v,u}}$.

If graph G is not acyclic: Start with set A, for each node v with at least one edge of the set A, determine whether vs live edge comes from A. If yes, v is reachable, if no then keep the source of vs live edge unknown. Expose a new set of reachable nodes $A1$ in the first stage, then produce sets $A2$, $A3$,.... Similarly to the proof of *TIC*, we can easily get that the *TIT* is a specific case of the *LT* but with the time constraint and the decay process, end.

According to the Theory 6 we have shown, a simple greedy algorithm can be used to efficiently solve the new problem of *TIC*, and *TLT*.

Algorithm 10: Time Constraint Greedy Algorithm (TGA)

input : τ, k, g, δ_m
output: seed set S

```
S ← ∅;
while |S| < k do
    u ← argmax_{ω∈V\S}(δ_m(S ∪ {ω}) − δ_m(S));
    S ← S ∪ {u};
    Recalculate the probability on of new edge by function g;
    if the time exceed the threshold τ then
        Break;
```

We can easily get that our new algorithm has the same time complexity as *IC* and *LT*. Since we consider the feature of time constraint and probability decay, our models simulating the real phenomenon more accurately and the algorithms based on our models are also much

more practical.

Besides, we also provide an improved version degree discount algorithm [31] to solve our influence maximization models. Different from traditional degree discount, our algorithm as shown in Algorithm 11 considers the time constrain and the influence decay process, together with several optimizations in implementation.

Algorithm 11: Time Constraint Degree Discount Algorithm (TDDA)

input : τ, k, g, δ_m
output: seed set S

```
S ← ∅;
for each vertex v do
    calculate v's expected degree d_v;
    dd_v = d_v;
    t_v = 0;
    Recalculate the probability on of new edge by function g;
    if the time exceed the threshold τ then
        Break;
```

7.4 Experimental Study

We use both real world networks and synthetic networks to demonstrate the effectiveness and the efficiency of our model and algorithm. We also evaluated our algorithms in both quality and efficiency. All experiments were performed on a cluster with Hadoop and Spark environment.

7.4.1 Environment Setup

1. Single Node Setup

   We implemented the algorithms in Python 2.7.2 with the latest version of Snap.py[1]. All experiments are performed on a PC running Windows 10 with Intel(R) Core(TM) i3-2120 CPU 3.30GHz and 12GB memory.

[1]Python interface for SNAP

Table 7.1. Cluster description

Node Name	CPU	Memory	Function
gpu10	Dual Intel Xeon E5-2650	64 GB DDR3 (1866MHz)	Master & SecNameNode
gpu05	Dual Intel Xeon E5-2650	64 GB DDR3 (1866MHz)	Worker & DataNode
gpu06	Dual Intel Xeon E5-2650	64 GB DDR3 (1866MHz)	Worker & DataNode
gpu07	Dual Intel Xeon E5-2650	64 GB DDR3 (1866MHz)	Worker & DataNode
gpu08	Dual Intel Xeon E5-2650	64 GB DDR3 (1866MHz)	Worker & DataNode
gpu09	Dual Intel Xeon E5-2650	64 GB DDR3 (1866MHz)	Worker & DataNode
gpu11	Dual Intel Xeon E5-2650	64 GB DDR3 (1866MHz)	Worker & DataNode

Table 7.2. Details of synthetic data

Network model	Nodes	Edges
Syn-SmallWord(S)	1,000	7,356
Syn-SmallWord(L)	50,000	638,274
Syn-Kronecker(S)	1,000	19,215
Syn-Kronecker(L)	50,000	693,473

2. Cluster Setup

   We set up both Hadoop and Spark environment on the Apache Hadoop NextGen MapReduce (YARN). We installed the latest version Hadoop 2.7.0 which release on 21 April 2015. We use 7 of the cluster nodes finished the Map-Reduce process. Apache Spark's GraphX version 1.5.2 in Python and Apache Hadoop's Giraph version 2.0 in Java are implemented to deploy in the cluster. (Detail of nodes in cluster could be found in Table 7.1)

7.4.2 Synthetic Social Networks

Small-world graphs: the small-world network model is a very classical model following the small-world features according to "small-world" [151]. This model is referred to as *Syn-SmallWord*.

Kronecker graphs: this generative model proposed in [96] generates a network in a natural way. The networks grow from 5 initial nodes and then Kronecker's idea is repeatedly applied to expand the network. This model is referred to as *Syn-Kronecker*.

Table 7.3. Dataset in experiment

Data	Nodes	Edges	Nodes in LWCC	Nodes in LSCC
Amazon0302(A1)	262111	1234877	262111 (1.000)	241761 (0.922)
Amazon0312(A2)	400727	3200440	400727 (1.000)	380167 (0.949)
Amazon0505(A3)	410236	3356824	410236 (1.000)	390304 (0.951)
Amazon0601(A4)	403394	3387388	403364 (1.000)	395234 (0.980)

Based on the initial networks generated from the above models, we dynamically change each network based on the idea proposed in [9]. Since we have multiple synthetic networks in the experiments, the average summary of 10 networks' statistic features has been used instead. As shown in Table 7.2, we generate two scales (small (S) and large (L)) of networks for both models with time stamp length of 50 and 100, respectively.

7.4.3 Real Social Networks Data Experiment

Different kinds of real data sets are used in our experiment, the first group of data sets shown in Table 7.3 come from SNAP(Stanford Large Network Dataset Collection)[2] which is an open network data set collection builded by Stanford university for researchers doing their research. The network statistic were evaluated by number of nodes(edges), number of nodes(edges) in largest WCC(weakly connected component) and SCC(strongly connected component), and Diameter(longest shortest path). Table 7.3 is based on the *Customers Who Bought This Item Also Bought* feature of the Amazon website. Four networks come from March to May in 2003. Each network contains a directed edge from i to j if a product i is frequently co-purchased with product j [95]. In the co-purchased network, it is undirected graph, we generate the two direction on each edge.

Besides the Amazon product co-purchasing networks in Table 7.3, we also evaluate our algorithm in the real datasets below:

WikiVote is a data set obtained from the [94] which collected the vote history data of Wikipedia [3]. The network includes 7115 vertices and 103689 edges which contains the voting

[2] `http://snap.stanford.edu/data/`

[3] http://www.wikipedia.org/

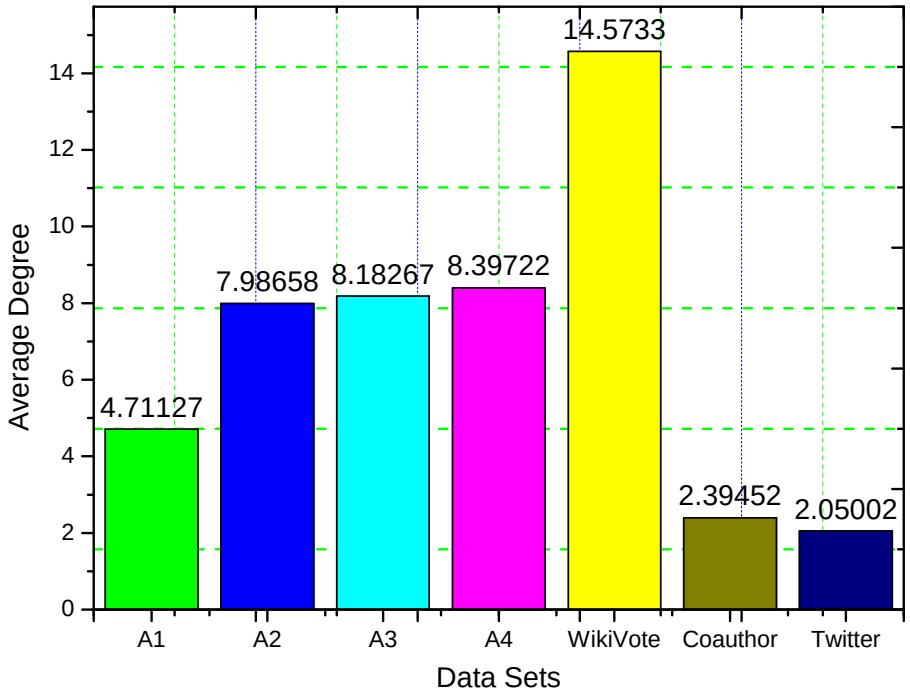


Figure 7.2. Average degree of real social networks

data of Wikipedia from the inception till January 2008. If user i voted on user j for the administrator election, there will be a directed edge from i to j.

Coauthor is a data set obtained from [134], which collected the authors' network by ArnetMiner [4]. We use the subset which include 53442 vertices and 127968 edges. Since the coauthor relationship is symmetrical, when the author i has a relationship with author j, there will be one direct edge from i to j and another edge from j to i .

Twitter is a data set obtained from [106, 79] which collected the information from Twitter [5]. We use the subset includes 92180 vertices and 188971 edges which represents the follower relationship.

As same as the experiment in [141], we set the positive probability as $1/deg(v)$ for an edge (u, v), where $deg(v)$ is the in-degree of v. And we let the negative probability on each edge be 10,30,50 and 80 percent of the positive probability.

As shown in Fig.7.2, we give the average degree of each real network, the WikiVote has

[4]http://arnetminer.org, an academic search system.
[5]https://twitter.com/

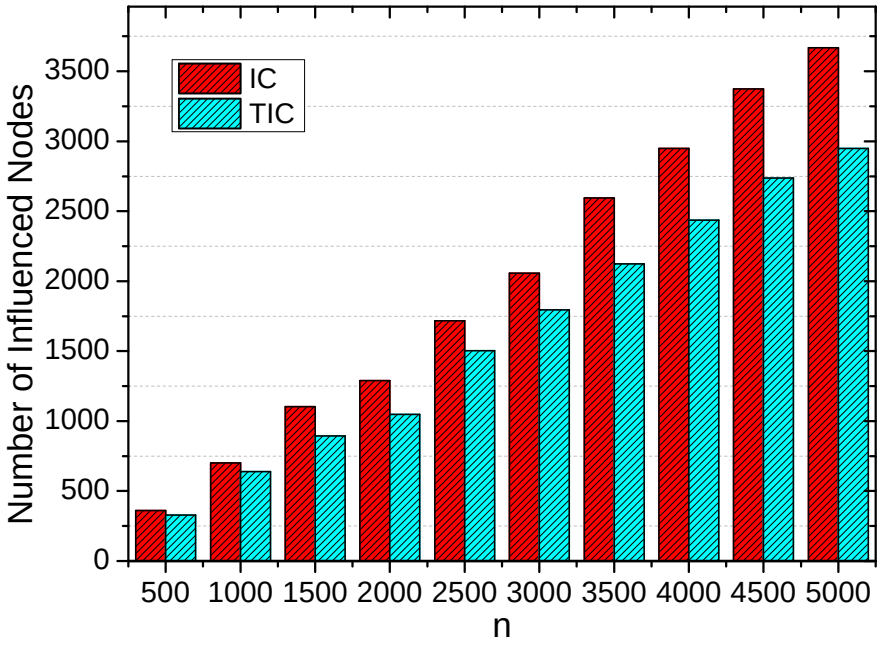


Figure 7.3. Comparison between *IC* and *TIC* with network size increase.

the highest average degree, which means the WikiVote potentially is the densest network among others.

7.4.4 Simulation Data Experiment

We present the experiment result of the Synthetic data on a single node first. By default, we test the two greedy algorithm GA and TGA corresponding to model *IC* and *TIC*.

As show in Fig.7.3, we test the change by the size of network under Kronecker model (the Small World model presents a similar result), when the network size increase, the different between our *TIC* model and *IC* model became larger. This trend is resulting from *TIC* model consider the time constraint and influence decay during the propagation. Also, when the network size was increased, the propagation scope increased, but limited by the time and influence damping, the number of influenced nodes was also decreased.

As shown in Fig.7.4 and Fig.7.5, we can have that the algorithm of *IC* can influence more nodes than our new model *TIC* because our new model considers the time constraint and the probability decay process. Even though, since the *IC* and *LT* are the ideal assumed models,

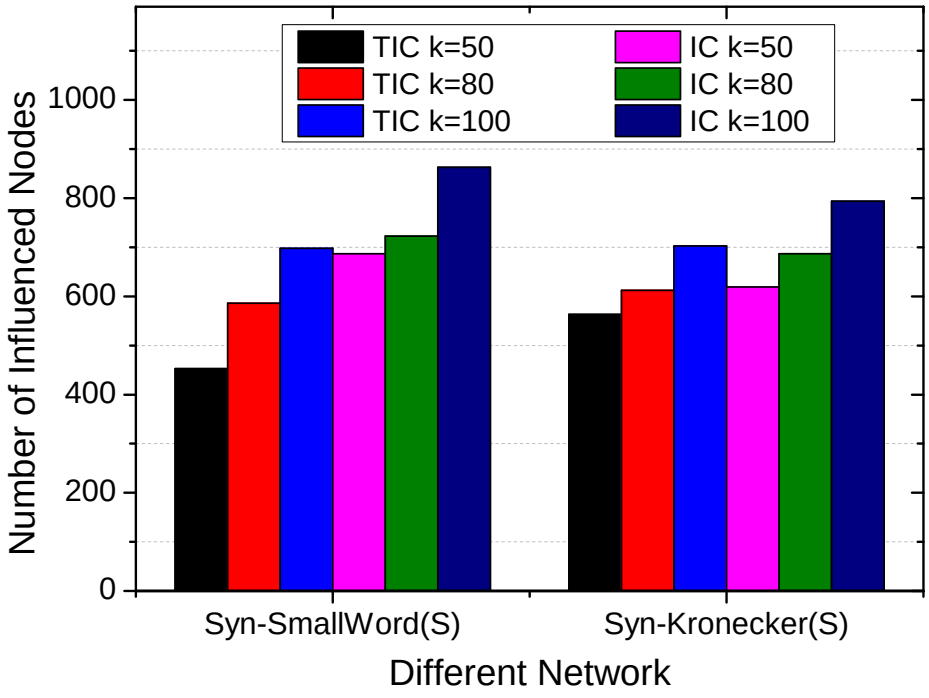


Figure 7.4. *IC* and *TIC* on small size synthetic data

our models do describe the influence process in a more practical way than the classical ones.

Since the result of *IC* and *LT* present a very similar result, limited by space, we mainly just show *IC*'s result.

The first advantage of our model is that it captures the real life attributes of influence and models them. The second advantage is our model could end the influence process limited by time. The time constraint and influence decay could end the iteration in advance, which is also speeding up the algorithm. As shown in Fig.7.6 and Fig.7.7, the other model, *TIC*, could apparently save more time to find out the most influential nodes in the two scales networks.

All experiments above show the features and advantages of our model. But if one considers the running time for either networks or either model, although our model *TIC* could outperform the classical *IC* with the greedy algorithm, the running time is hard to be satisfied and if the size of data increases further, it is very hard to finish in a reasonable time.

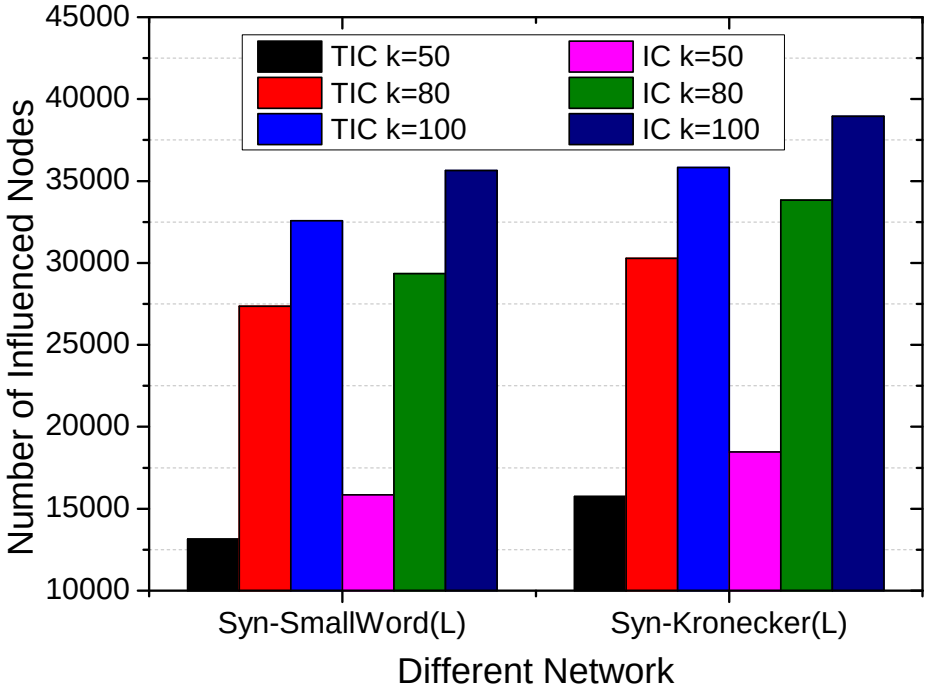


Figure 7.5. *IC* and *TIC* on large size synthetic data

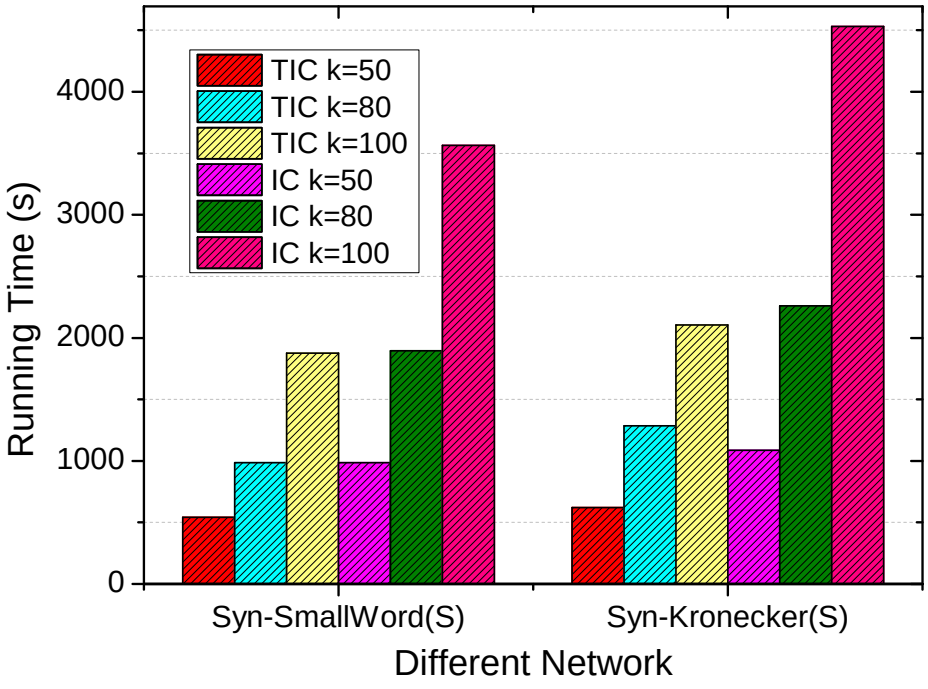


Figure 7.6. Running time of *IC* and *TIC* on small size synthetic data

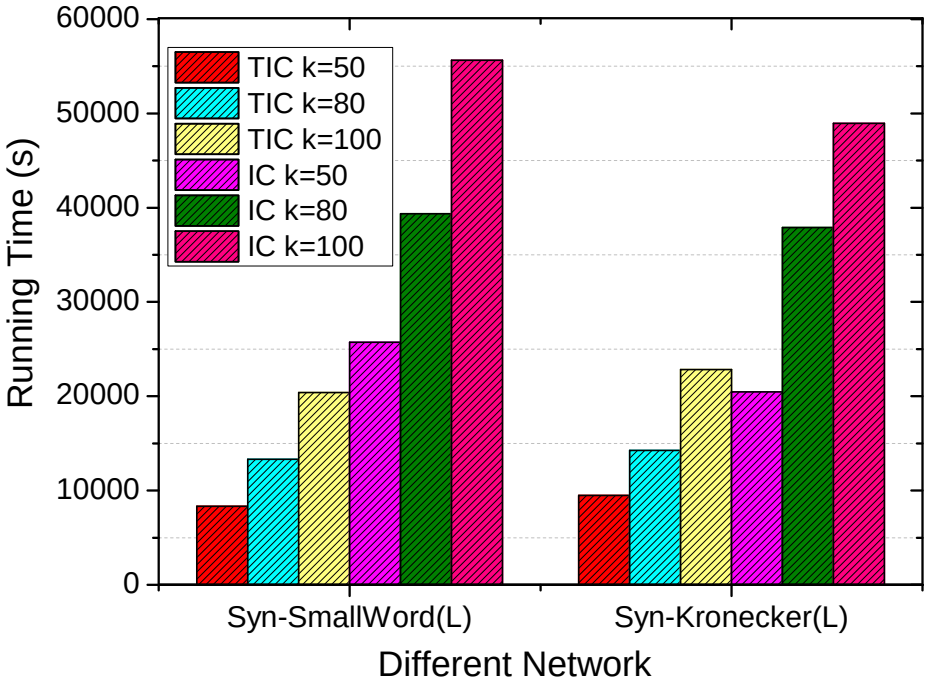


Figure 7.7. Running time of *IC* and *TIC* on large size synthetic data

Next, we implement model *IC* and *TIC* in the cluster (Hadoop) under both Small-world and Kronecker with greedy algorithms. As shown in Fig. 7.8 and Fig. 7.9, apparently that cluster could achieve much better performance. Later in this section, we will show more result and comparisons between Hadoop and Spark.

7.4.5 Real World Data Experiment

Fig. 7.10 and Fig. 7.11 show that the two real social network also follow the analysis of our theory. But the different networks data have different distribution and topology, in the Fig. 7.11 different network show a different changing trends.

To compare the performance of different Big Data frameworks, we also implemented the algorithm GA, TGA, and TDDA with our largest real world network Amazon(A3) with 410,236 nodes and 3356,824 edges on both Hadoop and Spark, we set the size of seed set k as 50. As shown in Fig. 7.12, the performance on Spark is much better than Hadoop. This result is based on the different mechanism of Hadoop and Spark. Spark basically is a memory

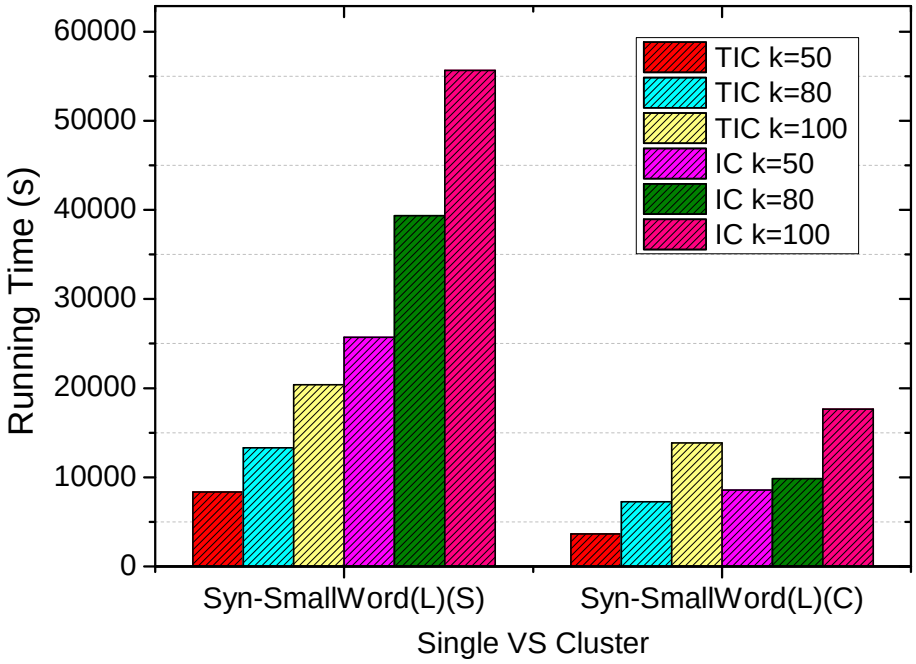


Figure 7.8. Running time of single machine and cluster under Small-world

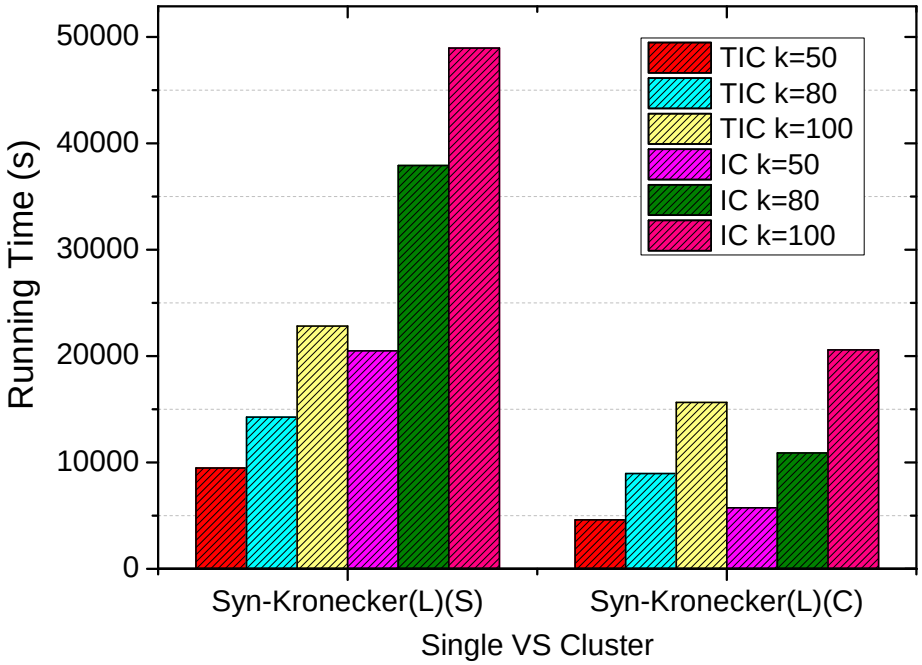


Figure 7.9. Running time of single machine and cluster under Kronecker

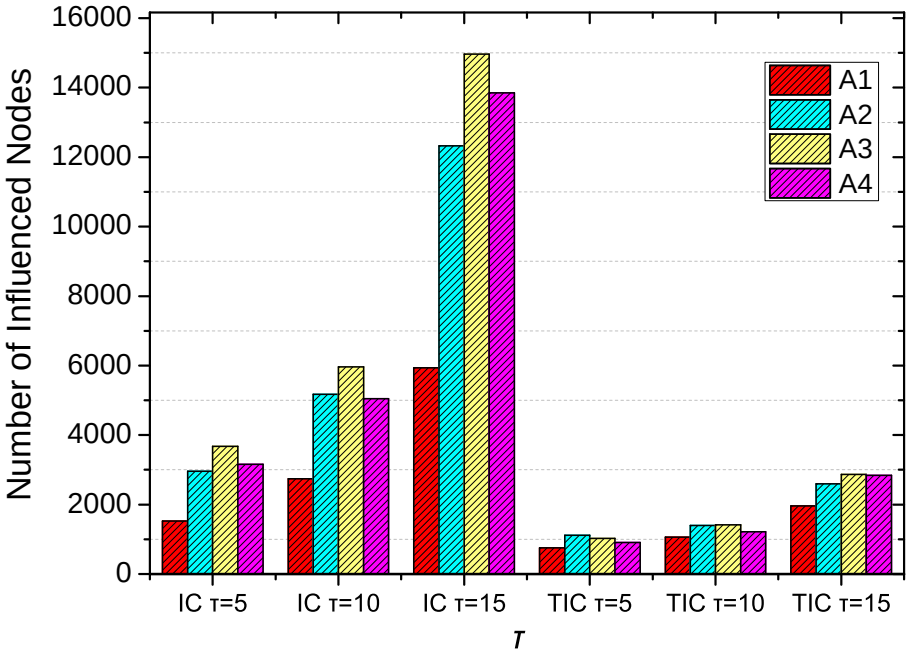


Figure 7.10. *IC* and *TIC* in Amazon co-purchase data.

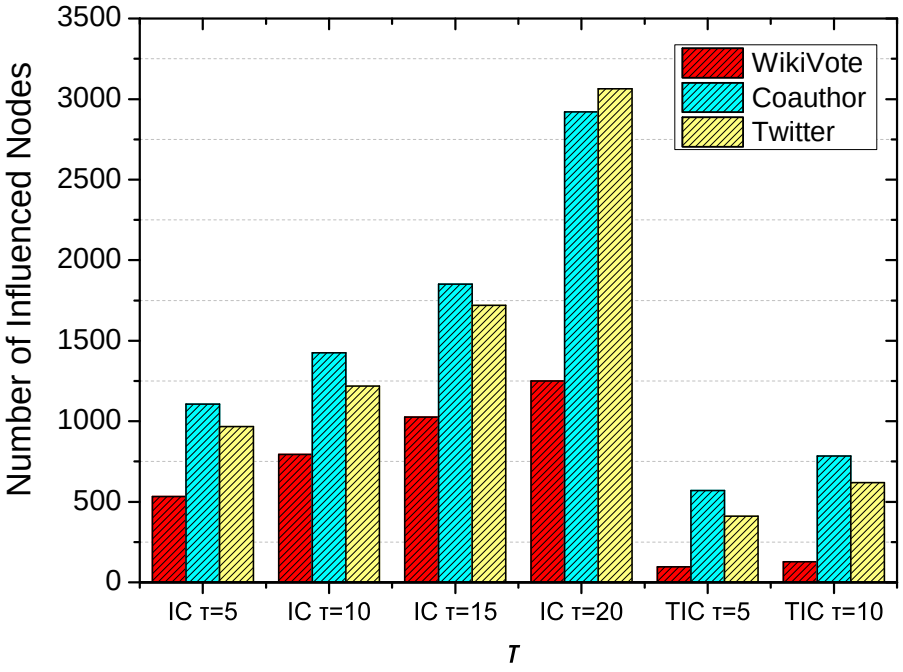


Figure 7.11. *IC* and *TIC* in other real social networks.

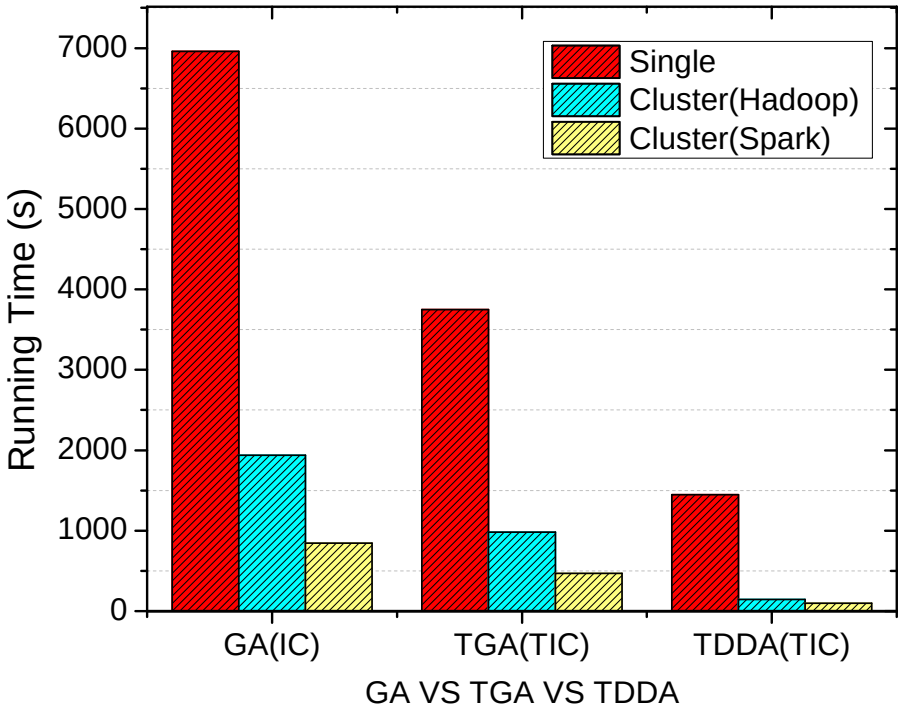


Figure 7.12. GA(IC) VS TGA(TIC) VS TDDA(TIC).

based framework for massive computation. Even though, as shown in the result, Hadoop is still much better than our single machine's performance. The reason our algorithm TDDA in Fig. 7.12 performs better than the other two greedy algorithms is because TDDA is a heuristic algorithm which has a much lower compute complexity.

Overall, the experiment shows that our model and algorithm match the theoretical result we proposed. And the performance results we tested on different big data platform on both synthetic data and real world data show that our algorithm and implementation could outperform most existing alternative approaches.

7.5 Summary

This chapter presents two new models TIC and TLT which extend the practicality of the classical IC and LT models in the influence maximization problem. The theoretical analysis shows that the two new models we propose both follow the monotonicity and submodularity which could help us to design simple greedy algorithm with a guaranteed

approximate ratio $(1 - 1/e)$. Both the simulation and real social network data implementation on a Hadoop-based environment show that our new algorithm can solve the problem efficiently and effectively.

Chapter 8

FUTURE RESEARCH DIRECTIONS

In this chapter, we are going to show more future problems and challenges for influence analysis, and pointing out the future research directions in the following.

8.1 Competitive Influence Analysis

All topics we discussed are working on single information resource, but there are many different situation in real life that more than one information resources are existed.

Bilateral competition diffusion model could be considered as there are two opposite opinions in the social scenario, where one is positive, and another is negative. How to analyze the information diffusion is a very challenging and meaningful research topic. In real life, it is quite common in the situation that different ideas are competing for their influence in the social networks. Such competing diffusion could range from two competing companies, friend and foe relations, two political candidates of the opposing parties to even the government tries to inject truth information to fight with rumors spread to the public. Besides the bilateral competition models, also from a competitive aspect, the competitors could be more than two. For example, in BMW, Ford, Honda, Toyota, and Tesla are all famous car brand, how to model the influence of multiple competitors are very challenge. These scenarios are arising in the real world. Many companies with comparable products, more than two political parties run for the election, and various rumors for some hot topics, *etc.* How to model many different competitors with or without confliction in a social network to propagate the influence are still very challenging problem. Map coloring and game theory might be very potential resolutions for multiple competitors influence problems. But there is still no practical resolution available, how to solve these kinds of problem might be one of the important further research direction in influence analysis.

8.2 Influence Analysis with Domain Knowledge

Domain knowledge could be used to refer to an area of human endeavour, an computer activity, or other specialized discipline. Incorporating many kinds of domain knowledge could greatly enhance the ability of influence analysis techniques.

As well as the development of modern mobile devices, the connection between cyber-physical network and online social network is significant strengthened. Integrate cyber-physical knowledge to influence analysis could improve both the accuracy and practicability. But the two kinds data of cyber-physical and online social network are very different from each other. How to combine the cyber-physical information and online information together to construct a novel framework for influence analysis is still an open problem. Besides the intuition problem of analysis, when we apply the cyber-physical knowledge to our problem, an imperative issue is cyber-physical world are carry a lot of privacy information of social participators. How to analysis the influence with consideration of privacy is also a very challenge problem. For example, location information has been studied in cyber-physical aspect for a long time. Location information could significantly improve the quality of our influence analysis since if one event happens in a particular location, it could direct influence all user around that area, and this kind of influence is more specific and observable. However, the location information is also very sensitive to users and the research that how to analysis the influence with privacy preserving is still blank.

Besides the domain knowledge in cyber-physical world, the marketing knowledge could be applied to many business applications. One of most important application of influence maximization is business marketing. In the domain of marketing, influence are using to promote new product, deliver promotion, and spread marketing campaign. In political life, political views, dissent, and attitude also need to spread and expand. Influence maximization could be one of very powerful tool for political parties. In health domain, how to spread the health lifestyle and reduce the un-health habit are also very related to influence analysis.

8.3 Influence Analysis in Massive Scale Data

As a number of available data increases, kinds of massive scale data are available which offers us more and more new issues. As probably the most notable big data platforms *Hadoop* and *Spark* provide us a potential solution for large scale networks to do the influence analysis. Hadoop is an Apache project provided a distributed file system and a framework for the analysis and transformation of very large data sets using the MapReduce paradigm. Hadoop is available via the Apache open source license, which provides us an opportunity to develop a big data environment for our influence analysis challenge. Spark is a very fast and general engine for big data processing, with built-in modules for streaming, SQL, machine learning and graph processing, which also allows us to do the in-memory analysis for influence. How to analyze the influence in massive scale social data especially in the innovative platforms is still very challenging question. In this case, we are going to do more research regarding the model and algorithm to investigate more potential of influence analysis. We also believe the big data will still have great potential and value to investigate.

CONCLUSION

This dissertation conducts the problem of influence analysis in big social data. Regarding this problem, we introduce the up-to-date research literatures and elucidate the opportunities as well as challenges in detail. Surrounding many issues of influence analysis in big social data, we study different models and algorithms to provide explorations and resolutions.

Firstly, we investigate a framework for generating uncertain networks based on historical network snapshots. Four uncertainty construction models are presented to capture the uncertainty from dynamic snapshots, then the sampling techniques are employed to improve the efficiency of the algorithm. To describe the relationship of users in uncertain networks in a more practical way. For this purpose, the 2-hop expectation distance is adopted to approximate the expected number of common neighbors. Both the theoretical analysis and our experiments demonstrate the effectiveness and efficiency of our proposed methods.

Secondly, we propose a practical probing framework to explore the dynamic of networks. The probing framework takes the community as a unit and updates network topology by only probing b communities instead of searching the entire network. Besides, a divide-and-conquer strategy is applied with dynamic programming technique to maximize the community-based influence. The comprehensive experiment results show that our model can achieve comparable influence diffusion performance compared to the node-based probing algorithm while having much better efficiency and more applicable to large-scale networks.

Thirdly, according to the observations in real life, we propose model $ICOT$ which incorporates both diffusion decay and opportunistic acceptance selection for dynamic networks. In addition, we develop model $BICOT$ to control the balance between influence depth and breadth. We take the first step to explore the potential of broad influence maximization. Through comprehensive experiments tests, the results show that our model can achieve a comparable influence diffusion result like the learning-based algorithm but do not need the

strict input requirement; and at the same time, our models have a much broader influence coverage.

Fourthly, we solve the influence maximization problem in a heterogeneous information network which combing the data from both sensed cyber-physical world and online social world. Four behavior patterns and corresponding formulated functions are proposed to model the users' behavior in sensed cyber-physical world. We adopt the state-of-the-art influence maximization technique and differential privacy to achieve an efficient influence maximization algorithm with privacy protection. The real life data experiments verified that the framework works well for the problem of influence maximization and our algorithm is outperformed the up-to-date resolutions.

Fifthly, we present two novel models *TIC* and *TLT* which extend the practicality of the classical *IC* and *LT* models for influence maximization. The theoretical analysis shows that the two new models we propose both follow the monotonicity and submodularity. This result could help us to design simple greedy algorithm with a guaranteed approximate ratio $(1 - 1/e)$. Both the synthetic and real social network data are tested by our implementation on Hadoop and Spark platforms show that our new algorithm could solve the problem efficiently and effectively.

At the end of this dissertation, we point out several very important, challenging, and potential further work directions for other scholars reference.

REFERENCES

[1] Charu C Aggarwal and S Yu Philip. A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):609–623, 2009.

[2] Rakesh Agrawal. Nature of information, people, and relationships in digital social networks. *IEEE Data Engineering Bulletin Issues*, 36(3):21–32, 2013.

[3] Rakesh Agrawal, Michalis Potamias, and Evimaria Terzi. Learning the nature of information in social networks. In *International AAAI Conference On Web And Social Media (ICWSM)*, 2012.

[4] Chunyu Ai, Longjiang Guo, Zhipeng Cai, and Yingshu Li. Processing area queries in wireless sensor networks. In *The 5th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2009)*, pages 1–8. IEEE, 2009.

[5] Chunyu Ai, Meng Han, Jinbao Wang, and Mingyuan Yan. An efficient social event invitation framework based on historical data of smart devices. In *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on*, pages 229–236. IEEE, 2016.

[6] Hussah Albinali, Meng Han, Jinbao Wang, Hong Gao, and Yingshu Li. The roles of social network mavens. In *The 12th International Conference on Mobile Ad-hoc and Sensor Networks (MSN 2016)*, pages 1–12, 2016.

[7] Angelos-Christos G Anadiotis, Charalampos Z Patrikakis, and A Murat Tekalp. Information-centric networking for multimedia, social and peer-to-peer communications. *Transactions on Emerging Telecommunications Technologies*, 25(4):383–391, 2014.

[8] David B Bahr, Raymond C Browning, Holly R Wyatt, and James O Hill. Exploiting social networks to mitigate the obesity epidemic. *Obesity*, 17(4):723–728, 2009.

[9] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[10] Kshipra Bhawalkar, Sreenivas Gollapudi, and Kamesh Munagala. Coevolutionary opinion formation games. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 41–50. ACM, 2013.

[11] Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Optimal geo-indistinguishable mechanisms for location privacy. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 251–262. ACM, 2014.

[12] Ilaria Bordino, Gianmarco De Francisci Morales, Ingmar Weber, and Francesco Bonchi. From machu_picchu to rafting the urubamba river: anticipating information needs via the entity-query graph. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 275–284. ACM, 2013.

[13] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 946–957. SIAM, 2014.

[14] Dustin Bortner and Jiawei Han. Progressive clustering of networks using structure-connected order of traversal. In *IEEE 26th International Conference on Data Engineering (ICDE)*, pages 653–656. IEEE, 2010.

[15] Ceren Budak and Rakesh Agrawal. On participation in group chats on twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pages 165–176. ACM, 2013.

[16] Scott Burton, Richard Morris, Michael Dimond, Joshua Hansen, Christophe Giraud-Carrier, Joshua West, Carl Hanson, and Michael Barnes. Public health community mining in youtube. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 81–90. ACM, 2012.

[17] Zhipeng Cai, Zhi-Zhong Chen, and Guohui Lin. A 3.4713-approximation algorithm for the capacitated multicast tree routing problem. *Theoretical Computer Science*, 410(52):5415–5424, 2009.

[18] Zhipeng Cai, Zhi-Zhong Chen, Guohui Lin, and Lusheng Wang. An improved approximation algorithm for the capacitated multicast tree routing problem. In *The 2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA2008)*, pages 286–295, St. Johns Canada, August 2008. Springer.

[19] Zhipeng Cai, Yueming Duan, and Anu G Bourgeois. Delay efficient opportunistic routing in asynchronous multi-channel cognitive radio networks. *Journal of Combinatorial Optimization*, 29(4):815–835, 2015.

[20] Zhipeng Cai, Randy Goebel, and Guohui Lin. Size-constrained tree partitioning: approximating the multicast k-tree routing problem. *Theoretical Computer Science*, 412(3):240–245, 2011.

[21] Zhipeng Cai, Randy Goebel, and Guohui Lin. Size-constrained tree partitioning: A story on approximation algorithm design for the multicast k-tree routing problem. In *The 3nd Annual International Conference on Combinatorial Optimization and Applications (COCOA 2009)*, pages 363–374, Yellow Mountains China, June 2009. Springer.

[22] Zhipeng Cai, Randy Goebel, Mohammad R Salavatipour, Yi Shi, Lizhe Xu, and Guohui Lin. Selecting genes with dissimilar discrimination strength for sample class prediction. In *The 5th Asia-Pacific Bioinformatics Conference (APBC 2007)*, pages 81–90, HONG KONG, January 2007.

[23] Zhipeng Cai, Zaobo He, Xin Guan, and Yingshu Li. Collective data-sanitization for preventing sensitive information inference attacks in social networks. *IEEE Transactions on Dependable and Secure Computing*, 1(99):1–1, 2016.

[24] Zhipeng Cai, Guohui Lin, and Guoliang Xue. Improved approximation algorithms for the capacitated multicast routing problem. In *The 11th International Computing and Combinatorics Conference (COCOON 2005)*, pages 136–145, Kunming China, August 2005. Springer.

[25] Xin Cao, Gao Cong, and Christian S Jensen. Mining significant semantic locations from gps data. volume 3, pages 1009–1020. VLDB Endowment, 2010.

[26] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, pages 721–730. ACM, 2009.

[27] Quan Chen, Hong Gao, Siyao Cheng, and Zhipeng Cai. Approximate scheduling and constructing algorithms for minimum-energy multicasting in duty-cycled sensor networks. In *International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI 2015)*, pages 163–168, Beijing China, October 2015. IEEE.

[28] Quan Chen, Hong Gao, Siyao Cheng, Jianzhong Li, and Zhipeng Cai. Distributed non-structure based data aggregation for duty-cycle wireless sensor networks. In *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017)*. IEEE, May 2017.

[29] Wei Chen, Wei Lu, and Ning Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 2012, pages 1–5, 2012.

[30] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th international*

conference on Knowledge discovery and data mining (SIGKDD), pages 1029–1038. ACM, 2010.

[31] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 199–208. ACM, 2009.

[32] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *IEEE 10th International Conference on Data Mining (ICDM)*, pages 88–97. IEEE, 2010.

[33] Siyao Cheng, Zhipeng Cai, and Jianzhong Li. Curve query processing in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 64(11):5198–5209, 2015.

[34] Siyao Cheng, Zhipeng Cai, Jianzhong Li, and Xiaolin Fang. Drawing dominant dataset from big sensory data in wireless sensor networks. In *2015 IEEE Conference on Computer Communications (INFOCOM 2015)*, pages 531–539. IEEE, April 2015.

[35] Siyao Cheng, Zhipeng Cai, Jianzhong Li, and Hong Gao. Extracting kernel dataset from big sensory data in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(4):813–827, 2017.

[36] Siyao Cheng, Jianzhong Li, and Zhipeng Cai. O (ε)-approximation to physical world by sensor networks. In *2013 IEEE Conference on Computer Communications (INFOCOM 2013)*, pages 3084–3092, Turin Italy, April 2013. IEEE.

[37] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[38] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th international con-*

ference on Knowledge discovery and data mining (SIGKDD), pages 1082–1090. ACM, 2011.

[39] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379, 2007.

[40] Abhimanyu Das, Sreenivas Gollapudi, Rina Panigrahy, and Mahyar Salek. Debiasing social wisdom. In *Proceedings of the 19th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 500–508. ACM, 2013.

[41] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[42] Thang N Dinh, Dung T Nguyen, and My T Thai. Cheap, easy, and massively effective viral marketing in social networks: truth or fiction? In *Proceedings of the 23rd ACM conference on Hypertext and social media*, pages 165–174. ACM, 2012.

[43] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh international conference on Knowledge discovery and data mining (SIGKDD)*, pages 57–66. ACM, 2001.

[44] Zhuojun Duan, Wei Li, and Zhipeng Ca. Distributed auctions for task assignment and scheduling in mobile crowdsensing systems. In *The 37th IEEE International Conference on Distributed Computing Systems (ICDCS 2017)*, Atlanta USA, June 2017. IEEE.

[45] Zhuojun Duan, Mingyuan Yan, Zhipeng Cai, Xiaoming Wang, Meng Han, and Yingshu Li. Truthful incentive mechanisms for social cost minimization in mobile crowdsourcing systems. *Sensors*, 16(4):481, 2016.

[46] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. On the optimal placement of mix zones. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 216–234. Springer, 2009.

[47] Raghu K Ganti, Yu-En Tsai, and Tarek F Abdelzaher. Senseworld: Towards cyber-physical social networks. In *Information Processing in Sensor Networks (IPSN)*, pages 563–564. IEEE, 2008.

[48] Jing Gao, Jianzhong Li, Zhipeng Cai, and Hong Gao. Composite event coverage in wireless sensor networks with heterogeneous sensors. In *2015 IEEE Conference on Computer Communications (INFOCOM 2015)*, pages 217–225. IEEE, April 2015.

[49] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 241–250. ACM, 2010.

[50] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. A data-based approach to social influence maximization. volume 5, pages 73–84. VLDB Endowment, 2011.

[51] Amit Goyal, Francesco Bonchi, Laks VS Lakshmanan, and Suresh Venkatasubramanian. Approximation analysis of influence spread in social networks. *arXiv preprint arXiv:1008.2005*, 2010.

[52] Amit Goyal, Francesco Bonchi, Laks VS Lakshmanan, and Suresh Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, 3(2):179–192, 2013.

[53] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48. ACM, 2011.

[54] Sanjeev Goyal, Hoda Heidari, and Michael Kearns. Competitive contagion in networks. *Games and Economic Behavior*, 2014.

[55] Longjiang Guo, Chunyu Ai, Xiaoming Wang, Zhipeng Cai, and Yingshu Li. Real time clustering of sensory data in wireless sensor networks. In *The 28th IEEE International*

Performance Computing and Communications Conference (IPCCC 2009), pages 33–40, Phoenix USA, December 2009.

[56] Longjiang Guo, Yingshu Li, and Zhipeng Cai. Minimum-latency aggregation scheduling in wireless sensor network. *Journal of Combinatorial Optimization*, 31(1):279–310, 2016.

[57] Meng Han, Zhuojun Duan, Chunyu Ai, Forrest Wong Lybarger, Yingshu Li, and Anu G.Bourgeois. Time constraint influence maximization algorithm in the age of big data. *International Journal of Computational Science and Engineering*, 2017.

[58] Meng Han, Qilong Han Han, Lijie Li, Ji Li, and Yingshu Li. Maximizing influence in sensed heterogenous social network with privacy preservation. *International Journal of Sensor Networks*, 2017.

[59] Meng Han, Ji Li, Zhipeng Cai, and Qilong Han. Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks. In *The 9th IEEE International Conference on Social Computing and Networking (SocialCom 2016)*, pages 284–292, Atlanta USA, October 2016. IEEE.

[60] Meng Han, Jianzhong Li, and Zhaonian Zou. Finding k close subgraphs in an uncertain graph. *Jisuanji Kexue yu Tansuo*, 5(9):791–803, 2011.

[61] Meng Han, Jianzhong Li, and Zhaonian Zou. K-close: Algorithm for finding the close regions in wireless sensor networks based uncertain graph mining technology. *Journal of Software*, 22(1):131–141, 2011.

[62] Meng Han, Yi Liang, Zhuojun Duan, and Yingjie Wang. Mining public business knowledge: A case study in sec's edgar. In *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on*, pages 393–400. IEEE, 2016.

[63] Meng Han, Mingyuan Yan, Zhipeng Cai, and Yingshu Li. An exploration of broader influence maximization in timeliness networks with opportunistic selection. *Journal of Network and Computer Applications*, 63:39–49, 2016.

[64] Meng Han, Mingyuan Yan, Zhipeng Cai, Yingshu Li, Xingquan Cai, and Jiguo Yu. Influence maximization by probing partial communities in dynamic online social networks. *Transactions on Emerging Telecommunications Technologies*, 2016.

[65] Meng Han, Mingyuan Yan, Jinbao Li, Shouling Ji, and Yingshu Li. Generating uncertain networks based on historical network snapshots. In *COCOON*, pages 747–758, 2013.

[66] Meng Han, Mingyuan Yan, Jinbao Li, Shouling Ji, and Yingshu Li. Neighborhood-based uncertainty generation in social networks. *Journal of Combinatorial Optimization*, 28(3):561–576, 2014.

[67] Meng Han, Wei Zhang, and Jian-Zhong Li. Raking: An efficient k-maximal frequent pattern mining algorithm on uncertain graph database. *Jisuanji Xuebao(Chinese Journal of Computers)*, 33(8):1387–1395, 2010.

[68] Xinran He and David Kempe. Price of anarchy for the n-player competitive cascade game with submodular activation functions. In *International Conference on Web and Internet Economics*, pages 232–248. Springer, 2013.

[69] Zaobo He, Zhipeng Cai, Siyao Cheng, and Xiaoming Wang. Approximate aggregation for tracking quantiles and range countings in wireless sensor networks. *Theoretical Computer Science*, 607(3):381–390, 2015.

[70] Zaobo He, Zhipeng Cai, Siyao Cheng, and Xiaoming Wang. Approximate aggregation for tracking quantiles in wireless sensor networks. In *The 8th Annual International Conference on Combinatorial Optimization and Applications (COCOA 2014)*, pages 161–172, Maui USA, December 2014. Springer.

[71] Zaobo He, Zhipeng Cai, Qilong Han, Weitian Tong, Limin Sun, and Yingshu Li. An energy efficient privacy-preserving content sharing scheme in mobile social networks. *Personal and Ubiquitous Computing*, 20(5):833–846, 2016.

[72] Zaobo He, Zhipeng Cai, and Yingshu Li. Customized privacy preserving for classification based applications. In *Proceedings of the 1st ACM Workshop on Privacy-Aware Mobile Computing (PAMCO 2016)*, pages 37–42, Paderborn Germany, July 2016. ACM.

[73] Zaobo He, Zhipeng Cai, Yunchuan Sun, Yingshu Li, and Xiuzhen Cheng. Customized privacy preserving for inherent data and latent data. *Personal and Ubiquitous Computing*, 21(1):1–12, 2017.

[74] Zaobo He, Zhipeng Cai, and Xiaoming Wang. Modeling propagation dynamics and developing optimized countermeasures for rumor spreading in online social networks. In *IEEE 35th International Conference on Distributed Computing Systems (ICDCS 2015)*, pages 205–214, Columbus USA, June 2015. IEEE.

[75] Zaobo He, Zhipeng Cai, Jiguo Yu, Xiaoming Wang, Yunchuan Sun, and Yingshu Li. Cost-efficient strategies for restraining rumor spreading in mobile social networks. *IEEE Transactions on Vehicular Technology*, PP(99):1–1, 2016.

[76] Mehdi Heidari, Masoud Asadpour, and Hesham Faili. Smg: Fast scalable greedy algorithm for influence maximization in social networks. *Physica A: Statistical Mechanics and its Applications*, 420:124–133, 2015.

[77] Petteri Hintsanen and Hannu Toivonen. Finding reliable subgraphs from large probabilistic graphs. *Data Mining and Knowledge Discovery*, 17(1):3–23, 2008.

[78] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[79] John Hopcroft, Tiancheng Lou, and Jie Tang. Who will follow you back?: reciprocal relationship prediction. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1137–1146. ACM, 2011.

[80] Yan Huang, Min Chen, Zhipeng Cai, Xin Guan, Tomoaki Ohtsuki, and Yan Zhang. Graph theory based capacity analysis for vehicular ad hoc networks. In *Global Communications Conference (GLOBECOM 2015)*, pages 1–5, San Diego USA, December 2015. IEEE.

[81] Yan Huang, Xin Guan, Zhipeng Cai, and Tomoaki Ohtsuki. Multicast capacity analysis for social-proximity urban bus-assisted vanets. In *International Conference on Communications (ICC 2013)*, pages 6138–6142, Budapest Hungary, June 2013. IEEE.

[82] José Luis Iribarren and Esteban Moro. Impact of human activity patterns on the dynamics of information diffusion. *Physical review letters*, 103(3):038702, 2009.

[83] Shouling Ji, Zhipeng Cai, Meng Han, and Raheem Beyah. Whitespace measurement and virtual backbone construction for cognitive radio networks: From the social perspective. In *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*, pages 435–443. IEEE, 2015.

[84] Ruoming Jin, Lin Liu, and Charu C Aggarwal. Discovering highly reliable subgraphs in uncertain graphs. In *Proceedings of the 17th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 992–1000. ACM, 2011.

[85] Ruoming Jin, Lin Liu, Bolin Ding, and Haixun Wang. Distance-constraint reachability computation in uncertain graphs. volume 4, pages 551–562. VLDB Endowment, 2011.

[86] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile computing*, pages 153–181. Springer, 1996.

[87] Jagat Narain Kapur, Prasanna K Sahoo, and Andrew KC Wong. A new method for

gray-level picture thresholding using the entropy of the histogram. *Computer vision, graphics, and image processing*, 29(3):273–285, 1985.

[88] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth international conference on Knowledge discovery and data mining (SIGKDD)*, pages 137–146. ACM, 2003.

[89] Jinha Kim, Seung-Keol Kim, and Hwanjo Yu. Scalable and parallelizable processing of influence maximization for large-scale social networks? In *IEEE 29th International Conference on Data Engineering (ICDE)*, pages 266–277. IEEE, 2013.

[90] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. *World Wide Web*, 8(2):159–178, 2005.

[91] Ohbyung Kwon and Yixing Wen. An empirical study of the factors affecting social network service use. *Computers in human behavior*, 26(2):254–263, 2010.

[92] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[93] Ioannis Leftheriotis and Michail N Giannakos. Using social media for work: Losing your time or improving your work? *Computers in Human Behavior*, 31:134–142, 2014.

[94] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th international conference on World wide web*, pages 641–650. ACM, 2010.

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

[96] Jurij Leskovec, Deepayan Chakrabarti, Jon Kleinberg, and Christos Faloutsos. Realistic, mathematically tractable graph generation and evolution, using kronecker mul-

tiplication. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 133–145. Springer, 2005.

[97] Guoliang Li, Shuo Chen, Jianhua Feng, Kian-lee Tan, and Wen-syan Li. Efficient location-aware influence maximization. In *Proceedings of the 2014 international conference on Management of data (SIGMOD)*, pages 87–98. ACM, 2014.

[98] Ji Li, Siyao Cheng, Zhipeng Cai, Qilong Han, and Hong Gao. Bernoulli sampling based (epsilon, delta)-approximate frequency query in mobile ad hoc networks. In *The 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2015)*, pages 315–324, Qufu China, August 2015. Springer.

[99] Ji Li, Cai Zhipeng, Mingyuan Yan, and Yingshu Li. Using crowdsourced data in location-based social networks to explore influence maximization. In *IEEE Conference on Computer Communications (INFOCOM 2016)*, pages 1–9, San Francisco USA, April 2016. IEEE.

[100] Jianzhong Li, Siyao Cheng, Hong Gao, and Zhipeng Cai. Approximate physical world reconstruction algorithms in sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(12):3099–3110, 2014.

[101] Jinbao Li, Xiaohang Guo, Longjiang Guo, Shouling Ji, Meng Han, and Zhipeng Cai. Optimal routing with scheduling and channel assignment in multi-power multi-radio wireless sensor networks. *Ad Hoc Networks*, 31:45–62, 2015.

[102] Rong-Hua Li, Lu Qin, Jeffrey Xu Yu, and Rui Mao. Influential community search in large networks. volume 8, pages 509–520. VLDB Endowment, 2015.

[103] Yingshu Li, Chunyu Ai, Zhipeng Cai, and Raheem Beyah. Sensor scheduling for p-percent coverage in wireless sensor networks. *Cluster Computing*, 14(1):27–40, 2011.

[104] Omar Lizardo, Michael Penta, Matthew Chandler, Casey Doyle, G Korniss, Boleslaw K

Szymanski, and Jonathan Z Bakdash. Analysis of opinion evolution in a multi-cultural student social network. *Procedia Manufacturing*, 3:3974–3981, 2015.

[105] Tiancheng Lou and Jie Tang. Mining structural hole spanners through information diffusion in social networks. In *Proceedings of the 22nd international conference on World Wide Web*, pages 825–836. ACM, 2013.

[106] Tiancheng Lou, Jie Tang, John Hopcroft, Zhanpeng Fang, and Xiaowen Ding. Learning to predict reciprocity and triadic closure in social networks. *Transactions on Knowledge Discovery from Data (TKDD)*, 7(2):5, 2013.

[107] Junling Lu, Zhipeng Cai, Xiaoming Wang, Lichen Zhang, Peng Li, and Zaobo He. Primary and secondary social activity aware routing for cognitive radio networks. In *International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI 2016)*, Beijing China, October 2016. IEEE.

[108] Yanfei Lu, Jianmin Ren, Jin Qian, Meng Han, Yan Huo, and Tao Jing. Predictive contention window-based broadcast collision mitigation strategy for vanet. In *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on*, pages 209–215. IEEE, 2016.

[109] Yunmei Lu, Yun Zhu, Meng Han, Jing Selena He, and Yanqing Zhang. A survey of gpu accelerated svm. In *Proceedings of the 2014 ACM Southeast Regional Conference*, page 15. ACM, 2014.

[110] Kathy Macropol and Ambuj Singh. Scalable discovery of best clusters on large graphs. volume 3, pages 693–702. VLDB Endowment, 2010.

[111] Prateek Mittal, Charalampos Papamanthou, and Dawn Song. Preserving link privacy in social network based systems. *arXiv preprint arXiv:1208.6189*, 2012.

[112] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1):265–294, 1978.

[113] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[114] S Yu Philip, Jiawei Han, and Christos Faloutsos. *Link mining: Models, algorithms, and applications*. Springer, 2010.

[115] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *International Symposium on Computer and Information Sciences*, pages 284–293. Springer, 2005.

[116] Michalis Potamias, Francesco Bonchi, Aristides Gionis, and George Kollios. K-nearest neighbors in uncertain graphs. volume 3, pages 997–1008. VLDB Endowment, 2010.

[117] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

[118] George Ritzer et al. *The Blackwell encyclopedia of sociology*, volume 1479. Blackwell Publishing Malden, MA, 2007.

[119] Sheldon M Ross. *Introduction to probability models*. Academic press, 2014.

[120] Ryan A Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. Modeling dynamic behavior in large evolving graphs. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 667–676. ACM, 2013.

[121] Kazumi Saito, Masahiro Kimura, Kouzou Ohara, and Hiroshi Motoda. Learning asynchronous-time information diffusion models and its application to behavioral data analysis over social networks. *arXiv preprint arXiv:1204.4528*, 2012.

[122] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 67–75. Springer, 2008.

[123] Marcel Salathé, Maria Kazandjieva, Jung Woo Lee, Philip Levis, Marcus W Feldman, and James H Jones. A high-resolution human contact network for infectious disease transmission. volume 107, pages 22020–22025. National Acad Sciences, 2010.

[124] Tuo Shi, Siyao Cheng, Zhipeng Cai, and Jianzhong Li. Adaptive connected dominating set discovering algorithm in energy-harvest sensor networks. In *2016 IEEE Conference on Computer Communications (INFOCOM 2016)*, pages 1–9, San Francisco USA, April 2016. IEEE.

[125] Tuo Shi, Siyao Cheng, Zhipeng Cai, Yingshu Li, and Jianzhong Li. Retrieving the maximal time-bounded positive influence set from social networks. *Personal and Ubiquitous Computing*, 20(5):717–730, 2016.

[126] Tuo Shi, Siyao Cheng, Zhipeng Cai, Yingshu Li, and Jianzhong Li. Exploring connected dominating sets in energy harvest networks. *IEEE/ACM Transactions on Networking*, PP(99):1–15, 2017.

[127] Tuo Shi, Siyao Cheng, Jianzhong Li, and Zhipeng Cai. Constructing connected dominating sets in battery-free networks. In *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017)*. IEEE, May 2017.

[128] Tuo Shi, Jialin Wan, Siyao Cheng, Zhipeng Cai, Yingshu Li, and Jianzhong Li. Time-bounded positive influence in social networks. In *International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI 2015)*, pages 134–139, Beijing China, October 2015. IEEE.

[129] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno

Lina, et al. High-resolution measurements of face-to-face contact patterns in a primary school. *PloS one*, 6(8):e23176, 2011.

[130] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(4):354–356, 1969.

[131] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 807–816. ACM, 2009.

[132] Jie Tang, Bo Wang, Yang Yang, Po Hu, Yanting Zhao, Xinyu Yan, Bo Gao, Minlie Huang, Peng Xu, Weichang Li, et al. Patentminer: topic-driven patent analysis and mining. In *Proceedings of the 18th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 1366–1374. ACM, 2012.

[133] Jie Tang, Sen Wu, and Jimeng Sun. Confluence: Conformity influence in large social networks. In *Proceedings of the 19th international conference on Knowledge discovery and data mining (SIGKDD), pages=347–355, year=2013, organization=ACM.*

[134] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 990–998. ACM, 2008.

[135] Xuning Tang and Christopher C Yang. Ranking user influence in healthcare social media. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):73, 2012.

[136] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 International Conference on Management of Data (SIGMOD)*, pages 1539–1554. ACM, 2015.

[137] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 international conference on Management of data (SIGMOD)*, pages 75–86. ACM, 2014.

[138] My T Thai, Zhipeng Cai, and Ding-Zhu Du. Genetic networks: processing data, regulatory network modelling and their analysis. *Optimisation Methods and Software*, 22(1):169–185, 2007.

[139] Ferdinand Tönnies. *Gemeinschaft und Gesellschaft: Abhandlung des Communismus und des Socialismus als empirischer Culturformen*. Fues, 1887.

[140] Chinh Vu, Zhipeng Cai, and Yingshu Li. Distributed energy-efficient algorithms for coverage problem in adjustable sensing ranges wireless sensor networks. *Discrete Mathematics, Algorithms and Applications*, 1(3):299–317, 2009.

[141] Chi Wang, Wei Chen, and Yajun Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, 25(3):545, 2012.

[142] Jinbao Wang, Zhipeng Cai, Chunyu Ai, Donghua Yang, Hong Gao, and Xiuzhen Cheng. Differentially private k-anonymity: Achieving query privacy in location-based services. In *International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI 2016)*, Beijing China, October 2016. IEEE.

[143] Xiaoming Wang, Yaguang Lin, Lichen Zhang, and Zhipeng Cai. A double pulse control strategy for misinformation propagation in human mobile opportunistic networks. In *The 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2015)*, pages 571–580, Qufu China, August 2015. Springer.

[144] Xiaoming Wang, Yaguang Lin, Shanshan Zhang, and Zhipeng Cai. A social activity and physical contact-based routing algorithm in mobile opportunistic networks for emergency response to sudden disasters. *Enterprise Information Systems*, 11(5):597–626, 2015.

[145] Xiaoming Wang, Yaguang Lin, Yanxin Zhao, Lichen Zhang, Juhua Liang, and Zhipeng Cai. A novel approach for inhibiting misinformation propagation in human mobile opportunistic networks. *Peer-to-Peer Networking and Applications*, 10(2):1–18, 2016.

[146] Xinjing Wang, Longjiang Guo, Chunyu Ai, Jinbao Li, and Zhipeng Cai. An urban area-oriented traffic information query strategy in vanets. In *The 8th International Conference on Wireless Algorithms, Systems and Applications (WASA 2013)*, pages 313–324, Zhangjiajie China, August 2013. Springer.

[147] Yingjie Wang, Zhipeng Cai, Guisheng Yin, Yang Gao, and Qingxian Pan. A trust measurement in social networks based on game theory. In *The 4th International Conference on Computational Social Networks (CSoNet 2015)*, pages 236–247, Beijing China, August 2015. Springer.

[148] Yingjie Wang, Zhipeng Cai, Guisheng Yin, Yang Gao, Xiangrong Tong, and Qilong Han. A game theory-based trust measurement model for social networks. *Computational Social Networks*, 3(1):1–16, 2016.

[149] Yingjie Wang, Zhipeng Cai, Guisheng Yin, Yang Gao, Xiangrong Tong, and Guanying Wu. An incentive mechanism with privacy protection in mobile crowdsourcing systems. *Computer Networks*, 102:157–171, 2016.

[150] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *Proceedings of the 16th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 1039–1048. ACM, 2010.

[151] Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *Nature*, 393(6684):440–442, 1998.

[152] Ralf Wölfer and Herbert Scheithauer. Social influence and bullying behavior: Intervention-based network dynamics of the fairplayer. manual bullying prevention program. *Aggressive behavior*, 40(4):309–319, 2014.

[153] Michael Workman. New media and the changing face of information technology use: The importance of task pursuit, social influence, and experience. *Computers in Human Behavior*, 31:111–117, 2014.

[154] Yonghui Xiao and Li Xiong. Dynamic differential privacy for location based applications. *CoRR*, abs/1410.5919, 2014.

[155] Ke Xu and Xinfang Zhang. Mining community in mobile social network. *Procedia Engineering*, 29:3080–3084, 2012.

[156] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 824–833. ACM, 2007.

[157] Mingyuan Yan, Chunyu Ai, Meng Han, Zhipeng Cai, and Yingshu Li. Data aggregation scheduling in probabilistic wireless networks with cognitive radio capability. In *The 59th annual IEEE Global Communications Conference (GLOBECOM 2016)*, DC USA, December 2016.

[158] Mingyuan Yan, Shouling Ji, Meng Han, Yingshu Li, and Zhipeng Cai. Data aggregation scheduling in wireless networks with cognitive radio capability. In *Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference on*, pages 513–521. IEEE, 2014.

[159] Lei Yu, Karan Sapra, Haiying Shen, and Lin Ye. Cooperative end-to-end traffic redundancy elimination for reducing cloud bandwidth cost. In *The 20th IEEE International Conference on Network Protocols (ICNP 2012)*, pages 1–10, Austin USA, October 2012. IEEE.

[160] Lei Yu, Haiying Shen, Karan Sapra, Lin Ye, and Zhipeng Cai. Core: Cooperative end-to-end traffic redundancy elimination for reducing cloud bandwidth cost. *IEEE Transactions on Parallel and Distributed Systems*, 28(2):446–461, 2017.

[161] Xiao Yu, Ang Pan, Lu-An Tang, Zhenhui Li, and Jiawei Han. Geo-friends recommendation in gps-based cyber-physical social network. In *Advances in Social Networks Analysis and Mining (ASONAM)*, pages 361–368. IEEE, 2011.

[162] Ye Yuan, Lei Chen, and Guoren Wang. Efficiently answering probability threshold-based shortest path queries over uncertain graphs. In *International Conference on Database Systems for Advanced Applications*, pages 155–170. Springer, 2010.

[163] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. Social influence locality for modeling retweeting behaviors. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 13, pages 2761–2767, 2013.

[164] Kejia Zhang, Qilong Han, Zhipeng Cai, Guisheng Yin, and Junyu Lin. Doami: A distributed on-line algorithm to minimize interference for routing in wireless sensor networks. *Theoretical Computer Science*, 2016.

[165] Kejia Zhang, Qilong Han, Zhipeng Cai, Guisheng Yin, and Junyu Lin. Metric and distributed on-line algorithm for minimizing routing interference in wireless sensor networks. In *The 9th Annual International Conference on Combinatorial Optimization and Applications (COCOA 2015)*, pages 279–292. Springer, Houston USA, December 2015.

[166] Lichen Zhang, Zhipeng Cai, Peng Li, and Xiaoming Wang. Exploiting spectrum availability and quality in routing for multi-hop cognitive radio networks. In *The 11th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2016)*, pages 283–294, Bozeman USA, August 2016. Springer.

[167] Lichen Zhang, Zhipeng Cai, Junling Lu, and Xiaoming Wang. Mobility-aware routing in delay tolerant networks. *Personal and Ubiquitous Computing*, 19(7):1111–1123, 2015.

[168] Lichen Zhang, Zhipeng Cai, Junling Lu, and Xiaoming Wang. Spacial mobility prediction based routing scheme in delay/disruption-tolerant networks. In *International*

Conference on Identification, Information and Knowledge in the Internet of Things (IIKI 2014), pages 274–279, Beijing China, October 2014. IEEE.

[169] Lichen Zhang, Zhipeng Cai, and Xiaoming Wang. Fakemask: a novel privacy preserving approach for smartphones. *IEEE Transactions on Network and Service Management*, 13(2):335–348, 2016.

[170] Lichen Zhang, Xiaoming Wang, Junling Lu, Peng Li, and Zhipeng Cai. An efficient privacy preserving data aggregation approach for mobile sensing. *Security and Communication Networks*, 9(16):3844–3853, 2016.

[171] Jichang Zhao, Junjie Wu, Xu Feng, Hui Xiong, and Ke Xu. Information propagation in online social networks: a tie-strength perspective. *Knowledge and Information Systems*, 32(3):589–608, 2012.

[172] Xu Zheng, Zhipeng Cai, Jianzhong Li, and Hong Gao. Scheduling flows with multiple service frequency constraints. *IEEE Internet of Things Journal*, PP(99):1–1, 2016.

[173] Xu Zheng, Zhipeng Cai, Jianzhong Li, and Hong Gao. A study on application-aware scheduling in wireless networks. *IEEE Transactions on Mobile Computing*, PP(99):1–1, 2016.

[174] Xu Zheng, Zhipeng Cai, Jianzhong Li, and Hong Gao. An application-aware scheduling policy for real-time traffic. In *The 35th International Conference on Distributed Computing Systems (ICDCS 2015)*, pages 421–430, Columbus USA, June 2015. IEEE.

[175] Xu Zheng, Zhipeng Cai, Jianzhong Li, and Hong Gao. Location-privacy-aware review publication mechanism for local business service systems. In *The 36th Annual IEEE International Conference on Computer Communications (INFOCOM 2017)*. IEEE, May 2017.

[176] Xu Zheng, Jianzhong Li, Hong Gao, and Zhipeng Cai. Capacity of wireless networks with multiple types of multicast sessions. In *The 15th ACM International Symposium*

on Mobile Ad Hoc Networking and Computing (MOBIHOC 2014), pages 135–144, Philadelphia USA, August 2014. ACM.

[177] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[178] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph clustering based on structural/attribute similarities. volume 2, pages 718–729. VLDB Endowment, 2009.

[179] Tongxin Zhu, Xinrui Wang, Siyao Cheng, Zhipeng Cai, and Jianzhong Li. Critical point aware data acquisition algorithm in sensor networks. In *The 10th International Conference on Wireless Algorithms, Systems, and Applications (WASA 2015)*, pages 798–808, Qufu China, August 2015. Springer.

[180] Honglei Zhuang, Yihan Sun, Jie Tang, Jialin Zhang, and Xiaoming Sun. Influence maximization in dynamic social networks. In *IEEE 13th International Conference on Data Mining (ICDM)*, pages 1313–1318. IEEE, 2013.

[181] Zhaonian Zou, Hong Gao, and Jianzhong Li. Discovering frequent subgraphs over uncertain graph databases under probabilistic semantics. In *Proceedings of the 16th international conference on Knowledge discovery and data mining (SIGKDD)*, pages 633–642. ACM, 2010.

[182] Zhaonian Zou, Jianzhong Li, Hong Gao, and Shuo Zhang. Finding top-k maximal cliques in an uncertain graph. In *IEEE 26th International Conference on Data Engineering (ICDE)*, pages 649–652. IEEE, 2010.