Trust-aware Link Prediction in Online Social Networks

by

Samah Aloufi

A thesis Submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Degree of Master in Computer Science

Ottawa-Carleton Institute for Computer Science
School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Ottawa, Ontario, Canada, 2012

Abstract

As people go about their lives, they form a variety of social relationships, such as family, friends, colleagues, and acquaintances, and these relationships differ in their strength, indicating the level of trust among these people. The trend in these relationships is for people to trust those who they have met in real life more than unfamiliar people whom they have only met online. In online social network sites the objective is to make it possible for users to post information and share albums, diaries, videos, and experiences with a list of contacts who are real-world friends and/or like-minded online friends. However, with the growth of online social services, the need for identifying trustworthy people has become a primary focus in order to protect users' vast amounts of information from being misused by unreliable users. In this thesis, we introduce the Capacity- first algorithm for identifying a local group of trusted people within a network. In order to achieve the outlined goals, the algorithm adapts the Advogato trust metric by incorporating weighted social relationships. The Capacity-first algorithm determines all possible reliable users within the network of a targeted user and prevents malicious users from accessing their personal network. In order to evaluate our algorithm, we conduct experiments to measure its performance against other well-known baseline algorithms. The experimental results show that our algorithm's performance is better than existing alternatives in finding all possible trustworthy users and blocking unreliable ones from violating users' privacy.

Acknowledgements

Foremost, I would like to express my heart-felt gratitude to my father, Bader, for his encouragement and persistent support, and for standing by my side throughout my life. No words can express my love and appreciation to him. Thank you father for everything you did for me and thank you for every minute you spent here in Canada to support me in order to accomplish my dream.

My sincere thanks go to my supervisor Prof. Abdulmotaleb El Saddik for his guidance, words of encouragement, valuable comments, and support during these last few years.

Besides my supervisor, I am grateful to Dr. Heung-Nam Kim, for his invaluable help and feedback, hours of meetings and discussions, and his patience. Without his support and assistance, this thesis would not have been possible. Thank you Nami.

I would like to express my deepest thanks to my beloved mother and siblings, who have been a source of infinite love, strength, and support throughout my life.

I am thankful to Dr. Mohamad Eid who encouraged and advised me throughout my graduate studies. Special thanks to my best friends for their comments, help, and encouragement during this challenging time, the long hours of working together, and the fun we had. I will not forget to thank all Discover and MCR lab mates for the good times I had working in such a friendly environment.

Lastly, I dedicate this work to my parents, Bader and Souad.

Table of Contents

Abstract ii

Acknowledgements iii

Table of Contents iv

List of Figures vi

List of Tables viii

Chapter 1 Introduction 1

1.1 Motivation 3

1.2 Problem Statement 5

1.3 Thesis Contribution 6

1.4 Author's Publication 6

1.5 Thesis Organization 7

Chapter 2 Background and Related Work 9

2.1 Trust in Social Networks 9

2.1.1 Defining Trust 9

2.1.2 Online Social Networks 13

2.1.3 Trust in Online Social Networks 15

2.2 Advogato Trust Metric 20

2.3 Related Work 23

Chapter 3 Identifying People of Trust 29

3.1 Overview 29

3.2 Problem Formalization 30

3.3 Adapted Advogato Algorithm 31

3.3.1 Assigning a Capacity to a Seed Node 31

3.3.2 Building a Weighted Personal Network 33

3.3.3 Propagating the Capacity through the Personal Network 39

3.3.4 Capacity-first Maximum Flow 42

Chapter 4 Application Scenarios 50

4.1 System Overview 51

4.2 Controlling Access Permission in Social Network Services 54

4.3 Recommending New Friends 58

Chapter 5 Experiments and Results 62

5.1 Experimental Networks 62

5.2 Evaluation Design and Metrics 63

5.2.1 Evaluation Metrics 64

5.2.2 Baseline Algorithms 67

5.3 Sensitivity to Parameters 69

5.3.1 Decay Factor *d* 70

5.3.2 Capacity Size *m* 72

5.4 Effect of Capacity-first Maximum Flow 74

5.4.1 Small Dataset Comparisons 74

5.4.2 Large Dataset Comparison 78

5.5 Comparison with Other Methods 84

5.5.1 Comparison for Small Dataset 84

5.5.2 Comparison for large dataset 89

Chapter 6 Conclusion and Future Work 94

List of Figures

Figure 2.1: Illustrates transitive property in a simple chain between three people in the network [21]. 18
Figure 2.2: A received information about G from two paths: A-> B-> D-> G, and A-> C-> F-> G. A has to combine this information about G to form his opinion about G [21]. 19
Figure 2.3: The procedure of transforming a graph's nodes into Advogato structure [36]. 22
Figure 3.1: Assigning initial capacity for a given user. 33
Figure 3.2: Original network that shows relationships among people in a simple social networ....34
Figure 3.3: Personal network of user A that shows the relationships. 35
Figure 3.4: Personal network of user B that shows the relationships 35
Figure 3.5: WPN of user A....38
Figure 3.6: WPN of user B. 38
Figure 3.7: Propagating capacity through WPN. 40
Figure 3.8: Selecting the closest relationships in case of conflict. 41
Figure 3.9: The WPN for seed *S.* 47
Figure 3.10: Initial capacity computation and it's propagation among nodes in the WPN. 48
Figure 3.11: conversion the WPN into Advogato structure. 49
Figure 4.1: Access Permissions and Friend Recommender system architecture. 53
Figure 4.2: Partial sequence diagram for Access Permissions and Friend Recommender system. 59
Figure 5.1: Recall, precision, and error-hit scores at different values of d when *m*=1 for Top-10 72
Figure 5.2: Precision and recall of Advogato and Capacity-first at different N cases in Training 90%. 75
Figure 5.3: Precision and recall of Advogato and Capacity-first at different N cases in Training 80%. 76

Figure 5.4: Error-rate of Training 90% at different N values. 77
Figure 5.5: Error-rate of Training 80% at different N values. 78
Figure 5.6: : Precision and recall of Advogato and Capacity-first at different N values when m=1. 81
Figure 5.7: Error-hit obtained by Capacity-first and Advogato when m=1. 82
Figure 5.8: Error-hit obtained by Capacity-first and Advogato when m=6. 83
Figure 5.9: Precision and recall of baseline algorithms and capacity-first in Training 90%. .86
Figure 5.10: Precision and recall of baseline algorithms and capacity-first in Training 80%. 86
Figure 5.11: Error-rate of Training 90% at different N values. 88
Figure 5.12: Error-rate of Training 80% at different N values. 88
Figure 5.13: Precision and recall of baseline algorithms and Capacity-first at different N values when m=6.....90
Figure 5.14: Precision and recall of baseline algorithms and Capacity-first at different N values when m=1.....91
Figure 5.15: Error-rate of baseline algorithms and Capacity-first at different N when m=1..92
Figure 5.16: Error-rate of baseline algorithms and Capacity-first at different N when m=6..93

List of Tables

Table 4.1: Main differences between our work and Advogato trust metric.....61
Table 5.1: Small and large dataset descriptions. 63
Table 5.2: Recall, precision, and error-hit scores at different values of d when m=1 for Top-10. 71
Table 5.3: precision, recall, and error hit at different values of d when m=6 for Top-10.....71
Table 5.4: shows recall, precision, and error hit of *m*'s values at Top-10. 73
Table 5.5: Precision of Advogato and Capacity-first when m=6. 80
Table 5.6: Recall of Advogato and Capacity-first when m=6.....80

Chapter 1

Introduction

Human societies are built upon social relationships. Each person in this world interacts with his/her communities through social relations (for example, family, neighbour, school, or sports team) and the interaction among these people is dependent on the relationships that connect them through what is known as a *social network [49]*. From the early days of the World Wide Web, the goal was to create a dynamic environment that allowed people to exchange information and communicate with each other easily [51]. With the development of Web 2.0, people are no longer solely recipients of information. Instead, they become active participants in generating information, sharing this information with others via online communities such as forums and blogs [49]. In online communities, people are connected through common interests, even if they do not have an interpersonal relationship. In addition to online communities, Internet users want to bring their offline social networks to the Internet world in order to maintain these relationships, share information, and socialize even when physically separated from other members of their community. Today, they can do this with Online Social Network Sites (OSNSs). These sites are the digital format for social networks which allow users to build their networks based on social relationships [49], [51].

OSNSs are web-based services that allow users to create profiles and share information, experiences, and media with a list of users. These users can be family members, friends, colleagues, and so on. Most OSNSs have similar functionalities such as private messaging, commenting on a friend's profile, uploading photos or videos, and discussion groups. The

main objective of OSNSs is to use these features to maintain existing relationships and to establish new ones. However, as Boyed states, "*What makes social network[ing] sites unique is not that they allow individuals to meet strangers, but rather that they enable users to articulate and make visible their social networks*" [8],[49]. Exposing one's social network makes these sites unique, but much more complicated at the same time.

For example, OSNSs, or we can refer to them as Social network services (SNSs) or Web-based Social Networks (WBSNs), require that users build these networks, but they all differ in their friendship request confirmation processes. Some SNSs, like Facebook, require bi-direction confirmation, and the connection between pairs of users is labeled "Friend". Others, like Twitter, accept unidirectional confirmation, and the only link between two users is known as "Follower"[8].

In recent years, WBSNs have dramatically increased in popularity among online users. A large number of Internet users are participants in social network services, and they have integrated these websites into their lives by using them on a daily basis, spending a good amount of time actively participating. According to Facebook, the number of active users in April, 2007 was 20 million, and this number has increased to over 800 million active users in 2011[1],[2]. They upload on average more than 250 million photos per day and interact with more than 900 million objects [1]. This enormous amount of valuable information and personal data make online social networks a prime target for attackers or malicious users. Sophos' security reports for 2010 find that the number of spam and malware incidents has risen by 70% in the last 12 months. Also, 57% of the participants in Sophos' security report have been spammed and 36% of them received malware [3],[4]. This growth in the social

networking world and the number of new users (in the hundreds of millions) make identifying malicious users within the networks very difficult. As Graham Cluley, senior technology consultant at Sophos, said “We shouldn't forget that Facebook is by far the largest social network and you'll find more bad apples in the biggest orchard”.

Since then, members in online social networks have been restricting their information and data sharing to a closed group of trustworthy users in order to protect their information from being misused. This result in a loss of many of the benefits of SNSs. Accordingly, identifying trustworthy people is a primary concern in online social networks. For this reason, this thesis presents a proposal of a new algorithm for discovering people of trust within a network. In order to achieve this goal, this Capacity-first maximum flow algorithm adapts the Advogato trust metric by incorporating weighted social relationships and propagating the initial capacity based on the strength of the relationship to other nodes. This enables us to identify people of trust and to prevent unreliable users from impinging on a personal connection.

1.1 Motivation

Millions of users, from all cultures and countries, join SNSs and spend time communicating with a list of friends by sharing information, photos, and experiences, indicating that online social networks have become one of the important daily practices in their lives. Participants in WBSNs are likely to share a significant amount of information with trusted people,

however, this exponential growth in the size and number of registered users does not come without a price. As mentioned in the Sophos' 2010 Security threat report above, large number of SNSs' users admitted that security and privacy are becoming a concern. Members in online social networks reveal valuable and sensitive information when sharing it with compatible people, yet these posts of personal information are an invaluable source of private information for attackers or malicious users. One of the motivations that lead to this work is that participants in online social networks want to share information, albums, videos, and opinions with like-minded, trusted users, but online social networks are often misused by spammers and malicious users who violate legitimate users' privacy. Accordingly, finding trustworthy people in online social networks to share information with is a fundamental step to ensure that there is no privacy violation. Questions that come to the surface are: how can we identify reliable users within the chain of connection? and how can we prevent spammers and malicious users from accessing the network and misusing information? In order to answer these questions, we look to human characteristics like how people trust those who have strong ties with them, such as friends, family members or colleagues, rather than unfamiliar ones. With the adoption of this feature, by maintaining explicit relationships among users in an online social network, we can quantify the strength of each relationship use it to indicate a trust level. As an example, Facebook users could express precise social relations that connect them to their list of friends, such as "family member", "close friend", "co-worker", "neighbour", or "acquaintance", rather than using the term "friend" as the only relationship. To address this issue, a new method to identify trustworthy people within a network is proposed: the Capacity-first maximum algorithm adapts the Advogato trust metric

by incorporating weighted social relationships to identify a local trustworthy group of users within a social network.

1.2 Problem Statement

Online social networks are web-based services where participants communicate and share information, opinions, videos, photos, and post news and thoughts with a list of like-minded people who are trusted. However, the enormous growth of social network sites in recent years has lead to unreliable users who misuse these services by committing acts considered to be privacy violations. People in online social networks, therefore, restrict their information, and the sharing of this information is becoming more closed, making social network services lose their original benefits. This is due to the existence of malicious users impinging within a chain of social connections. As a result, identifying a trustworthy group of people, and thus preventing unreliable ones from accessing private data so participants' information can be protected from spammers and malicious users is a fundamental problem in online social networks. To address this problem, we are introducing the Capacity-first maximum flow method in order to determine a local group of trust based on a given user's perspective. Our method adapts Advogato trust metric by incorporate weighted social relationships in order to propagate an initial capacity along a chain of personal connections and repeatedly determines reliable people by using Capacity-first maximum algorithm.

1.3 Thesis Contribution

The main contributions of this thesis are illustrated in the following points:

1. Analysis and effectively incorporating weighted social relationships into the Advogato trust metric. We propagate the initial capacity to successor nodes based on the strength of the relationships that exist between the seed and its neighbours.
2. Design and development of the Capacity-first maximum flow algorithm that is designed to identify a local group of trust based on a given user's perspective. The proposed algorithm in this work identifies a local group of reliable users based on their level of trust rather than using network flow. Moreover, our algorithm returns a ranked list of users based on their level of trust instead of categorizing them by using a binary scale like in the Advogato method.
3. We show how our algorithm effectively works in identifying trustworthy users and preventing malicious users by providing detailed experimental evaluations with a real dataset. Also, we present how our algorithm can be used for diverse social networks services in order to offer some real benefits to users.

1.4 Author's Publication

- Samah Al-Oufi, Heung-Nam Kim, and Abdulmotaleb El Saddik. Controlling Privacy

with Trust-aware Link Prediction in Online Social Networks. In proceeding of the 3rd International Conference on Internet Multimedia Computing and Service, ACM ICIMCS'11, August 5–7, 2011, Chengdu, Sichuan, China.

1.5 Thesis Organization

This thesis is organized as follows: Chapter 2 presents background information and related work. In the background information we provide the literature review of trust's definition in different disciplines other than trust in online social networks. We also survey some studies that are related to our work.

Chapter 3 describes the proposed algorithm in detail: Capacity-first maximum flow. This chapter illustrates the algorithm procedures and shows how to build the user's weighted personal network that is based on their relationships. Also, it describes how to calculate the initial capacity and then propagate it through the network. There is then a computation for identifying the reliable people within the network.

Chapter 4 discusses some application scenarios where the algorithm is useful and can be applied.

Chapter 5 presents evaluation metrics that are used to measure the algorithm performance and some baseline algorithms for identifying trustworthy users and recommending new

friends. Also, it presents experimental results that show the performance of this approach by comparing it with earlier works.

Finally, Chapter 6 summarizes the thesis work and presents potential future work for the research.

Chapter 2 Background and Related Work

" *...trust is a term with many meanings.*"

Oliver Williamson

" Perhaps there is no single variable which so thoroughly influences interpersonal and group behavior as does trust."

Golembiewski and McConkie, 1975

2.1 Trust in Social Networks

The concept of trust is as old as human society because it plays an essential role in human interactions. Yet trust is a complicated concept that is difficult to define. Scholars have spent significant time and effort studying and defining trust, but they have not agreed on a common definition. Since trust has been studied in a variety of disciplines, it has resulted in a variety of definitions. Ranging from the study of human society to computer science, the concept of trust can have many uses and contexts, so each domain has different specifications when defining it. For the sake of addressing trust in the context of online social networks, we will briefly provide definitions in several related disciplines: psychology, sociology, and computer science [33].

2.1.1 Defining Trust

Cambridge Dictionary Online defines trust as a "belief or confidence in the honesty, goodness, skill or safety of a person, organization or thing" [5]. This is a starting point for defining the general meaning of trust.

Trust needs to be defined more precisely according to the domain where it is applied. By surveying the literature, we found variations in the definitions of trust. These variations are a consequence of two things. The first reason is that trust is an abstract concept that is usually used interchangeably with related concepts like confidence, reliability, and capability. Secondly, trust is a multi-dimensional concept that most likely incorporates emotional, behavioural, and cognitive dimensions. Thus, when many studies are conducted on trust over a variety of disciplines, each of these studies reaches a different understanding of the concept [57].

Psychologists and sociologists view trust as an integral part of a person's life and a vital part of relationships and interactions between people. Some studies of trust in psychology and sociology focus on interpersonal trust, while other studies pay more attention to the motivational dimension [57]. For example, one of the earliest works in defining trust in psychology was conducted by Deutsch. Deutsch defined trust in the following way:

"An individual may be said to have trust in the occurrence of an event if he expects its occurrence and his expectation leads to behavior which he perceives to have greater negative motivational consequences if the expectation is not confirmed, than positive motivational consequences if it is confirmed." [11],[7].

When Deutsch defines trust, he focuses on expectations based on previous negative and positive outcomes when the new event occurs. This definition highlights that choosing to trust someone is based on expectations, and this involves the possibility of risk and negative consequences [33].

Interpersonal trust was defined by Rotter in 1967 as "*an expectancy held by individuals or groups that the word, promise, verbal, or written statement of another can be relied on*'' [57] This definition is frequently cited and used by researchers in order to study interpersonal trust. To distinguish the differences in trust between individuals, Rotter developed the Interpersonal Trust Scale (ITS). Rotter pointed out that generalized expectation is based on an individual's characteristics [57].

In sociological theory, Sztompka introduces trust as *"a bet about the future contingent actions of others"*. This definition consists of two primary components: belief and commitment. The belief is that a trusted person will behave in a favourable way, and specific actions will be committed based on this belief. Overall, expectations and beliefs are the core components of trust in sociology and psychology, and they can be used to derive a trust definition for online fields [14],[20].

In computer science, trust is one of the topics that has received wide attention in diverse fields such as access control, security, distributed systems, game theory, and online social networks. In social networks which is the main focus of this work, trust is based on

relationships between users. In order to compute a trust metric in social networks, a solid definition is required. To get to this definition, we must first consider the idea of trust in the realm of computer science.

Electronic commerce (e-commerce) requires a considerable amount of trust because it involves the sale of a product or service over the Internet and not directly through a person. Trust is defined in this context as a set of expectations that lead to an action or behavioural intentions that evoke the possibility of loss due to absence of control on the trusted party [17]. In e-commerce, many researchers agree that the set of *beliefs* in this context are ability, integrity, and benevolence of the trusted party [16]. The belief in ability refers to the skills and competence of the trusted party. When consumers have doubt about the ability of an online vendor to provide them with excellent services, or they have limited understanding of the market in which the vendor works, they most likely will not be interested in dealing with that vendor. Moreover, shoppers typically do not wish to interact with a vendor if they are not sure about the vendor's integrity; they may suspect that the vendor will not abide by the rules or keep their promises. Having doubts about a vendor's benevolence by thinking that they do not care or they have bad intentions toward customers may also be a reason for consumers to not deal with that vendor.

Peer-to-peer (P2P) networks also rely on trust to provide services. P2P networks are distributed systems where peers interact with each other through a decentralized infrastructure. All peers have two roles: acting as both consumers and providers. Trust has been widely studied in peer-to-peer networks, and these studies distinguish between two types of trust in this domain. Firstly, trust is defined as a peer's belief in capability,

reliability, and honesty of other peers [50]. Secondly, the global trust value for a peer is known as a peer reputation over the entire network. Reputation is similar to trust, however, it is based on observations received from other peers in the system about that peer's past behavior [58], [39], [56].

With the ideas and definitions above, it is possible to move forward and describe online social networks. Online social networks are interactions between people over the Internet. Defining trust in such a platform must consider both human nature and online environments. In the next subsection we cover the topic of trust in social networks, its definition, and its properties.

2.1.2 Online Social Networks

In the last decade we have witnessed the phenomenon of social networking sites (SNSs) that have captured the attention of a wide range of people and become a routine part of their daily lives. There are many SNSs with different languages, goals, and purposes, yet some of these websites have become more popular than others. SNSs are web-based services that allow users to create a personal profile and share information, experiences, and opinions with a collection of friends. The first appeance of a social network site was the introduction of a site called SixDegrees in 1997. It allowed users to connect with friends and create profiles. The

launch of Friendster in 2002 represented the next wave of social sites. In early 2004, Facebook appeared on the scene as a social network site for Harvard university students. Facebook quickly expanded, starting with the inclusion of high school students, until it became open to the general public [12],[13].

The base of most SNSs is the user profile. Profiles (or home pages) often contain general information about the member such as age, gender, hometown, interests, and other information that a user would like to add to their profile and share with others. Most social network sites implement similar functionalities such as writing comments on their friends' profiles and private messaging. However, there are other features that differentiate them from each other. Granting access to a profile varies from site to site. Some social sites allow the public to access a user's profile while some only allow a user's friends who are connected to the system to view the profile. Some social network sites have photo-sharing and/or video-sharing services, some have blogging and instant messaging technology, and some of them provide mobile interactions. Furthermore, SNSs are different in their strategies and objectives. While some SNSs are essentially about networking and meeting new people, other social networks, such as Facebook, are about perpetuating relationships that are initiated offline [12],[8].

Another difference between SNSs is the action of confirming a friendship. Some SNSs require bi-directional confirmation of friendship while others do not. Friends in social network sites that require one-directional bonds are often labeled as "fans" or "followers" [8][16]. Statistics show that users in social network websites are likely to connect both with

friends that they already know as well as new people with whom they are not acquainted [32].

2.1.3 Trust in Online Social Networks

In recent years, online social networks have become very popular among the online community. Users in SNSs share their information, photos, favourite movies, books, opinions, and experiences. Trust is, therefore, an invaluable notion in such a platform where there is sharing of significant amounts of information between people. Trust helps identify users who we can communicate with, share information with, and form friendships with. Due to the absence of sociology factors in online social networks, such as a user's background information, histories of interactions between people, and relationships between users, trust requires an accurate and simple definition to effectively compute trust in social networks. With this definition, users can describe clearly what they mean by "trusting" others and can quantify the amount or the strength of this trust.

March's work [1994], which is highly cited, focuses on “Formalising Trust as a Computational Concept”[41]. March's model is one of the pioneering works in computing trust and integrates many of the trust factors from the field of sociology. However, this model is theoretical and too complex to be used in online social networks [45].

To provide a simpler definition for online social networks, J.Golbeck and J.Hendler in [20] used the main social components of trust definitions that were proposed by Sztompka and Deutsch to form an appropriate definition of trust in web-based social networks. Golbeck and Hendler adopted the ideas of commitment and belief (as were previously mentioned) to summarize the nature of the online social network environment. The definition proposed by Golbeck states that:

"Trust in a person is a commitment to an action based on a belief that the future actions of that person will lead to a good outcome."[20],[21]

Golbeck and Hendler adduce that trust in online social networks has three primary characteristics. These characteristics are transitivity, composability, and asymmetry [20]. They state that trust is not perfectly transitive as it applies in mathematics. For instance, if A trusts B, and B trusts C, that does not require A to trust C with the same amount of trust that is given by B to C. There are actually two types of transitivity of trust for online social networks: trust in a person and trust in a person's recommendation of other persons. For example, John trusts Charles's opinion about books but does not trust his recommendation of other peoples' opinions about books [21].

Although there are two types of trust, social network sites prefer to use a single value to represent both types of trust. The trust definition indicates that trust in a person is based on a belief that the trusted person's action will lead to good outcome [20]. From this definition

and the trust transitivity, if A asks B about his/her opinion about C, A will use B's opinion about C to take an action with regards to C. A trusts B and believes that B will give a recommendation that will produce a good outcome. Thus, a trusted person's opinion is used as a foundation for building trust in new people [21]. Figure 2.1 depicts a simple example of transitive trust.

As mentioned above, trust is not perfectly transitive, trust decreases along a chain of contacts. This concept is known as propagating trust through a chain of people in a social network. In the case where a person receives many recommendation about how much to trust an unknown person, the initiator of the request needs to compose the information to make a decision as to whether he/she will trust this person or not. As shown in figure 2.2, all the information coming from neighbours flows into a composition "combination" function in order to compute the trust value [21].

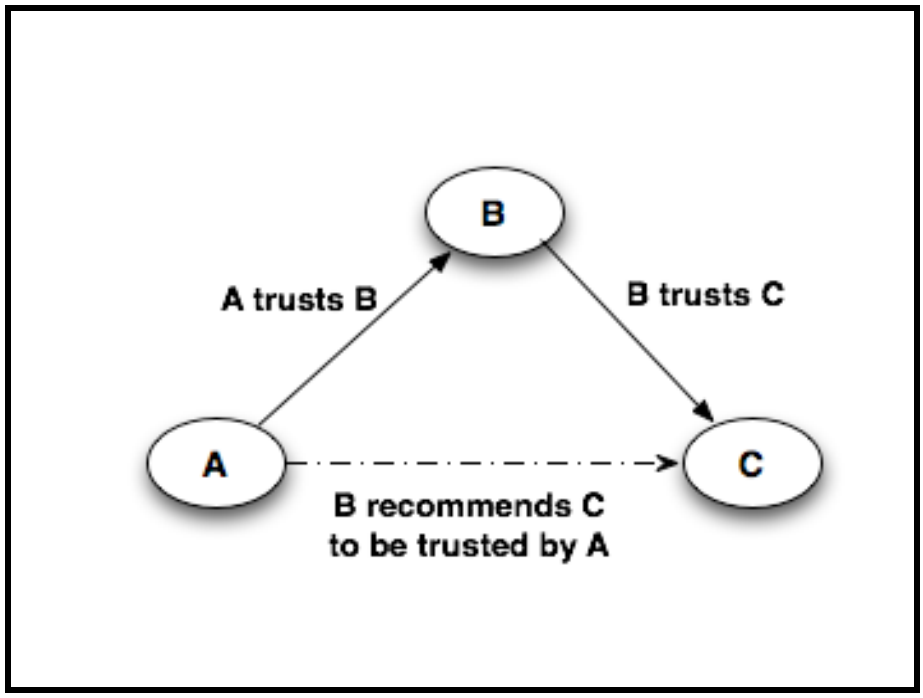

Figure 2.1: Illustrates transitive property in a simple chain between three people in the network [21].

Trust *personalization* is an important characteristic of trust in online social networks. That means trust is a personal opinion. In reality, people have varied opinions about the trustworthiness of others. If one asks several friends about the trustworthiness of a specific person, they will get very different opinions because trust is a personal opinion based on personal experiences. Thus, computation of trust should be based on an individual perspective [21].

Since people have different backgrounds, histories, and experiences, when two people are involved in a relationship, it is not necessary to have the same trust value in both directions between people. This is called the asymmetric property of trust, and it occurs in human relationships and appears without any doubt in social networks. In general, trust is reciprocal between parties but with differences in the amount of trust that each one assigns to the other. This property of trust can be seen between, for example, students and professors, employees and managers, and children and parents [21].

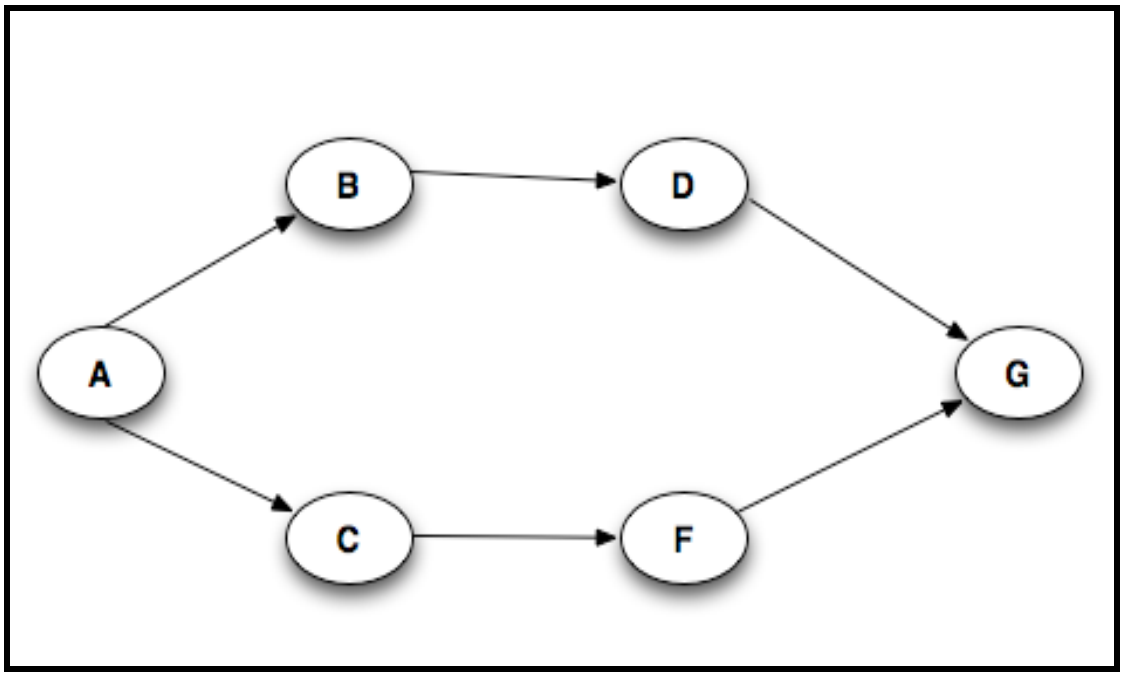

Figure 2.2: A received information about G from two paths: A-> B-> D-> G, and A-> C-> F-> G. A has to combine this information about G to form his opinion about G [21].

Social network sites differ in the way they represent trust among users. Some WBSNs have trust based on the connections between users while other websites use labels to indicate trust values like, low, moderate, and high trust. Moreover, Orkut, a social website operated by Google, represents trust ratings between users by using a count of smiley faces (from zero for no trust up to three for high trust). In addition, other social network sites use a numeric scale to identify the trust rate between users. Some of these websites use a binary scale (0,1) while others express trust explicitly using a numeric scale ranging for instance, from zero to five or zero to ten. Numeric scales are more appropriate to use for computations in social networks because they can be easily factored into mathematical calculations [21].

2.2 Advogato Trust Metric

Finding a group of trusted people while preventing attackers from accessing private information is the essential goal of designing and implementing trust metrics in an online world. Advogato is one of the well known group trust metrics on the web and was proposed by Levien [38]. Levien deployed Advogato on the Advogato.com website, which is an online community for free software developers, to test his trust metric. Advogato is designed to accept as many trusted users as possible while reducing the influence of unreliable users. The input of Advogato is a graph that represents Advogato's members. Each node in the graph represents an account, and a directed edge indicates a certificate. Advogato's group trust

metric consists of three main steps. The first step is assigning capacities to each node in the network. In the second step, the graph is converted into one with a designated node called the Supersink. Finally, it computes the maximum network flow to identify a group of trust.

Capacities are assigned to each node based on the shortest-path distance from the source to a node. The shortest-path is computed by applying a *breadth-first* search algorithm. The capacity of the seed node is equal to the number of trust nodes in the graph. The capacity of each successive level is equal to the previous level capacity divided by the average outdegree. Basically, the capacities are integer numbers and greater than or equal to one.

The graph should transform into a structure that meets the standard network flow algorithm (Ford-Fulkerson) specifications. Thus, the graph is converted into a single seed and a single sink with weight attached to edges rather than nodes. The graph conversion procedure is as follows: every node v in the graph splits into two nodes v- and v+. A link is added from v- to v+ with capacity C(v)-1 and an edge from v- to the supersink which has a capacity of one unit. Additionally, all inedges of v in the original graph become inedges of v- and all outedges from v to node x are represented as links from v+ to x- with infinite capacities. The transformation process is illustrated in figure 2.3 [36],[38],[61],[62].

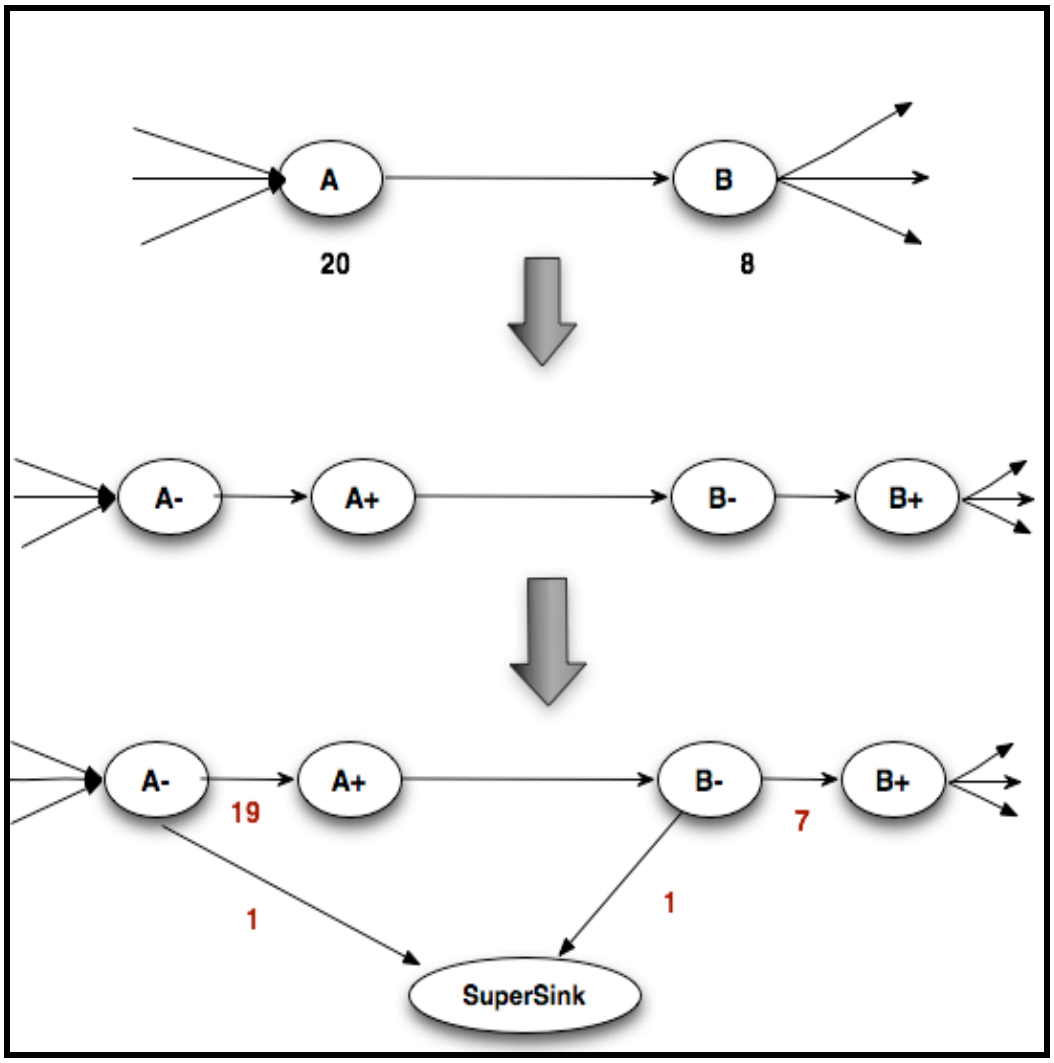

Figure 2.3: The procedure of transforming a graph's nodes into Advogato structure [36].

After the transformation, computation of a maximum network flow from the seed node to the supersink is applied by using standard algorithms such as Ford-Fulkerson algorithm. Nodes that are accepted by Advogato are the nodes that have flow across their corresponding edges to the supersink. Also, there is an additional constraint state that for all nodes in the graph, if there is flow from v- to the supersink, there should be flow from v- to v+. [36][61].

2.3 Related Work

Identifying trusted people in a network and preventing attackers from accessing private information or damaging the network has been studied by a number of researchers in a variety of networking domains. Trust algorithms resulting from these studies can be classified into two types: global and local. In the global approach, for each user in the network the algorithm calculates an overall trust value for the entire network. This is known as the *reputation* of a user. Alternatively, computing a personalized trust value is the goal of local algorithms. Local algorithms compute trust values based on the perspective of the requester [21].

Levien and Aiken [37] presented the Advogato trust metric. It builds a structure that accepts all possible good nodes, thereby reducing the influence of malicious users. Advogato computes global trust values by building a trust metric through the network. The trust metric was designed to have one seed node and one designated supersink to apply a maximum network flow algorithm for finding connections between good nodes and cutting off unreliable parts of the network. This concept has been proven by running the metric on the Advogato online community which has three levels of trust among its members. Based on users' levels, it assigns authority that determines the users' permissions for posting or modifying the website content.

Ziegler and Lausen [62] proposed a local trust metric called *Appleseed*. Appleseed was inspired by a spreading activation model that propagates energy through the graph. In Appleseed, propagating energy from a node to successors is based on the link weight. Both Advogato and Appleseed are local group trust metrics. Advogato applies maximum flow to assign capacities depending on the depth of the path, while Appleseed adopts spreading activation models.

Trust and reputation have received significant effort from researchers in P2P systems. In [30], the EigenTrust Algorithm calculates unique global reputation values for each peer in the system based on a peer's previous behavior. This can be done by normalizing the local trust values to compute the left principal egienvector. Gong et al. [23] provide a searching algorithm for a resource in an unstructured P2P system. It is fully based on social networks and makes use of trust relationships between peers. Each node in the network has to build its knowledge index. This knowledge index includes three components: a friend list which contains a list of peer friends who respond to a query associated with the topic of the request; a list of malicious nodes that did not respond to the query; and a record of the trust values of the friends in the trust list. Nodes in the network use the knowledge index to find a trusted node to transfer a query message between nodes while excluding malicious nodes.

Generally, trust in peer-to-peer networks is based on the capability and reliability of a peer to provide a service. With regards to P2P systems, a node is either trusted or not as to whether it will perform a task. This is fundamentally different than "from" trust in online social networks. In online social networks, people have different opinions about a topic or a person, and this variety of opinions is related to the differences in peoples' personalities. Because of this, one person can decide how much to trust another person depending on his/her personal perspective. Furthermore, computing a global trust value in P2P networks is valuable because each peer in the system will expect to have the same responses as all the others do. For this reason, personalized trust in P2P systems does not have the same importance as in social networks [21].

Related to mobile social networks where individuals communicate with each other using mobile phones, [55] presented a searching algorithm for finding an expert or suitable person in a specific field and the matching trust path. The calculation of integrated weight is based on three factors: trust value, credibility, and the number of intermediate nodes between two individuals according to the provided formula. This work considers a person's specialty as the main factor when integrating the weight of nodes and selecting the most suitable person among a group. In Recommender systems, Jamali and Ester [29] introduce the TrustWalker algorithm. It is a random walk method that combines trust-based and collaborative filtering approachs in order to recommend items. The TrustWalker method considers the rating of items similar to the exact item to avoid further searching that provides less reliable ratings. The main idea of this algorithm is to search the trust network using random walk to find a

user who rates the exact item i or a similar item j. The random walk runs several times and each time it returns different rating values. By aggregating all these values, TrustWalker returns the predicted rating for item i.

The context of social networks (SNs) has been the subject of a wide range of studies for inferring trust between users, searching for experts, or recommending friends. With regards to searching in SNs, Hangal et al. [24] studied a search algorithm that considers various social asymmetric relationships and their strengths in social networks. By representing relationships as weights on a directed graph, an adapted Djikstra's algorithm was applied to find the strongest path between nodes. In [27] the proposed model is designed to find friends in an online social network who share common interests. This model consists of three components: a trust engine, a popularity engine, and a rank engine. To compute trust values, they used a modified Dijkstra's algorithm along with the PageRank algorithm to calculate the popularity of a user in relation to a certain keyword. By combining the values inferred by the trust and popularity engines, the ranking engine produced the final ranked results of the search.

Hong and Shen [26] constructed a trusted sub-network from online social networks based on transitive relationships to control access permissions of connected users. The main idea of this work was to assign permission values to direct and indirect contacts based on their transitive relationships and owner preferences, thereby rendering owner data to them as

accessible or not. They consider context when propagating trust into the same group of people.

The main idea of the RN-Trust algorithm, which was proposed in [48] by Taherian, M. et al, is to infer trust by using the concept of a resistive circuit. Nodes in the trust network are mapped to nodes in the resistive network. Resistors are placed between connected nodes to represent the relationships between nodes, and a diode is placed between these nodes to satisfy the asymmetric characteristic of social relationships. To compute the trust value from the seed node to the target node, RN-Trust calculates the equivalent resistance of this circuit.

Furthermore, Jennifer Golbeck has proposed a variety of algorithms to infer and compute trust in social networks. For instance, TrustMail is an application that uses reputation analysis which was introduced by Golbeck in [19]. TrustMail is an email client that rates email messages based on the sender's reputation. The trust level of the email sender can be a general level of trust or associated to certain topics.

In addition to TrustMail, Kuter and Golbeck in [34] provided a trust algorithm to infer trust using probabilistic interpretation to estimate confidence in social networks. This algorithm is named SUNNY. SUNNY generates a Bayesian Network that corresponds to the trust network. It performs probabilistic sampling techniques to estimate the confidence values. Based on confidence values, SUNNY computes the trust values for nodes.

Moreover, Katz and Golbeck in [31] proposed TidalTrust which is an algorithm for inferring trust in social networks using numeric trust values. TidalTrust takes into consideration that the shortest path from the source to the sink gives an accurate result. In addition, it assumes that the accurate information can be retrieved from the highest trust adjacent nodes. This algorithm is based on the Breadth-First Search to find the shortest path. Unlike TidalTrust, which is highly cited, the algorithm proposed in this thesis is based on the strongest relationships between users rather than using the shortest path. In reality, the shortest path does not always lead to the most trusted person or give accurate results. Let us consider this situation:

If John asks about a good programmer, he will forward messages to people connected directly to him. People connected to John have different relationships to him of varying strength. Most likely, people who have strong ties with John will recommend a programmer to John or ask their friends to recommend one. In the reverse path, John will more likely trust the opinions of people that he knows rather than those he does not. Rather than relying on the shortest path, this depends on the strength of the relationships and the fact that people trust the opinion of people they know in their lives rather than unfamiliar ones.

Chapter 3 Identifying People of Trust

3.1 Overview

In online social networks, members communicate and share their personal information, opinions, experiences, diaries, and photos. People are likely to share this information with like-minded people whom they trust. However, with the prosperity of these services, online social networks are often misused by malicious users and spammers in ways that violate other users' privacy. Hence, identifying trusted users is a major concern in online social networks.

Identifying reliable users involves two variations of algorithm: global and local computation algorithms. Global algorithms are designed to compute a universal trust value for each user in the network regardless of the requester's perspective. On the other hand, local algorithms calculate trust values to find trustworthy users within the network and are dependant on the requester's perspective [21].

In order to discover trustworthy people in SNSs, we use the local computation algorithm approach since there are various reasons for connecting people. The type and strength of connection among users in online social networks allow us to express relationships in SNSs in a more specific way than just labeling people as "friends". Accordingly, our approach for identifying reliable users is based on a given user's perspective and uses the strength of that user's relationships to compute a trust value.

Overall, this chapter describes the Capacity-first maximum flow for determining a local group of trustworthy people based on a given user's perspective. The goal is to protect the user's information from unauthenticated people. To do so, we adapt the Advogato trust metric by incorporating social relationships and then propagate capacities along a chain of connected users. By applying Capacity-first maximum flow, we determine the group of trust according to trust values obtained from relationships.

3.2 Problem Formalization

An online social network is usually modeled by a graph structure. Let $G = (V, E)$ denote a directed, acyclic, weighted graph where V is a set of *vertices* which correspond to users and E is a set of ordered two-element subsets of V: $E \subseteq V \times V$, called *edges* that point out directed relationships between participants. The weight on the edge between pairs of vertices (individuals) is a measurement of the strength of the relationship. For a given node $u \in V$, let $\mathcal{G}(u) = (v_u, \varepsilon_u)$ signify a subgraph of G where $v_u \subseteq V$ is the set of nodes that can be reachable from u along any edge $e \in E$ and $\varepsilon_u \subseteq E$ is the set of edges that connect pairs of nodes in v_u. For a given node u, we denote by $I(u)$ and $O(u)$ the set of in-neighbors and out-neighbors of u, respectively. In the graph $\mathcal{G}(u)$, a path P from u to a certain node $v \in v_u$ can be represented as a sequence of nodes, $uk_1 \ldots k_n v$ such that $\forall n$: $k_n \in v_u$. We call such nodes k_n *internal nodes*. For the path P, if there are no internal nodes, i.e. the node v is directly

linked to node u, then we call this connection a 1-hop connection. Otherwise, we call it an (n+1)-hop connection according to the number of internal nodes n. From the abovementioned notations, we formalize the problem of identifying a set of trustworthy people as follows: Given a personal graph $\mathcal{G}(u) \subseteq G$, identify an ordered set of users $\mathcal{T}'_u$ who are likely to be trusted by user u such that $|\mathcal{T}'_u| \leq N$ and $\forall\ v \in \mathcal{T}'_u : v \notin O(u)$.

3.3 Adapted Advogato Algorithm

The adapted Advogato algorithm contains the essential steps for computing the trustworthy group. Firstly, we assign an initial capacity to a given user; this is used to propagate capacity to the personal network. Then we propagate capacities among connected users based on the strength of their ties with the given user. Finally, we identify the trust group of people by repeatedly applying the Capacity-first maximum algorithm.

3.3.1 Assigning a Capacity to a Seed Node

To identify reliable people in a social network, we have to first assign an initial capacity to an individual user who is used as the seed node. Initial capacity is dependant on the number of directly connected nodes to the seed. The capacity of a seed node $S \in V$ is precisely computed by equation 3.1.

$$C(S) = 2^m \times \left|O(S)\right| \tag{3.1}$$

Where m indicates the controlling parameter that controls the size of the trusted group. The parameter m has a value that is greater than or equal to zero, $m \geq 0$, and $O(s)$ represents the set of out-neighbour nodes that have direct links from the seed. Contingent upon the information sensitivity, we can limit the number of trusted people who can access this information. Moreover, depending on outgoing edges of the seed node, we can allocate the controlling size parameter m. Figure 3.1 demonstrates an example that shows the initial capacity computation for the seed node *S*. The seed node *S* has four out-neighbors *A, B, C, D*, and the controlling size parameter m is set to take an integer value between zero and six based on the preferences of the user for the purpose of identifying the trustworthy group.

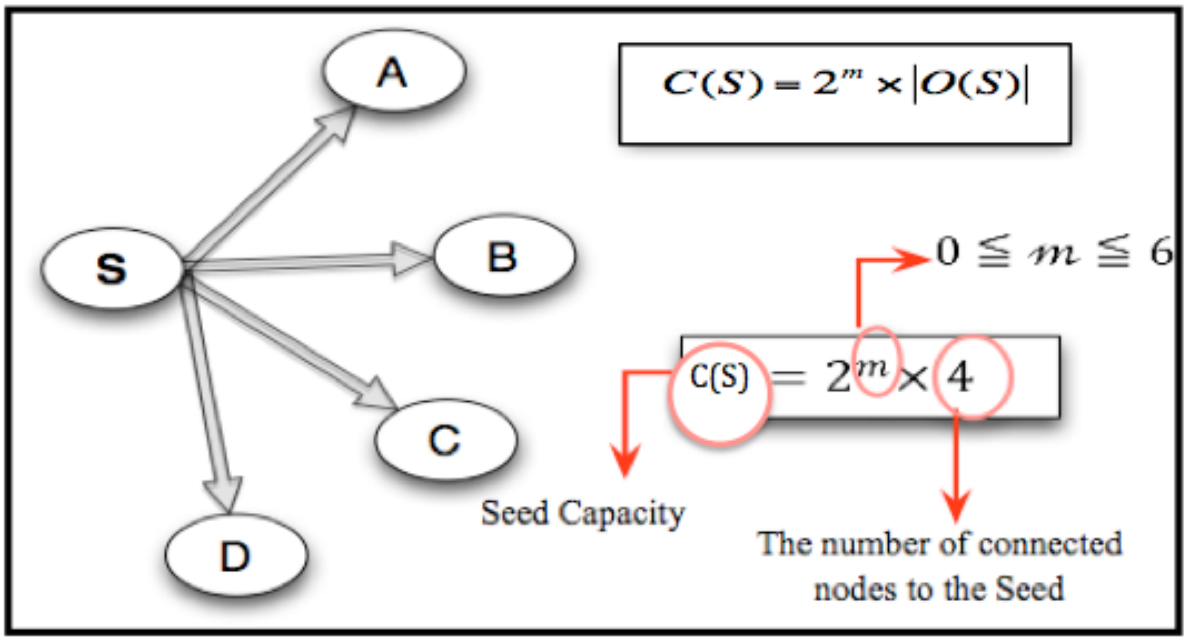

Figure 3.1: Assigning initial capacity for a given user.

3.3.2 Building a Weighted Personal Network

Since local trust computation algorithms identify trustworthy people in a network based on the seed user's perspective, constructing a weighted personal network (WPN) for the seed is a fundamental step. A personal network (PN) depicts a sub-network of the original social network. It represents the neighbour nodes' relationships with the seed node. In online social networks there are wide range of social relationships that connect people. Even though two people in a social network have common neighbors, the relations that connect them are distinct. Figure 3.2 features a simple social network. Consider users A and B. They have common neighbors F and D, yet the social relations that connect users A and B to F and D

are variant. While a family relationship connects A and D, B and D are just colleagues. Moreover, F is a member of B's family and a friend of A. This example shows how F and D are common neighbors to A and B, but they have different types of relationships that lead to different weight values. The PNs of A and B are depicted in figures 3.3 and 3.4 respectively.

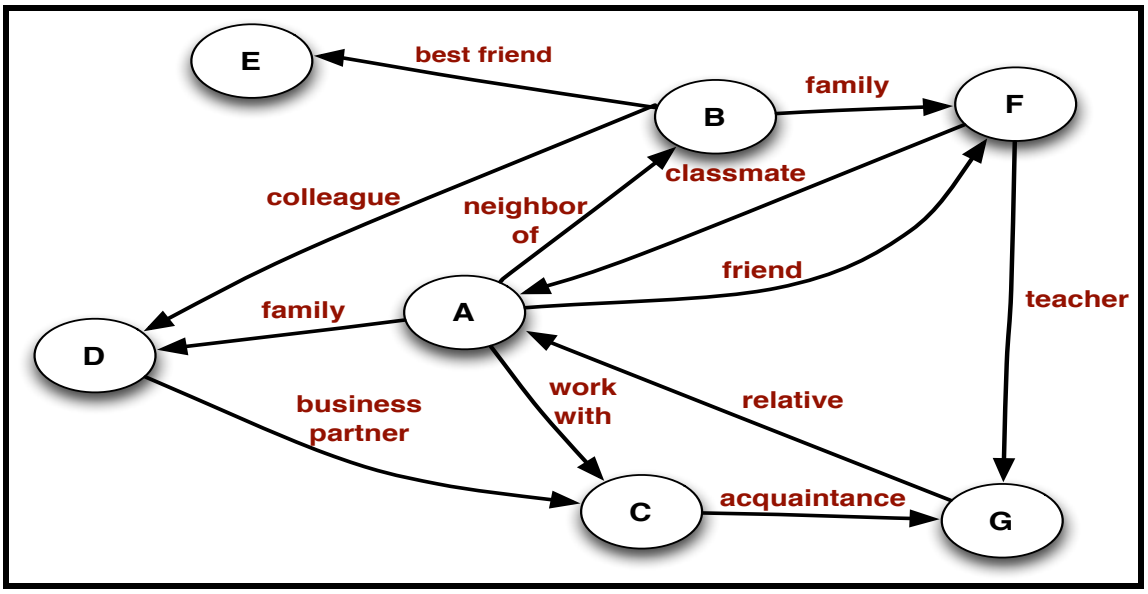

Figure 3.2: Original network that shows relationships among people in a simple social network

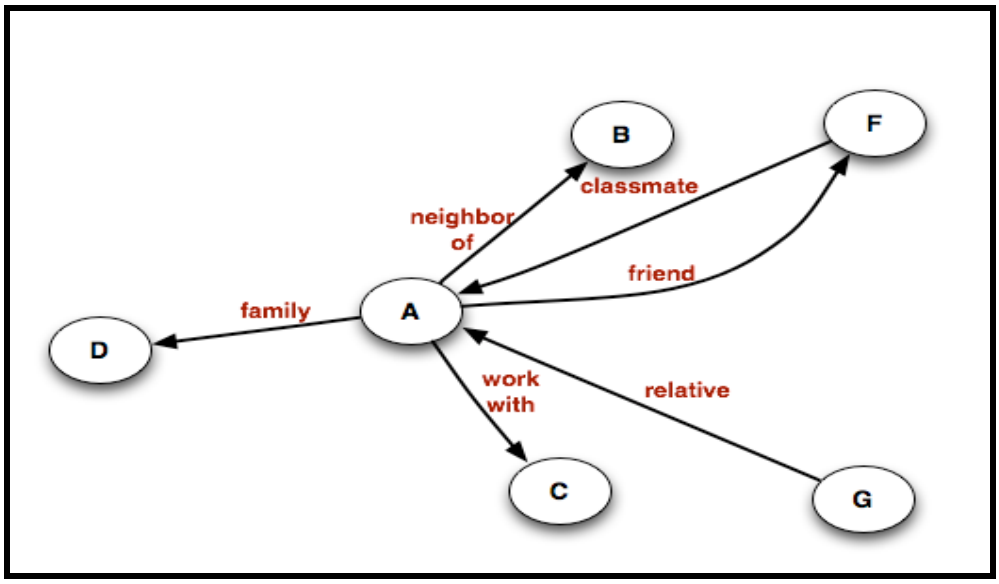

Figure 3.3: Personal network of user A that shows the relationships.

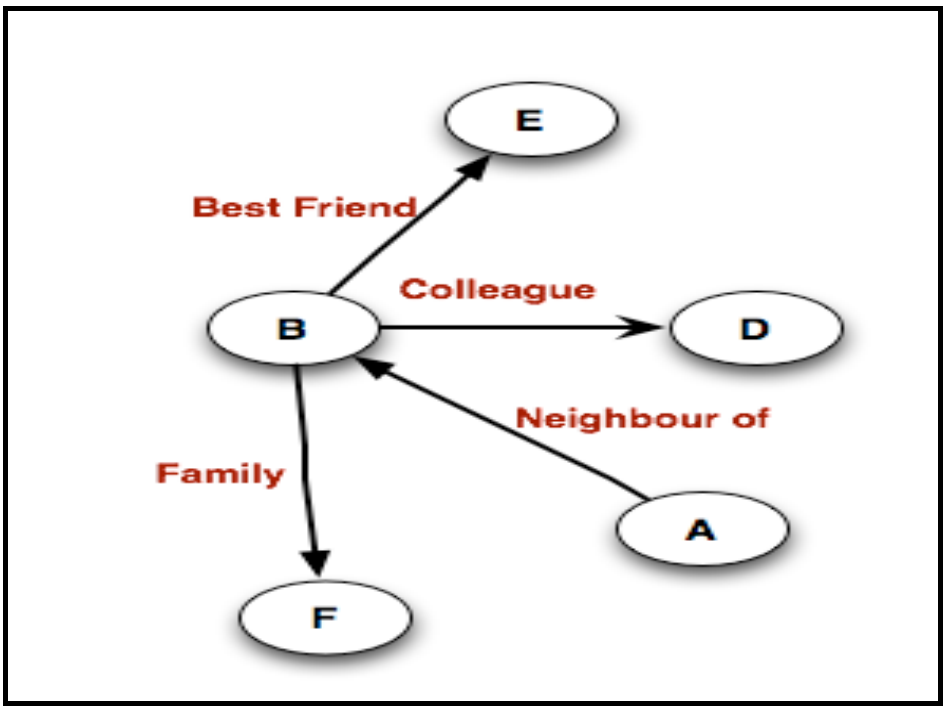

Figure 3.4: Personal network of user B that shows the relationships.

Figures 3.3 and 3.4 illustrate the differences between personal networks for users A and B and are constructed from the same original social network. In order to identify trusted people, we need to build the weighted personal network. A WPN is a weighted, directed graph without self-loops that is represented by the weighted adjacency matrix *W*. We build a WPN from a PN by considering all outgoing edges from the seed node to its neighbours and assign weights to the edges that represent the strengths of the connections. Computing the weight between nodes depends on the application. There are a variety of techniques that can be used to compute the weight based on the goal and domain of the application. In some domains we are able to infer more explicit relationships between people, such as in Facebook. We can assign weight based on types of relationships where higher values associated to edges represent strong relations and weak ties are assigned lower values. Using figure 3.4 as an example, we could assign the maximum weight (e.g. 1) on the edge from B to E since the relationship is "best friend". In addition, we can assign less weight (e.g. 0.48) on the edge from A to B since the relationship is just "Neighbour" or "Acquaintance". How we assign the value to edges will depend on application-specific factors.

Some other ways of assigning weights include measuring the importance of a node based on out-degrees or in-degrees or measuring the similiarities between two users. In our case, the weight on the edge that connects two vertices v and u $\in O($v) is obtained by using the normalized Jaccard coefficient in equation (3.2):

$$W_{vu} = \frac{1}{\max_{k \in O(v)} w_{vk}} \times \frac{\left|O(v) \bigcap O(u)\right|}{\left|O(v) \bigcup O(u)\right|} \tag{3.2}$$

such that $0 < w_{vu} \leq 1$. If $w_{vu} = 0$, there is no interaction between user v's out-neighbours O(v) and user u's out-neighbours O(u), so we set w_{vu} to the lowest value among w_{vk} where $k \in O$(v). As mentioned before, W represents a weighted matrix that has entry w_{vu} which denotes the weight from node v to node u.

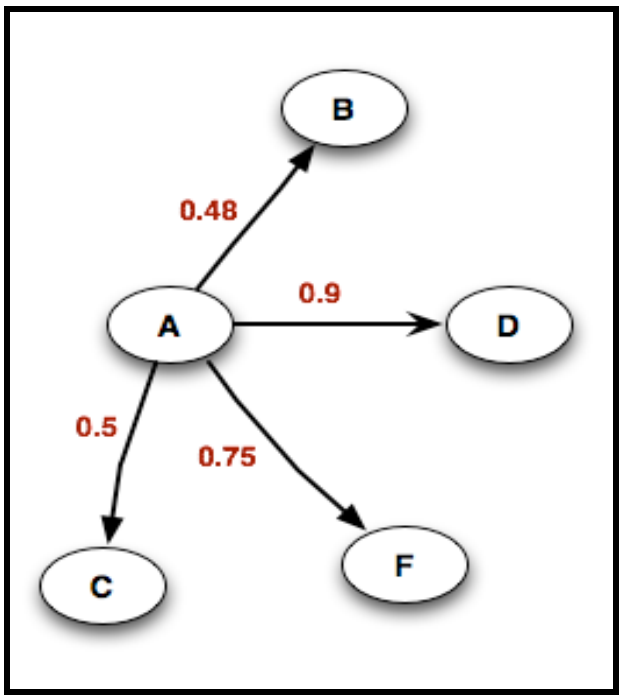

Figure 3.5: WPN of user A.

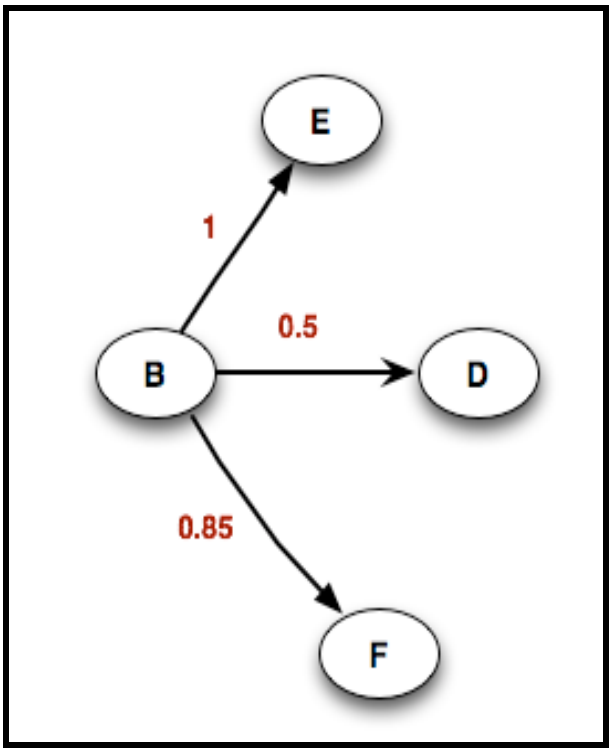

Figure 3.6: WPN of user B.

3.3.3 Propagating the Capacity through the Personal Network

Once we have assigned the capacity and built the personal network for the given seed node, we recursively diffuse the seed's capacity along paths from the seed to its out-neighbour nodes. Disseminating the capacity into successive nodes involves two factors: the strength of the relationship and the distance between the seed and its out-neighbours. The more reliable a node is rated by in-neighbours, the more the capacity spreads to that node, and the closer a node is to the seed, the more the capacity disseminates to that node. Considering the distance in addition to the strength of the tie is due to long chains of connection signifying weak communication.

More formally, for a given node *u* that can be directly or indirectly reached by the seed *s* within l hop, the capacity of *u* is computed as shown in formula 3.3:

$$C_{(u)} = \arg\max_{v \in I_s(u)} \left(d \times W_{vu} \times C_{(v)} \right) \tag{3.3}$$

where $I_s(u) \subseteq \mathcal{V}_s$ is the set of node u's in-neighbours that can be reachable from the seed *s*. $d \in (0,1)$ is a decay factor, and w_{vu} is a weight on the edge between v and u. As an example, consider user A in figure 3.7. A has a trust value equal to one, representing a strong relationship and powerful connection with the seed (S). To assign a capacity to A, we assign the decay factor an initial value of d=0.5 and multiply the weight of the connection between

S and A by the value of *d* and by the capacity of the seed node. We use the seed node capacity here because A is directly connected to S, but if A is located 2 hops from the seed, then we will use the intermediate node capacity that connects the seed node with A which is in 1-hop from *S* (previous level).

In general, the path from s to u can be represented by a sequence of nodes (users). For instance, $s \rightarrow v1 \rightarrow v2 \rightarrow v3 \rightarrow v4 \rightarrow v5 \rightarrow u$ is a sequence that has a maximum number of hops from s to u set to 6.

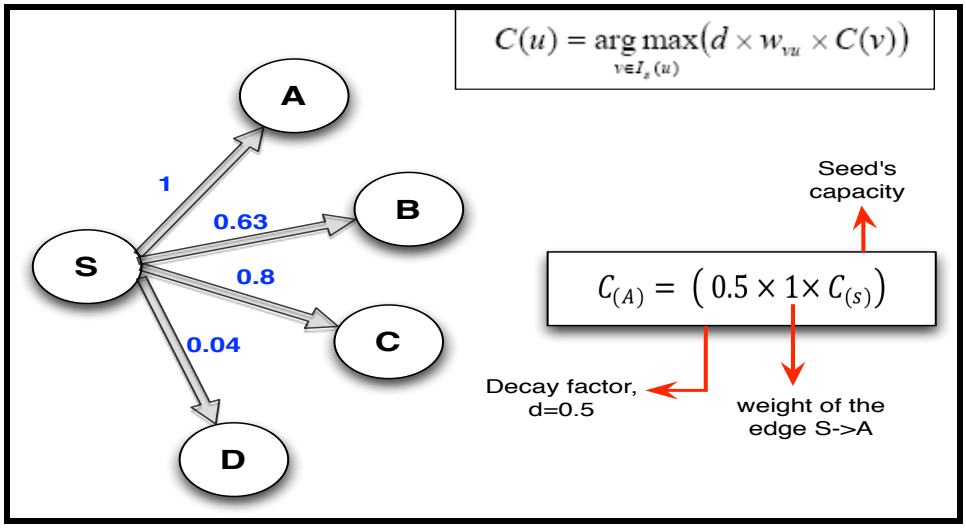

Figure 3.7: Propagating capacity through WPN.

In the proposed algorithm, the power of the relationships generates differences in dissminating capacity among nodes, even if the nodes have the same number of hops from the seed node. The strength of the relationship leads to more capacity assigned to the out-neighbour nodes. Thus, in cases of conflict between the strengths of the relationships between nodes, we consider the closest relationship. For instance, consider the situation in

figure 3.8 where node F is 2 hops from the seed and is a common neighbour to nodes A and B in 1 hop. Node F has a low weight value indicating a weak connection with B, but it has a strong relationship with A illustrated by the high weight value labeled on the edge between A and F. In such a case we consider the more powerful connection (the highest weight value) to compute F's capacity.

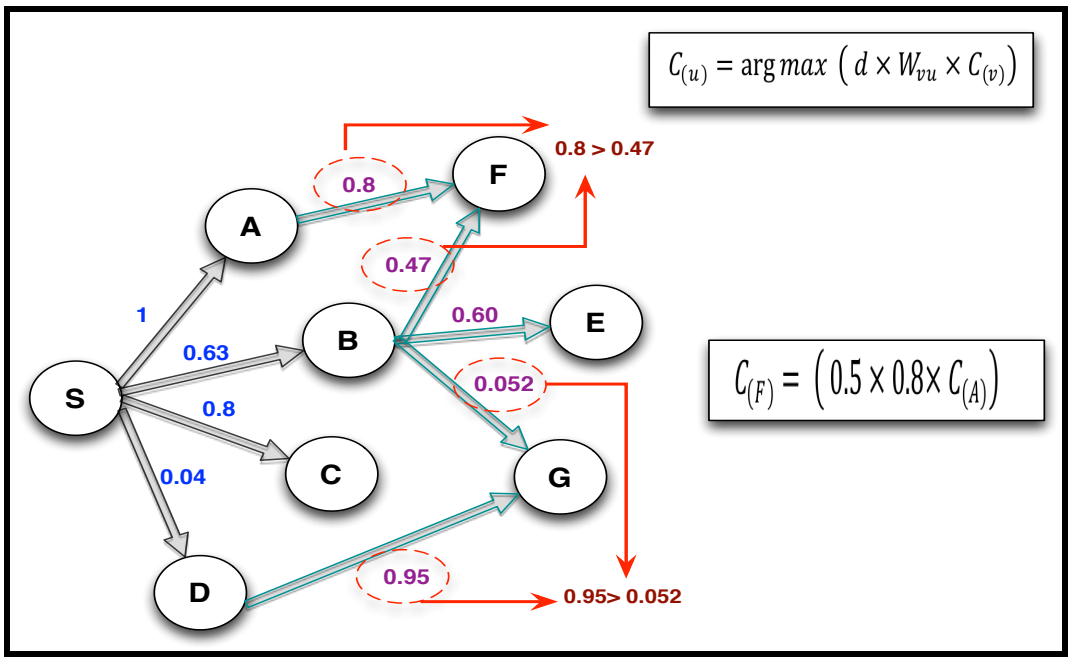

Figure 3.8: Selecting the closest relationships in case of conflict.

3.3.4 Capacity-first Maximum Flow

After assigning the node capacities along the network paths, the WPN has to be converted into the Advogato trust metric structure. The Advogato structure has a unique virtual supersink node δ. As mentioned before, an Advogato node v is split into positive and negative nodes v+ and v-. The capacity of node v is reduced by one and attached to the edge between v+ and v-. The reduced unit is assigned as the capacity for the link between v- and the supersink δ. The link that connects node v to node u is represented as a link with infinite capacity from v+ to u-. Algorithm 3.1 demonstrates the pesudocode for transforming the weighted personal graph into the Advogato structure [61].

Function transform (G=(A, E, C_A)){

Set $E' \leftarrow \varnothing, A' \leftarrow \varnothing$;

For all $x \in A$ **do**

Add node x^+ to A';

Add node x^- to A';

If $C_A(x) \geq 1$ **then**

Add edge (x^-, x^+) to E';

Set $C_{E'}(x^-, x^+) \leftarrow C_A(x) - 1$;

For all $(x, y) \in E$ **do**

Add edge (x^+, y^-) to E'

Set $C_{E'}(x^+, y^-) \leftarrow \infty$;

End do

Add edge (x^-, supersink) to E';

Set $C_{E'}(x^-$, supersink$) \leftarrow 1$;

End if

End do

Return $G' = (A', E', C_{E'})$;

}

Algorithm 3.1: Convert the original graph into the Advogato structure based on [61].

To identify the trustworthy users, we designed a maximum capacity-first search algorithm. In our method we identify a person's reliability according to the strength of her/his relationships by applying the maximum capacity-first search algorithm instead of using the breadth-first search algorithm. This algorithm has been named the Capacity-first maximum flow. Basically, the Capacity-first maximum flow algorithm computes the network flow according to node capacities; the higher a node's capacity is, the higher trust value it has.

The Capacity-first maximum flow is designed to work in the following manner: Firstly, it compares the capacities of nodes that have direct connections from the seed s. These nodes are within one hop from the seed. Then it selects the node with maximum capacity among all nodes within one hop and marks this node as the highest trusted node among all nodes in the network. This means that the selected node has the strongest relationship with the seed node. The algorithm then considers the selected node's out-neighbours to be compared with the remaining nodes in the first level (the seed's out-neighbours). Each time the algorithm adds a new visited node, it recursively examines its capacity and adds its out-neighbour nodes. In each iteration of the algorithm, when the algorithm finds a path from the seed to the supersink, it subtracts a unit from each node along the path starting from the seed and ending with the node that is connected directly to the supersink. To do this, nodes should have capacity greater than or equal to one unit. The algorithm recursively applies the above steps until one of the termination conditions occur. The Capacity-first maximum flow algorithm terminates when there is no new augmented path from the seed to the super sink or when the seed node exceeds its capacity. Whenever an augmented path is found, nodes in the path are added to a list of s' in trust group $\mathcal{T}_s$ if the node is not an out-neighbour of s. Note that we

rank trustworthy nodes in the order the added nodes. Algorithm 3.2 demonstrates the pesudocode of Capacity-first maximum flow algorithm.

Figure 3.9 shows the WPN for the seed node s. The edges between s and its out-neighbours are labeled by a weight value to indicate the power of the connections. In figure 3.10, we compute the intial capacity to the seed and propagate this capacity through the WPN according to the strength of the connections between nodes. From this figure, we can see that nodes in the same level have different capacities, unlike Advogato which assigns the same capacity for all nodes in the same level because it does not consider weight values. Then we transfer the WPN for s into Advogato structure as illustrated in figure 3.11. Finally, we apply Capacity-first maximum flow to identify the trustworthy people and rank them according to the reliability of their relationships.

```
Input: weighted directed graph G′, Seed node S
Output: Trusted Group TG
Function Max-Capacity Search (G′,S)
{

     Selected Node =Comparing nodes Capacity(S);

               Add selected node neighbors ->  candidate [ ];
                selected node capacity= -1;
                Mark selected node as visited;
do
    Selected Node= Compare Candidate nodes Capacity ();
     Path tracing (selected node, S);
     Mark selected node as visited;
     Capacity path[nodes]= -1;

End do when
     S'capacity ==0 ||  new path== null|| intermediate nodes' capacity
}

Comparison procedure:
{
   Selected node= Comparing Capacity();
   For all unvisited candidate nodes:
    If
      node_i ' Capacity == node_j ' Capacity
     Compare level -> select the one with shorter distance.
   Else
     Select the node with maximum capacity;

  Return selected node;
}
```

Algorithm 3.2: the proposed Capacity-first maximum flow algorithm.

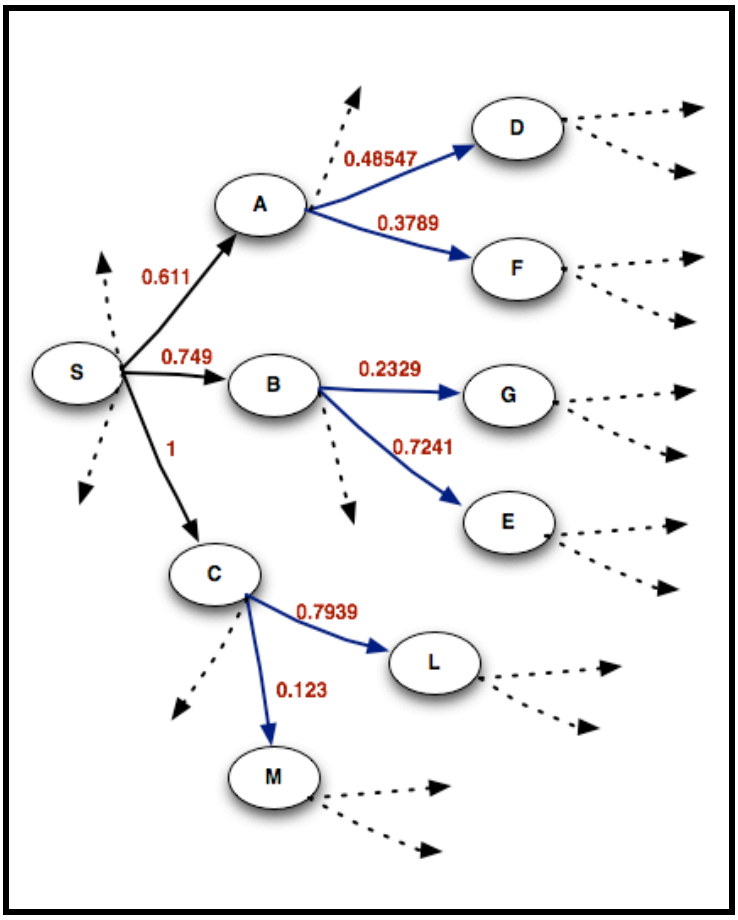

Figure 3.9: The WPN for seed *S*.

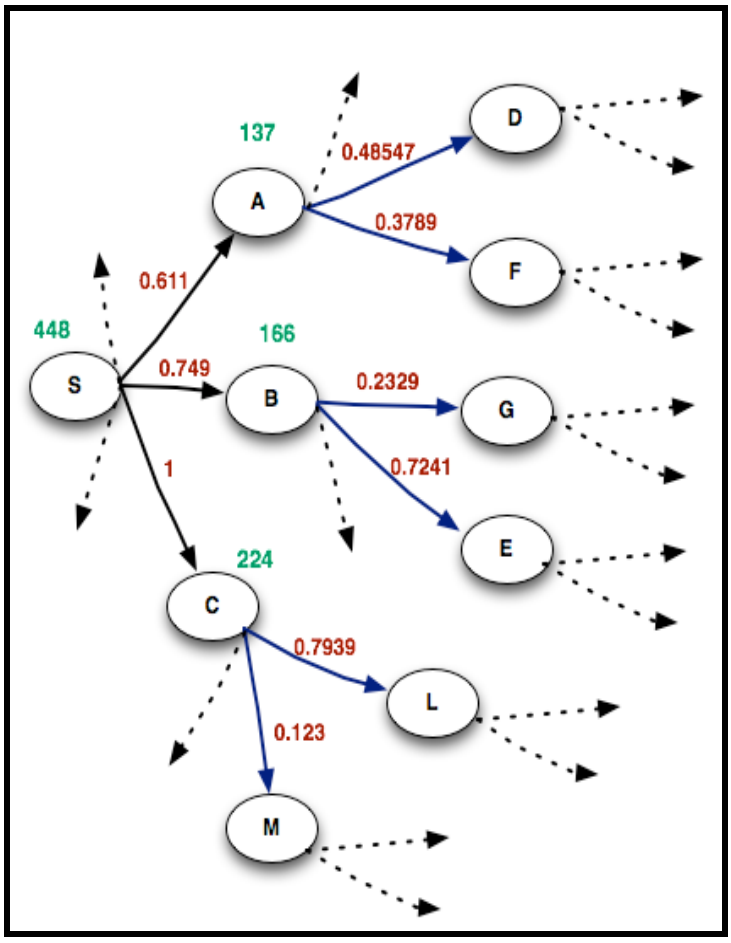

Figure 3.10: Initial capacity computation and it's propagation among nodes in the WPN.

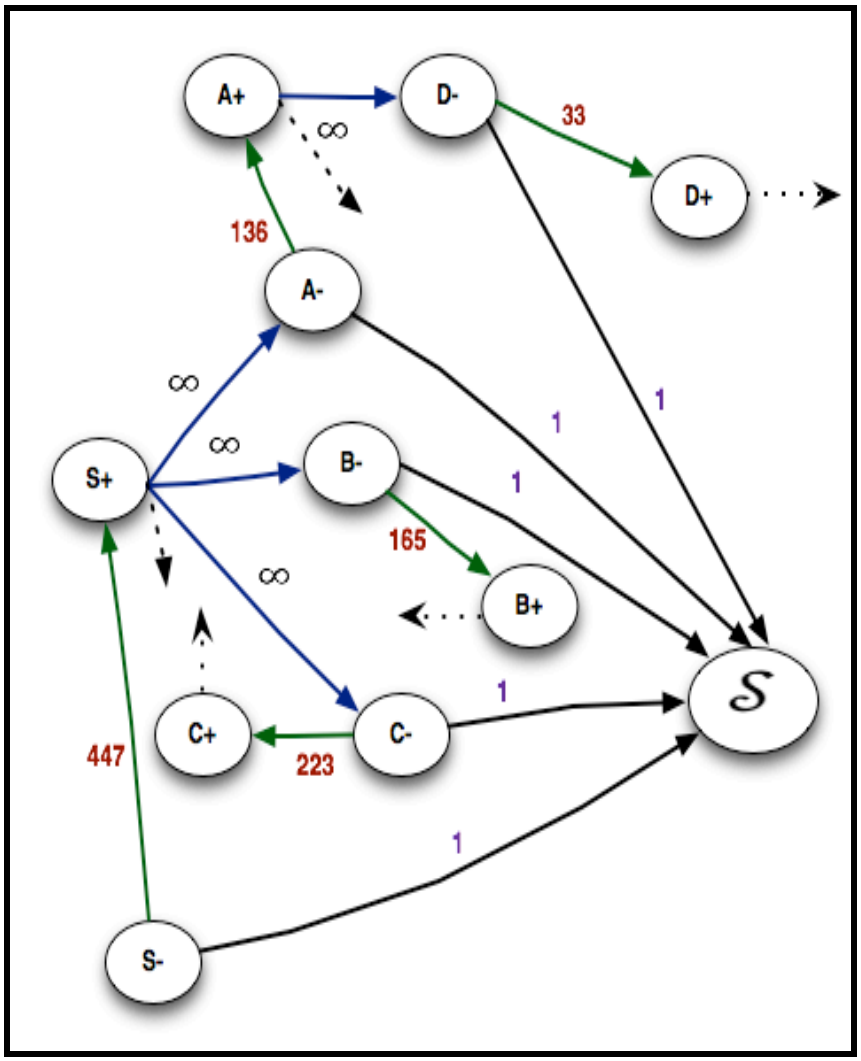

Figure 3.11: conversion the WPN into Advogato structure.

Chapter 4 Application Scenarios

This chapter describes application scenarios where one can apply the Capacity-first maximum flow algorithm in the context of online social network services. We classify the application scenarios for social network services into controlling access permissions, which is discussed in section 4.2, and link prediction for recommending new friends, which is briefly discussed in the final section of this chapter.

4.1 System Overview

Our "trustworthy group" identifying algorithm can be used, as illustrated in figure 4.1, to control access to personal information in an online social network and to help members in finding new friends in this network. The suggested Access Permissions and Friend Recommender system consists of the following modules:

- User interface
- Algorithm for identifying reliable users

The system user interface consists of three parts:

- User login information that allows the system to interact with an online social network in which that user participates.
- System options that appear to the user; a user can select one of them. The first option is for privacy settings that allow the user to set privacy permissions for his/her contacts. The second option is to recommend new friends by suggesting reliable friends to the user.
- After selecting the user operation from the system options, the user interface displays the results based on the selected options. The result could be a list of recommended new friends or a list of privacy permissions for his/her trusted contacts.

The algorithm involves:

- Construction of a weighted personal network (WPN) through interaction with an online social network to extract information about the user's personal network (PN). Construction of the WPN using the PN and assigning weight to the relationships between users based on the user's classification of the strength of their relationships.
- Propagation of capacities within the WPN. The main goal of this component is to assign an initial capacity and propagate this capacity through the WPN.
- Computation of a Trust group by applying the Capacity-first maximum flow algorithm that is proposed in this thesis. The result of this algorithm is a ranked list of trusted users.

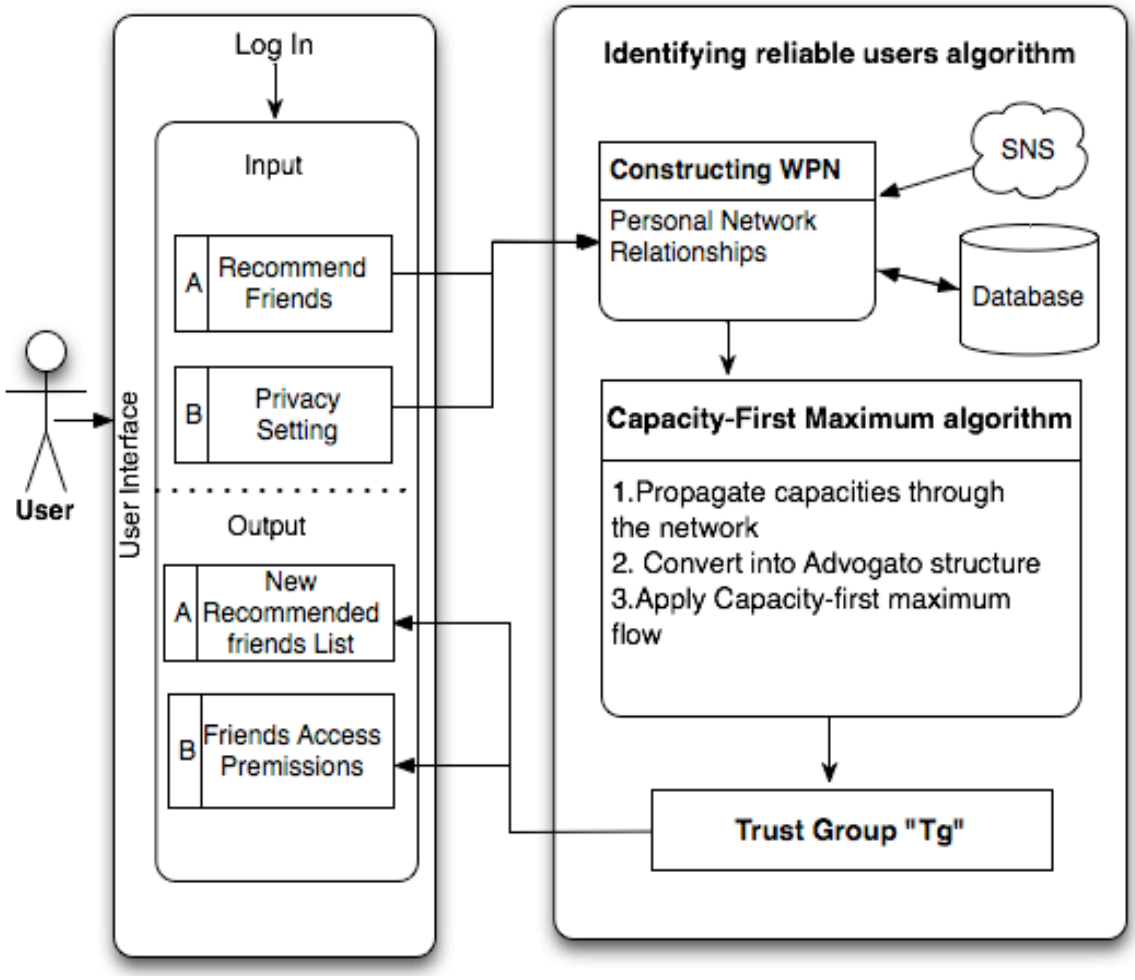

Figure 4.1: Access Permissions and Friend Recommender system architecture.

This system has two main goals: to manage the access of a user's information and to recommend new friends in an online social network based on the strength of the relationships. In next subsections, we explain in more detail how the system uses Capacity-first maximum flow algorithm in such a scenario, starting with controlling access and ending with recommending friends.

4.2 Controlling Access Permission in Social Network Services

The information posted in online social networking sites by participants needs to be protected from unauthorized users. Moreover, account owners in WBSNs would like to restrict the sharing of all their information and make it inaccessible to certain users in their friends list. Setting permission values for each and every post one puts on the social site for every user in one's contact list is not practical for the account's owner. Taking advantage of explicit relationships among users in the social network makes it possible for account owners to control the shared data effectively. By ranking friends based on how strong their relationship is, an account owner can easily control the bounds of sharing data with friends. When it comes to protecting a user's private information in online social networks and assigning permissions to contacts based on their relationships, we have proposed some application scenarios that show how the Capacity-first maximum flow algorithm can be used to do so.

Suppose there is an online social network participant named John who wants to control his shared information. In order to extract information from a social network where John posts his information and interacts with people, he has to log in to the system via a user interface. After he is logged in to the system, he selects his privacy setting options. We assume that we can extract John's information either by using an API from a service provider for a social network site such as Facebook or by crawling the network site. By selecting privacy settings,

the system derives the personal network from the online social network for which the user wants to set his/her privacy access. The system should be able to access the whole personal network that contains friends and friends of friends for up to 5-hops from the original user. The personal network or personal graph contains all directly or indirectly connected nodes to the user (seed node) and points out their relationships. Based on the user's classification of the relationships with these users, a weight is assigned to each explicit tie in the graph, and the system builds the weighted personal network. In John's case, we use the normalized Jaccard coefficient method to measure the power of the connections among users in John's PN. Notice that users can classify their relationship strengths based on data categories or any other methods that help them control their online profiles. All relationship classifications and the user's data about his/her contacts are stored in the system database. By assigning a weight to each edge that connects a pair of nodes, the weighted personal network is constructed.

An Advogato-adaptive component receives the WPN as input and assigns an initial capacity to the seed node (John). The initial capacity is the source of the capacities that disseminate to the successors nodes in the WPN. Initial capacity is based on the number of the seed's out-neighbours and the controlling size parameter *m*. When the initial capacity is assigned, the process of propagating capacities among the nodes is started. Dissemination of capacities is controlled by the strength of the relationship, the capacity of intermediate nodes that connect indirect neighbour nodes, and the decay factor *d*. The assigning of the initial capacity and its dissemination procedures have been addressed in more detail in section 3.2.

After building the graph that represents John's network and demonstrates node capacities, the Capacity-first maximum flow algorithm transforms the graph into the Advogato trust metric structure. Then the system applies Capacity-first maximum flow to identify and rank the reliable users. The algorithm searches the graph based on the max-capacity criteria, not based on the shortest path or lowest cost. Thus, it selects firstly the node with the highest capacity and identifies it as the most trustworthy of all the users. It then adds that node's out-neighbour nodes to be compared with the remaining nodes within the previous level. Each time the algorithm adds a new visited node, it recursively examines its capacity and adds its out-neighbour nodes. Whenever an augmented path is found, we add nodes in the path to a list of the seed's trust group $\mathcal{T}_{\mathcal{G}}$ if the node is not an out-neighbour of the seed. From this algorithm we obtained a ranked list of trustworthy users called the Trust group ($\mathcal{TG}$) list which is used to set the access permissions for John's friends. Note that the trustworthy nodes are not ranked in order of the network flow.

As mentioned in the scenario above, we assign weight based on the number of common neighbours between users in John's network. Now assume that John is a Facebook participant who wants to control data that is distributed through his profile. By using the ranked trustworthy group, he allows the users in Level 1, which represent the top 10 ranked trusted friends, to be able to access all his information, post on his wall, write comments, and share posts or photos. In contrast, friends who are distinguished to be in Level 2 are allowed to write comments on John's posts or photos, while friends in Level 3 only have permission

to read what John has posted on his wall. In addition, a user can control access authorities on different data categories, such as discussion topics, calendars, albums, and activities, based on these relationships. For instance, family albums are shared with family members and could be more restrictive by assigning full permission to those members most closely related to John. Members in the second or third level might only be able to view the photos without the comment or sharing rights.

At the same time, participants in online social networks would not like to share information with malicious users and would like to block spammers from accessing their account. Using Facebook as an example, when friends join an application, the application tracks his/her listed friends in order to advertise the application and post comments on their wall. Some users regard these applications as spammers and want to prevent this kind of posting. By using the strength of the application's relationship, we could manage access permissions and prevent these spammers from accessing users' profiles. Using the max-Capacity approach in identifying trusted people and ranking them based on their connection strength would make managing the access permissions convenient and efficient.

4.3 Recommending New Friends

The main difference between social websites and other networks is that social networking sites are designed to *connect* people through the web. Researchers have shown that participants in online social networking sites are most likely to connect to users they know in real life, but at the same time, they often connect with people whom they have never met. Facebook users are likely to connect with people whom they know offline while users in enterprise social networking (like LinkedIn) are interested in adding valuable people whom they do not know on a personal level. Therefore, a link prediction or friend recommendation service is a core feature that is provided to users in many online social network websites. Friend recommendation systems have been proposed as a way of addressing the problem of exploring known friends and suggesting compatible friends who share common interests. Some recommendation systems are using a predefined set of user interests, however, another approach to recommending friends in online social networks is based on computing the similiarities among users based on their profile content and comments. [60], [10], [32]

In addition to the above-mentioned method of recommending new friends in online social networking sites, identifying a personalized list of reliable users and ranking them according to their trust level is an approach for suggesting indirect friends to a given user. In this way, friends of friends who are highly trusted, show a high level of activity, and share many interests with the user's friends are more likely to be added to his/her network. Accordingly,

our algorithm can be used to recommend new friends by customizing the proposed approach for predicting new connections between users within the network based on the type of the relationships and their strengths.

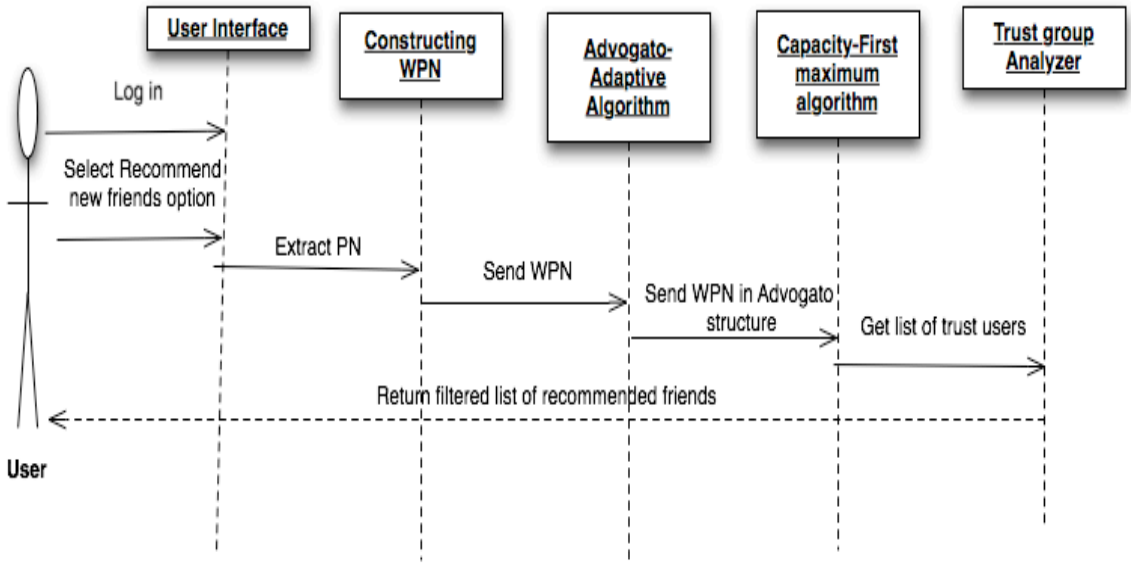

Figure 4.2: Partial sequence diagram for Access Permissions and Friend Recommender system.

As shown in figure 4.2, the input to the system is a given user in a online social network who represents the seed node. The user wants the system to recommend new friends to be added in his/her network. The system requires a social graph with edges that define the relationships between users; we define this as a personal network. The system then builds the WPN by adding weights that show the relationship strength between users in the PN. A user has a variety of ties with people in their network, so the WPN can be different according to the goal of the recommendations. To clarify this point, consider that a user would like to

have a list of friends that are trusted by his/her best friends. In this case, the contacts that have the "best friend" relationship with the user will be assigned a higher weight value, and other weights will be lower based on the strength of the friend relationship. This is similar to how a user might want to have people recommended in a system such as Facebook. In other situations, if the user wants to have family members or experts within their profession or field of study recommended, people who have the appropriate relationships will have a higher weight value than any others in the user's personal networks. The system is inspired by the relationship classification, and the strength of these relations will differ from situation to situation.

After building the WPN, the trust identification algorithm receives it and assigns an initial capacity to the seed node and disseminates this capacity among the nodes in the WPN. It is then transformed into the Advogato structure so that the Capacity-first maximum algorithm can be applied. As mentioned in the previous section and explained in detail in chapter 3, the Capacity-first algorithm will traverse the graph based on the highest capacity rather than the shortest path. Since the Capacity-first algorithm is based on the strength of the connections between users, it ranks the identified nodes by their trust level from the most trusted user to least trusted one.

Once the system identifies the trust group, it sends this list to the analyzer in order to filter the trust group list. The analyzer excludes the friends who are already in the user's friend list, the people who are directly connected to the seed, and the users with a low enough rank in the list. The system then sends the suggested list to the user by displaying the most trusted

users first. For example, the top 10 users are shown first, and then users in the second and third levels are shown respectively.

To summarize, the Capacity-first maximum algorithm is applicable for use in controlling user profile access permissions and for recommending friends in online social networks. Because participants in online social networks make an effort to sort their friends in lists or groups to explicitly identify the relationships, the Capacity–first algorithm can be used to generate this feature for identifying local, reliable people who can be used in online social networks. It is useful for both controlling privacy and predicting links with trust awareness.

In Table 4.1 we illustrate the main features that differentiate our work from the *Advogato* approach. We compare our method with *Advogato* in terms of intuition, the graph type used in each approach, computation type, and the difference in their results.

Table 4.1: Main differences between our work and Advogato trust metric.

Algorithm	Intuition	Graph type	Computation	Output type
Advogato	Network's maximum flow	Non weighted graph	Local computation algorithm	Binary results
Capacity-first maximum flow	Strength of social relationships	Directed weighted graph	Local computation algorithm	Ranked list

Chapter 5 Experiments and Results

This chapter presents the evaluation metrics that we have used to measure the performance of our algorithm along with a comparison of the results against baseline algorithms. Additionally, we discuss the algorithm's sensitivity to certain parameters.

5.1 Experimental Networks

To evaluate the Capacity-first maximum flow algorithm, we selected a directed Epinions signed social network. Members in Epinions can rate others' reviews in terms of usefulness to the users themselves. Moreover, members can indicate whether reviewers are trusted or non-trusted.

The original dataset contains 131,828 nodes and 841,372 edges. The dataset was divided into two sets: small and large sized sets. The average out-degree for the small dataset is 25.1 while the large datset has an out-degree of 85.4. Consequently, the small dataset contains 117 users and 2,941 edges between pairs. The large dataset has 1,681 users and 143,550 connections that link pairs of users. Considering that the social network is signed as to the trustworthiness of relationships, the total number of edges is divided into positive and

negative edges which imply trust and non-trust relationships. Table 5.1 shows the description of small and large datasets that are used to evaluate our algorithm.

	Number of nodes	Number of edges	Average out-degrees	Number of positive edges	Number of negative edges
Small dataset	117	2,941	25.1	2,212	729
Large dataset	1,681	143,550	85.4	130,409	13,141

Table 5.1: Small and large dataset descriptions.

5.2 Evaluation Design and Metrics

In order to measure the performance of our algorithm, two types of evaluation metrics were adopted: precision and recall. Other than these measurements, we report the "error-hit", which indicates the number of malicious users erroneously placed in the resulting list. Moreover, the algorithm is compared with several baseline algorithms to test its performance.

We trained our algorithm using two sets derived from the small dataset. The first one is referred to as "Training 90%", where for each user in the dataset 10% of his/her connected edges are used for test data (test 10%) The second set is "Training 80%", where for each user in the dataset, 20% of his/her connected edges are used for test data (test 20%). In the large

dataset comparisons, the dataset was divided into five training datasets. In each training dataset, we randomly erased 20% of the positive edges that belonged to each user and tried to find them by running algorithms over the five training datasets.

5.2.1 Evaluation Metrics

To evaluate the efficiency of our algorithm, the evaluation measurements were adopted as follows:

1) Precision and Recall of a user at Top-*N*:

Precision and recall measurements are widely used to evaluate the effectiveness of information retrieval systems (IRSs). Precision measures the ratio of retrieved items that are relevant. Precision is commonly used to evaluate recommendation systems. Yet precision's definition in evaluating recommendation systems is slightly different from the one used in IRSs since recommendation systems recommend a fixed number of items. Thus, precision measures the accuracy of the recommended items (users) list by comparing the top N items which represent the list of items that should be recommended to the user.

In our experiments, we were interested not only in measuring precision but also in measuring the recall. Recall estimates the percentage of the users that are identified by the algorithm compared to the number of actual reliable users. In precision and recall measures, we use the positive small and large datasets for each user to form a training set and a positive test set.

We train our algorithm on the training set to generate a list of reliable users, and this set is called the Top-*N*. In order to measure the precision, we take the ratio of the Top-*N* set to the test set itself. Precision of a user is calculated by formula 5.1 [6],[25],[44],[54].

$$\Pr e(u) = \frac{\left|Test^{+}(u) \bigcap TopN(u)\right|}{\left|TopN(u)\right|} \tag{5.1}$$

where $Test^{+}(u)$ is a dataset that contains trusted users who are identified by the user in his/her data by assigning them positive edges, and the Top*N*(u) is the set recommended by the algorithm. Measuring the average precision of all the users at Top-*N* can be formally obtained by equation 5.2.

$$AvgP = \frac{\sum_{u=1}^{k} pre(u)@topN}{k} \tag{5.2}$$

Computing recall makes use of the same datasets that are used to measure precision, but we measure the ratio of retrieved trusted users to the number of all trusted users as shown in equation 5.3. The average recall for all users at Top-*N* is computed by using formula 5.4, where *k* represents the total number of users.

$$\mathrm{Re}c(u) = \frac{\left|Test^{+}(u) \cap TopN(u)\right|}{\left|Test^{+}(u)\right|} \tag{5.3}$$

$$Avg\,\mathrm{R} = \frac{\sum_{u=1}^{k} \mathrm{Re}c(u)@topN}{k} \tag{5.4}$$

2) Error-hit:

Our algorithm is also expected to identify reliable users and prevent unauthenticated people from accessing a user network. To show the algorithm's accuracy for preventing unauthenticated users, we examine the error-hit. The idea behind the error-hit experiment is to measure how often the algorithm under examination includes untrustworthy users in the resulting list of ranked trusted users. In order to do this evaluation, we used the negative test dataset that considers all negative edges. We measure the ratio of the Top-*N* dataset, which is the result of the algorithm, to the negative test dataset to get the error-hit ratio. The formal computation of the error-hit measurement is illustrated in equation 5.5.

$$EH(u) = \frac{\left|Test^{-}(u) \cap TopN(u)\right|}{\left|TopN(u)\right|} \tag{5.5}$$

Where $Test^{-}(u)$ is the negative test dataset that contains all the negative nodes identified by the user as untrustworthy, and TopN(u) is the ranked list of identified trustworthy users

recommended by the algorithm. The average error-rate for all users at Top-*N* is computed by equation 5.6.

$$AvgEH = \frac{\sum_{u=1}^{k} EH(u)@N}{k} \qquad (5.6)$$

5.2.2 Baseline Algorithms

For performance comparisons, we conducted experiments with the following baseline algorithms:

1) The standard *Advogato* approach using the breadth-first search

Advogato is a standard algorithm to compute a maximum flow based on the shortest path. This approach applies a breadth-first search to find paths from the seed node to the supersink based on the shortest path. This algorithm terminates either when there is no new augmented path from the seed to the supersink or when the seed node exceeds its capacity. We implented the Advogato method and ran it on the dataset to compare the results with those obtained using our algorithm.

2) Common neighbours

Common neighbours is commonly used to predict edges between two nodes based on the number of common neighbors between these two nodes [29],[40]. Common neighbours is defined as shown in equation 5.7.

$$Common_{u,v} = \left|O(u) \cap O(v)\right| \qquad (5.7)$$

3) Jaccard's coefficient

Jaccard's coefficient measures the probability of having common neighbours between two nodes u and v to the number of unions of u and v's neighbour nodes. Formally, Jaccard coefficient is calaculated by using formula 5.8 [29],[40].

$$Jaccard|u,v| = \frac{|O(u) \cap O(v)|}{|O(u) \cup O(v)|} \qquad (5.8)$$

4) Random Walk with Restart

Random Walk with Restart (RWR) is a technique that measures how much two nodes are relevant. RWR is used in personalized PageRanks and automatic image captioning. The idea of this algorithm is to assume there is a random walker that has a start node with a certain probability. At every iteration the walker has to decide either to choose an edge to follow among the connecting edges or to go back to the start node based on the restart probability. RWR is computed by equation 5.9. For more details about RWR see [52], [42], and [50]. We denoted it as PageRank during our experiments.

$$\vec{r}_i = c\tilde{W}\vec{r}_i + (1-c)\vec{e}_i \tag{5.9}$$

5) Katz

Katz is a measure that is based more on paths existing between two nodes, where the shorter these paths are, the stronger the relationship is. The Katz measure is the sum of the number of paths from node *u* to *v*, exponentially dampened by length so that short paths are counted as having more weight. Formally, Katz is calculated using equation 5.10 [40].

$$Katz|u,v| = \sum_{\ell=1}^{\infty} \beta^{\ell} . \left| paths_{u,v}^{(\ell)} \right| \tag{5.10}$$

where $paths_{u,v}^{(\ell)}$ is the set of all paths of length-ℓ from *u* to *v*, and β is the damping factor.

5.3 Sensitivity to Parameters

In this section, we provide empirical results that show the effects of the parameter adjustments. Capacity-first maximum flow algorithm has two parameters that can be set to a

range of values: the decay factor *d* and the capacity size controller *m*. Note that in these experiments, we trained our algorithm on the large dataset for Top-10

To examine the effects of these parameters on the results of our algorithm, we set the capacity size controller *m* values as follows: *m*=1, *m*=2, *m*=3, *m*=4, *m*=5, and *m*=6. In addition to the parameter *m* we set the decay factor to values of *d*=1, *d*=0.7, *d*=0.5, *d*=0.3, and *d*=0.1. During the experiments we propagated the capacity of a seed to nodes from which the seed is reachable within 5 hops.

5.3.1 Decay Factor *d*

Assigning large values for the decay factor *d* allows nodes to propagate a large amount of their incoming capacities to successive nodes. Therefore, the algorithm resulted in an expanded list of reliable individuals. In contrast, small values assigned to *d* lead to a list of recommended users close to the seed and put the remote nodes at a disadvantage.[61]

To evaluate the effects of *d*, we compare the results by assigning diverse values to *d*: *d*=1, *d*=0.7, *d*=0.5, *d*=0.3, and finally *d*=0.1. Moreover, we split the experiment into two cases: the effects of *d* when the capacity size controller parameter *m*=1 and when *m*=6. Table 5.2 and 5.3 summarize the average precision, recall, and error rate results obtained when the above-mentioned values of d were assigned. As well, Figure 5.1 shows the average precision, recall, and error-rate at different values of *d* when *m*=1.

	Recall	Precision	Error-hit
d=0.1	0.0044	0.0142	0.0004
d=0.3	0.1022	0.1829	0.0089
d=0.5	0.1156	0.1892	0.0103
d=0.7	0.1168	0.1896	0.0112
d=1	0.1143	0.1877	0.0109

Table 5.2: Recall, precision, and error-hit scores at different values of d when m=1 for Top-10.

	Recall	Precision	Error-hit
d=0.1	0.1164	0.1896	0.0111
d=0.3	0.1172	0.1898	0.0116
d=0.5	0.1170	0.1897	0.0116
d=0.7	0.1168	0.1896	0.0116
d=1	0.1144	0.1877	0.0114

Table 5.3: precision, recall, and error hit at different values of d when m=6 for Top-10.

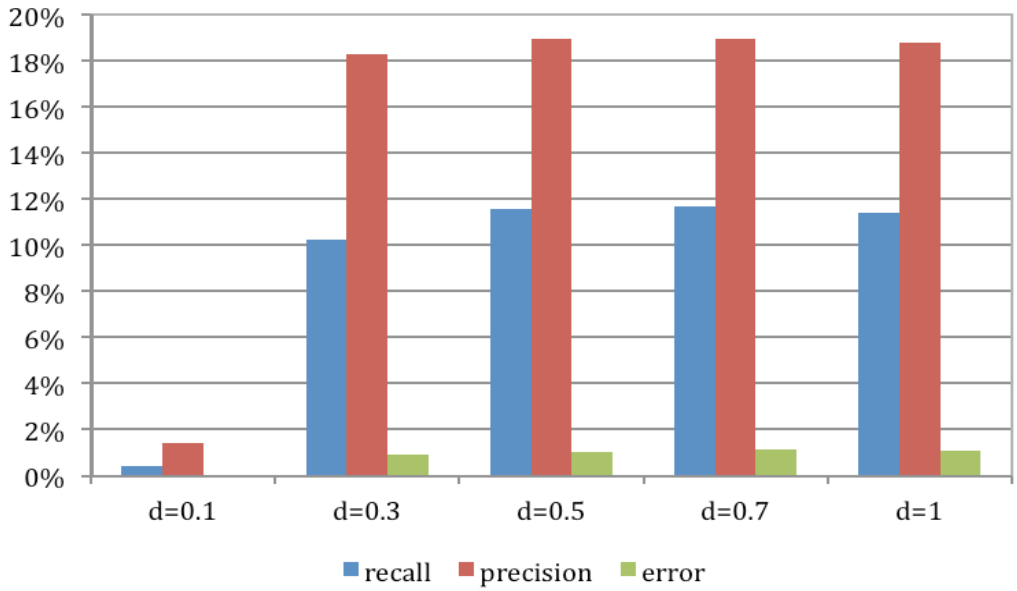

Figure 5.1: Recall, precision, and error-hit scores at different values of d when *m*=1 for Top-10 .

5.3.2 Capacity Size *m*

To control the capacity size, or in other words, to control the trusted group size, we can adjust the parameter *m*. Assigning small values to *m* will produce a small initial capacity for the seed. Consequently, small capacities will be propagated to the immediate neighbour nodes. Assigning a large value to *m* allows us to propagate capacities to a large number of neighbour nodes, which produces a list of an acceptable number of trusted users. The group size depends on the application's domain. For example, in recommending new friends we would like to expand the recommendation list of friends to encompass more users. We

compare the results obtained from setting different values to the parameter *m*, where these values range from 1 to 6.

Parameter m	Recall	Precision	Error-hit
m=1	0.1156	0.1892	0.0108
m=2	0.1171	0.1897	0.0114
m=3	0.1171	0.1897	0.0116
m=4	0.1171	0.1897	0.0116
m=5	0.1171	0.1897	0.0116
m=6	0.1171	0.1897	0.0116

Table 5.4: shows recall, precision, and error hit of *m*'s values at Top-10.

Table 5.4 illustrates the average values of recall, precision, and error-hit of parameter *m* for *Top-10* on the large dataset and the decay factor *d* set to be 0.5. From the table, we could see that from m=2 to m=6 there is no change in the precision, recall, or error-rate. This may happen because we had already assigned enough capacity to the seed node even when *m* was assigned a value of 1. However, *m* can be more effective if we normalize the number of neighbour nodes that have a connection with the seed node since the initial capacity that is assigned to the seed is large enough. In our large dataset, assigning *m* to 5 or 6 would not generate significant effects on the results.

5.4 Effect of Capacity-first Maximum Flow

In this section we compare the performance of the Capacity-first maximum flow algorithm against the *Advogato* approach, which uses the breadth-first search algorithm (BFS) to identify a trusted group of users. We measure the performance of both algorithms using precision and recall to judge how relevant the results are to the user. Moreover, we measure how often the results contain untrustworthy users by reporting the error-hit. These measurements are applied over the small and large sized datasets. We compare the results using different sizes of Top-*N*. For the small dataset, the number of returned users is plotted as data points on the graph curves where the first point of each curve refers to the Top-5 case and the last point is the Top-30 case. For the large dataset, the number of returned users is plotted as data points on the graph curves where the first point of each curve refers to the Top-10 case and the last point is the Top-50 case.

5.4.1 Small Dataset Comparisons

I. Users precision and recall at Top-*N*:

The results obtained from both training datasets show that our algorithm achieved higher precision and recall values than *Advogato* as is depicted in Figures 5.2 and 5.3. Comparing the recall in Training 90%, Capacity-first obtained 29.57% in Top-5 while *Advogato* obtained 19.63%. Capacity-first saw an improvement of 9.94% over the *Advogato* approach.

The Capacity-first improvement in recall is significantly higher than *Advogato* when the number for Top-*N* increases. For instance, in Top-30, Capacity-first saw an improvement of 17.01% over *Advogato*. Additionally, Capacity-first outperformed *Advogato* in terms of precision. *Advogato* had 8.07% precision while Capacity-first obtained 12.48% in Top-5. Precision decreased, as expected, when the number of trust users increased, but Capacity-first still obtained a higher precision than *Advogato,* as shown in figure 5.2. In the case of Training 80%, Capacity-first obtained higher recall and precision than *Advogato*. For example, in Top-10 our algorithm achieved a recall of 42.61% while Advogato attained 33.85%. In the same case, Advogato obtained a precision of 14.38%, whereas Capacity-first achieved 16.61%.

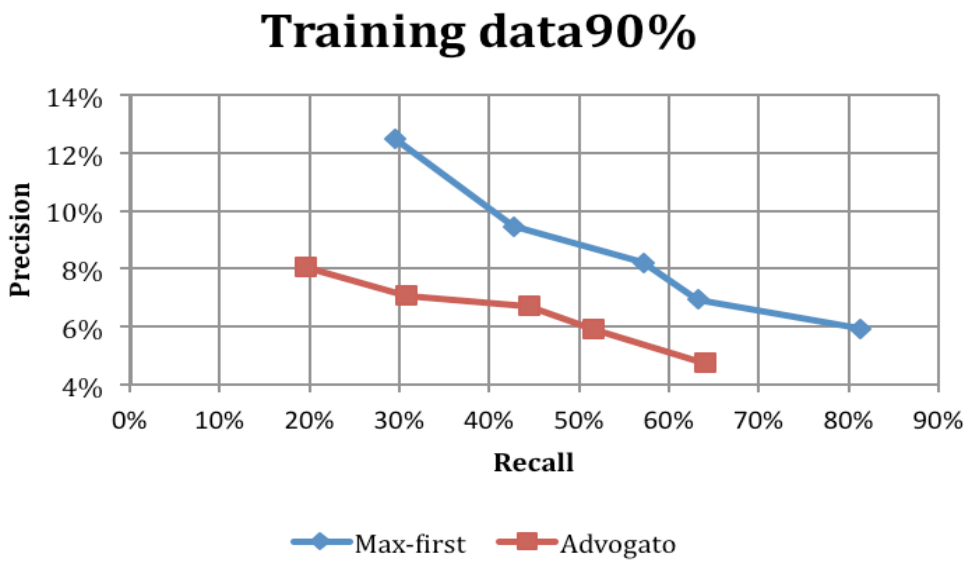

Figure 5.2: Precision and recall of Advogato and Capacity-first at different N cases in Training 90%.

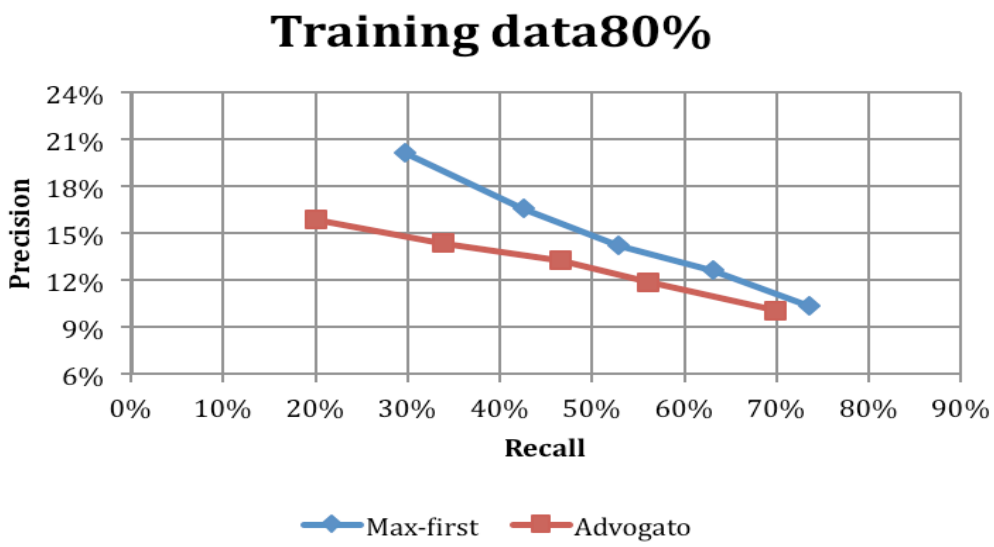

Figure 5.3: Precision and recall of Advogato and Capacity-first at different N cases in Training 80%.

II. Error-hit at Top-*N*:

Figures 5.4 and 5.5 show the results of different cases of Top-*N*. As a result, due to the social network structure used in our dataset, both methods contain some nodes that each user listed as untrustworthy. The test network data had a small number of nodes in which each node had a high degree. Nevertheless, within the Top-5 users, the Capacity-first approach included untrustworthy users at a rate of 8.3% and 7.24% in Training 90% and Training 80% respectively, whereas 10.4% and 10.34% of the users for the *Advogato* were untrustworthy.

Looking at the results within the Top-30 in both training datasets, we also observed that our method achieved less error-hit than the *Advogato* method did.

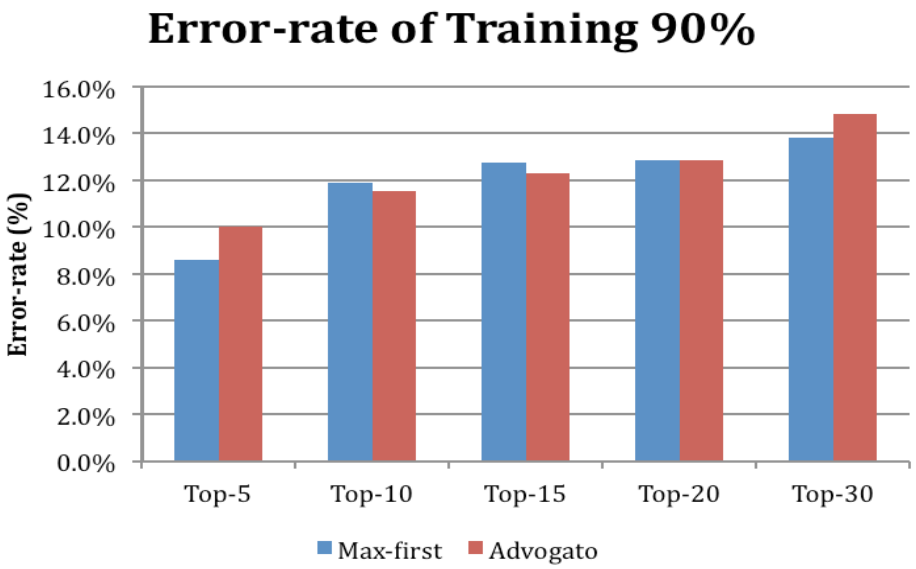

Figure 5.4: Error-rate of Training 90% at different N values.

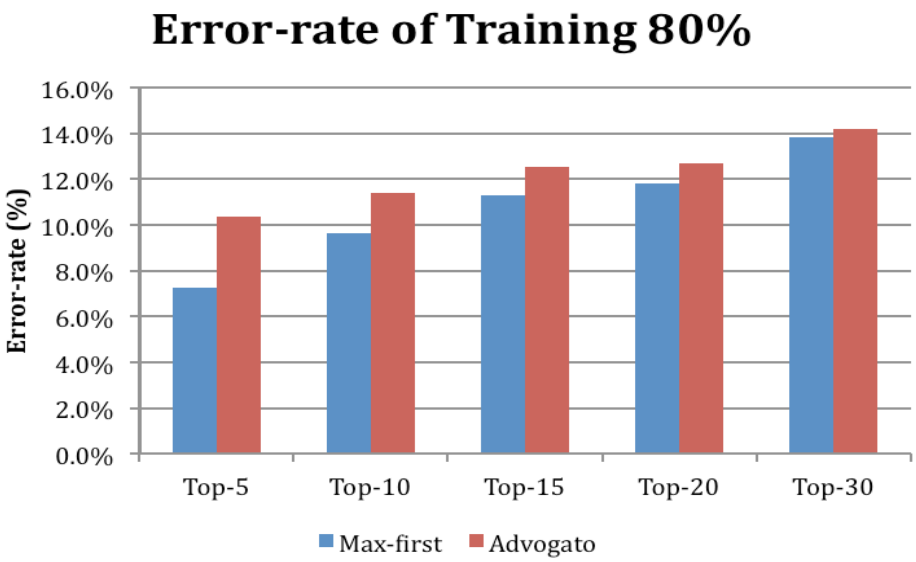

Figure 5.5: Error-rate of Training 80% at different N values.

5.4.2 Large Dataset Comparison

I. Precision and Recall at Top-*N*

We compared the performance of the Advogato approach to the Capacity-first algorithm using values of *m*=1 and *m*=6. The decay factor was set to a value of 0.5 for both comparisons.

The results indicate that our algorithm surpasses the Advogato approach in terms of precision and recall as shown in tables 5.5 and 5.6. When m=6, the Capacity-first algorithm has a

higher value for recall and precision, and the gap between the values of Capacity-first and Advogato increases as the Top-*N* value is increased. For instance, the recall at Top-10 Advogato obtained 5.66% while our method achieved 11.71%, which is twice as high as Advogato. The highest recall value that our algorithm acquired is 35.75% and the highest recall of Advogato is 21.69% at Top-50. For precision, Advogato achieved 7.50%, and our algorithm improved by 4.47% over Advogato when the trust group size was 50.

In the case of Top-20, Capacity-first has a precision of 17.10% while Advogato obtained 10.09%. When the size of the trusted group increases, the gap in precision values becomes smaller, but our algorithm still outperforms Advogato; it improved by 8.51% and 4.47% over Advogato in the cases of Top-10 and Top-50 respectively. Moreover, our proposed method in this work outperforms the Advogato approach when m=1. These results are depicted in figure 5.6, which shows precision and recall obtained by the algorithms at different sizes of N.

Table 5.5: Precision of Advogato and Capacity-first when m=6.

Precision at m=6	Advogato	Capacity-first
Top-10	0.1046	0.1897
Top-20	0.1009	0.1709
Top-30	0.0935	0.1536
Top-40	0.0858	0.1381
Top-50	0.0790	0.1237

Table 5.6: Recall of Advogato and Capacity-first when m=6.

Recall at m=6	Advogato	Capacity-first
Top-10	0.0566	0.1171
Top-20	0.1065	0.2014
Top-30	0.1503	0.2660
Top-40	0.1863	0.3181
Top-50	0.2169	0.3575

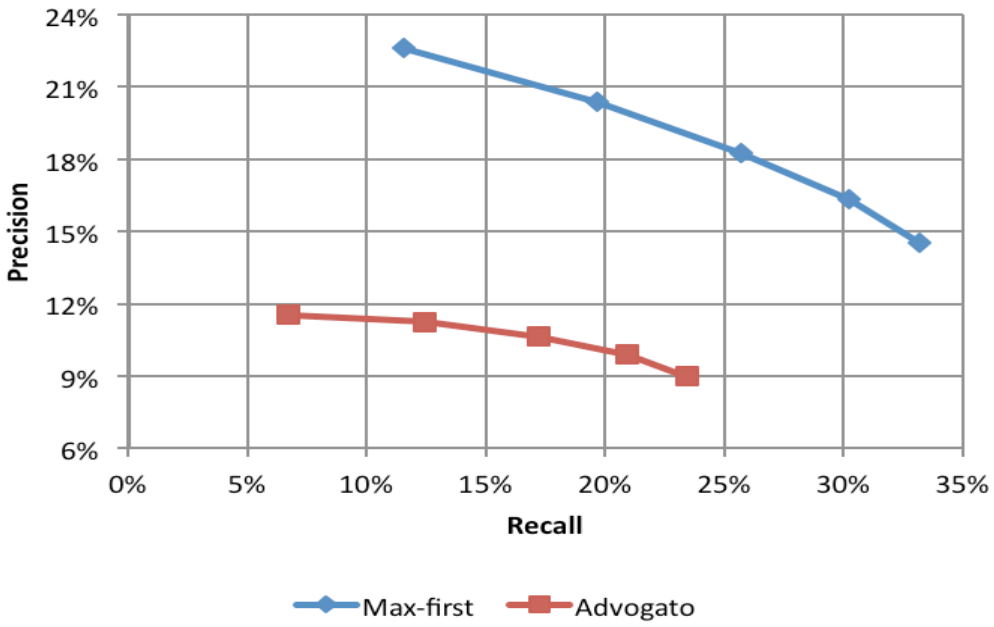

Figure 5.6: : Precision and recall of Advogato and Capacity-first at different N values when m=1.

II. Error-hit at Top-*N*:

Our goal in implementing the max-capacity algorithm is to identify a trust group within a network as well as to prevent untrustworthy people from accessing the private data of a given user. For this reason, we examine how often the resulting ranked reliable users include untrustworthy ones. Figures 5.7 and 5.8 illustrate the results in the cases of Top-10 to Top-50 for both Advogato and Capacity-first algorithms. From the results, one can see that both algorithms include untrustworthy users in their results, yet our method has a lower error rate than Advogato. Obviously, as the number of trust users increases, the difference between the values gets smaller. In general, our method accomplished a smaller error rate than did

Advogato. In both comparisons, when m=1 and m=6, Capacity-first shows a lower error-hit than Advogato, and the reason the results contain these unreliable users is due to the structure of the network. The average outdegree is 85.4 for nodes in the network, and this is considered to be a high degree.

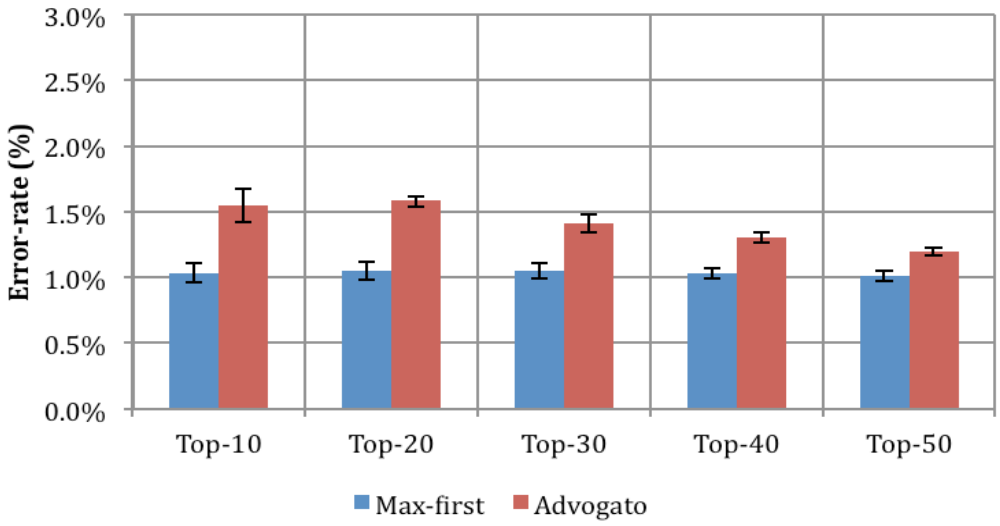

Figure 5.7: Error-hit obtained by Capacity-first and Advogato when m=1.

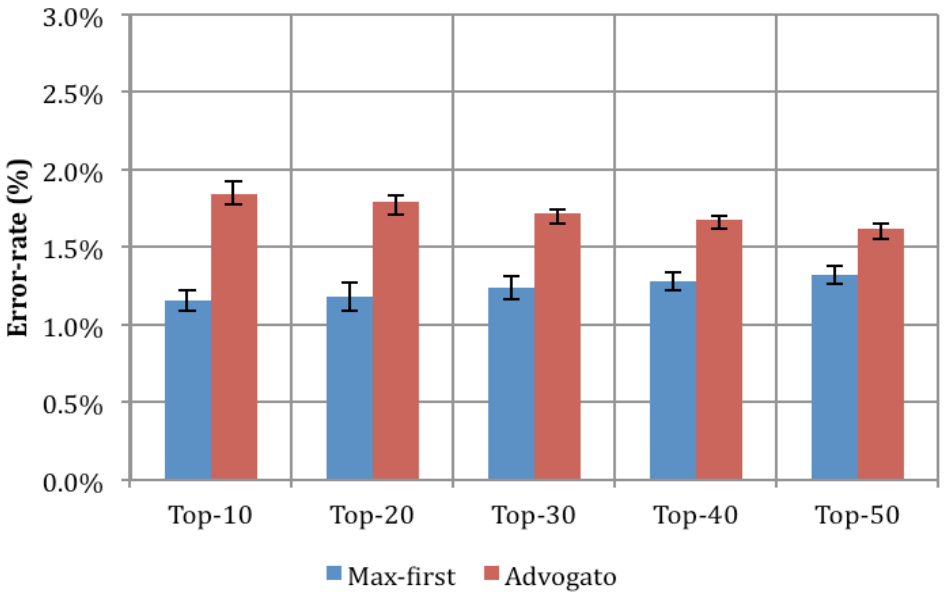

Figure 5.8: Error-hit obtained by Capacity-first and Advogato when m=6.

From the figures above, we conclude that our method records a better performance than the Advogato approach. In both small and large datasets, and when m=1 or m=6, Capacity-first outperformed Advogato, which uses the breadth-first search algorithm. Applying Capacity-first maximum flow over the WPN that has been converted into the Advogato structure identified more reliable users in the network with trust-awareness of untrustworthy people who were then eliminated.

5.5 Comparison with Other Methods

The previous section demonstrated the performance comparison between our algorithm and the *Advogato* approach. In this section, we highlight the results of *Capacity-first maximum flow* over the *Common neighbours*, *Jaccard coefficient, Random Walk with Restart,* and *Katz* methods. We compare the performance against the small set and the large set. Moreover, we evaluate the precision, recall, and error-hit of each algorithm at differ Top-*N* sizes.

5.5.1 Comparison for Small Dataset

I. Precision and Recall at Top-*N*

The precision and recall results that are depicted in figures 5.9 and 5.10 were obtained from Training 90% and Training 80% datasets. They show that our algorithm achieves higher precision and recall than all other baseline algorithms except Katz. The lowest values among the methods used in the comparison go to PageRank. In Training 90%, the highest precision was in the case of Top-5 using Katz (14.31%), and then Capacity-first (12.48%). The lowest precisions were obtained by Jaccard at 9.36%, Common neighbour at 5.69%, and PageRank at 3.49%.

As the size of the trust group increases, the differences in precision get smaller between Capacity-first and Katz. Capacity-first obtained a precision closer to Katz in the case of Top-30, whereas the Katz precision is 5.99% and our method's precision is 5.93%. On the other

hand, recall of the Capacity-first method significantly exceeds the Katz recall in Top-30 as illustrate in figure 5.9.

In the case of Top-5, Capacity-first achieved a recall of 29.57%, though Jaccard, Common Neighbour, and PageRank obtained 25.31%, 17.13%, and 12.95% respectively. Katz exceeded our method by 2.23% in the same case. However, although Katz obtained the highest precision and recall among all baseline algorithms, it also obtained a higher error-hit.

Precision and recall in Training 80% using Capacity-first was higher than all other methods excluding Katz. As mentioned before, Katz also achieved a high error-hit. These results show that the proposed algorithm, as compared to the other approaches, indeed accessed an ordered set of reliable users and discovered desirable hidden users who the user could trust.

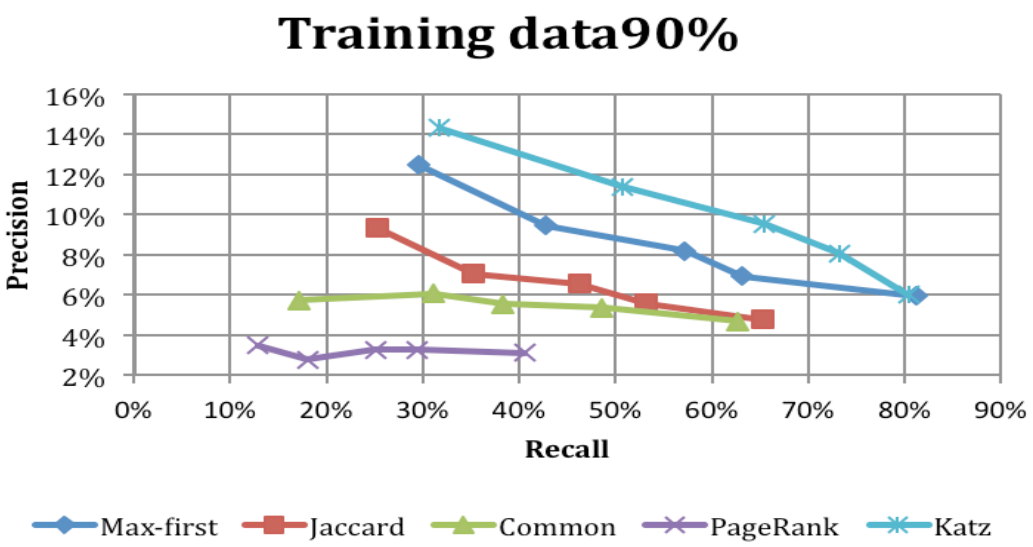

Figure 5.9: Precision and recall of baseline algorithms and capacity-first in Training 90%.

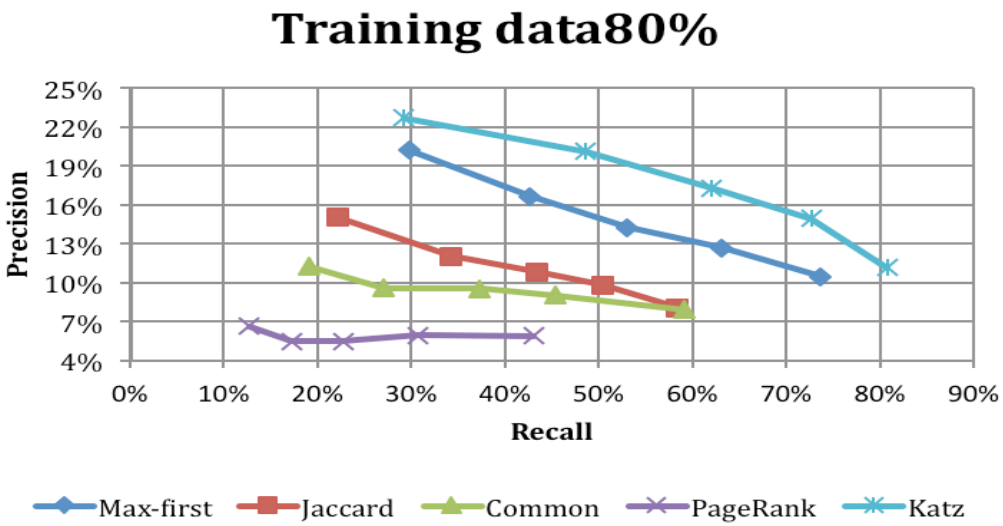

Figure 5.10: Precision and recall of baseline algorithms and capacity-first in Training 80%.

II. Error-hit at Top-N:

Figures 5.11 and 5.12 show the results of Training 90% and Training 80% in cases of different Top-*N*. As can be seen in the figures, our algorithm outperformed all of the baseline algorithms. It is clear that Capacity-first performs better than the baseline methods by having the lowest error rates. In the Top-5 users for Training 90%, Capacity-first included unreliable users 8.3% of the time. Katz, PageRank, Jaccard, and Common Neighbour obtained 11.03%, 15.9%, 11.4%, and 17.3% respectively. In Training 80%, within Top-30, Katz obtained the highest error rate at 15.50% while the Capacity-first approach included only 14% unreliable users. Graphs 5.11 and 5.12 summarize the results of the baseline methods. . Looking at the results of both training datasets within the different Top-*N*, we observe that our new method achieves less error-hit than the baseline methods. Despite these results, the lowest error-hit is demonstrated clearly in the experiments of the large sized dataset since it contains a large number of nodes.

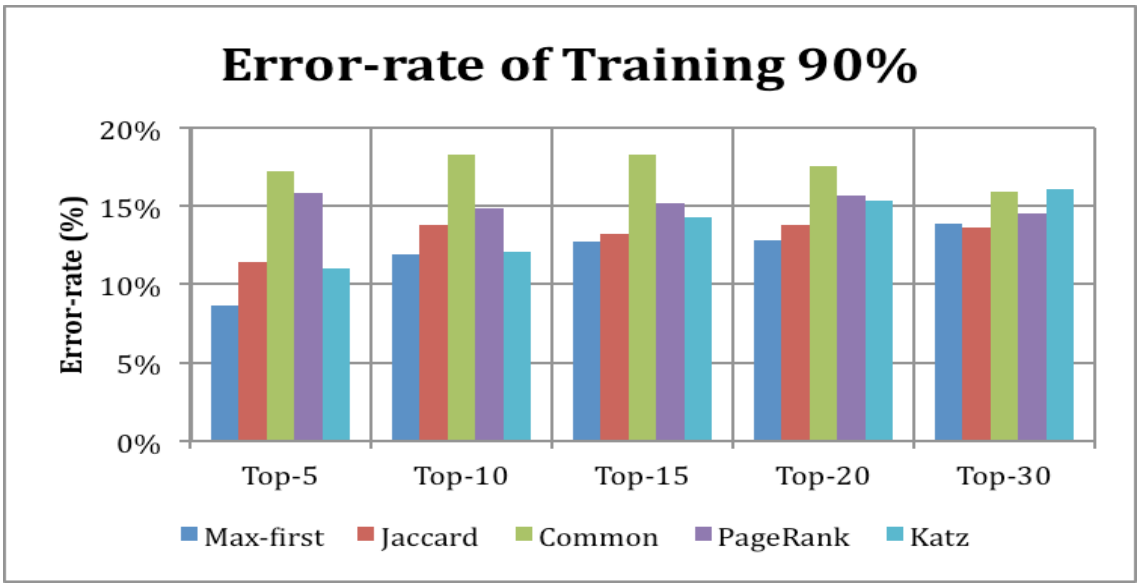

Figure 5.11: Error-rate of Training 90% at different N values.

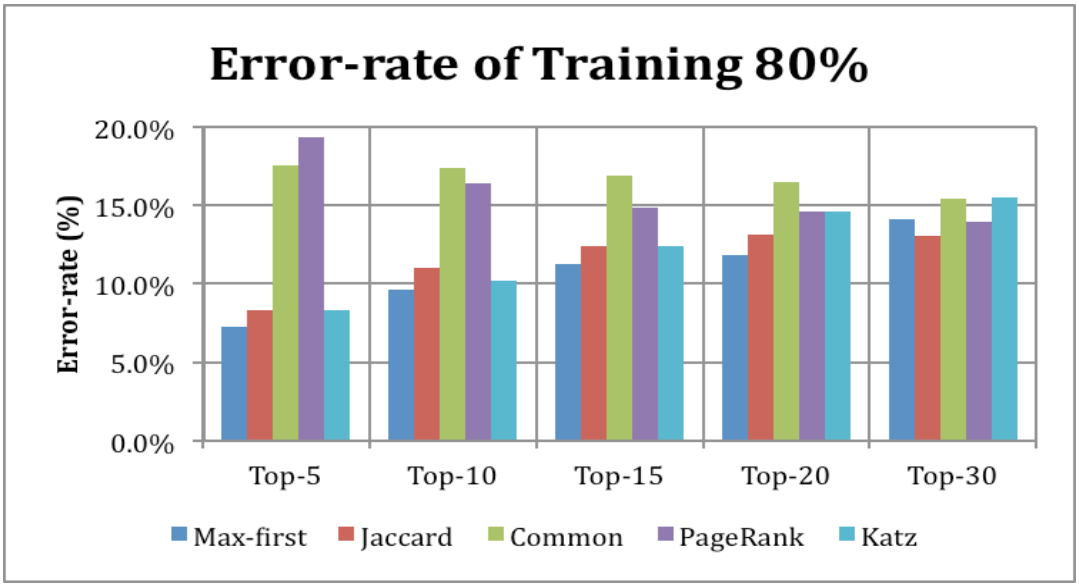

Figure 5.12: Error-rate of Training 80% at different N values.

5.5.2 Comparison for large dataset

I. Precision and Recall at Top-*N*:

Figure 5.13 shows the resulting averages of the compared algorithms by using precision and recall measures. The graph shows that our method outperforms both the Jaccard and Common Neighbour methods. However, Jaccard's precision surpassed the Max-first method by 0.27% in the case of Top-10. When the size of trust group is increased, the precision of Max-first improved over the Jaccard and Common Neighbour methods. For recall, our algorithm exceeds both methods in all cases. The results also note that the Common Neighbour method obtained the lowest value in both precision and recall. Moreover, our method exceeds the PageRank algorithm in all cases. To give more insight into the comparison, PageRank obtained better values than Common Neighbour, but lower values than the other compared algorithms. PageRank achieved 11.46% recall and got 16.77% for precision for the case of Top-10. In this case, Capacity-first obtained 11.71% recall and 18.97% precision. When the size of Top-N increased, the gap between Capacity-first and PageRank's precision and recall values decreased.

Furthermore, for the case of Top-50, Capacity-first has 35.75% in recall while PageRank obtained 32.75%. In the same situation, precision for Capacity-first was 12.37% whereas PageRank had 10.55%. In contrast, the Katz algorithm has the highest values in precision and recall over all the methods including our proposed method, however, the gap between Capacity-first and Katz is never significant. The results show that in Top-10 Katz is only better by 1.35% for recall and 2.7% for precision. This gap decreases in size when the

number of N increases. In Top-50 for instance, Katz improved by 0.61% over Capacity-first for precision.

Figure 5.14 shows the results of precision and recall obtained by the baseline algorithms and Capacity-first when we set m=1. It is apparent that Katz provides good results in identifying trusted people, but this is not the sole goal for proposing the Capacity-first algorithm. We consider preventing malicious users from accessing private data as well, and to this point Katz obtained a higher error rate, which indicates a high number of malicious users within the list of trusted users recommended by Katz. These results are discussed in next subsection.

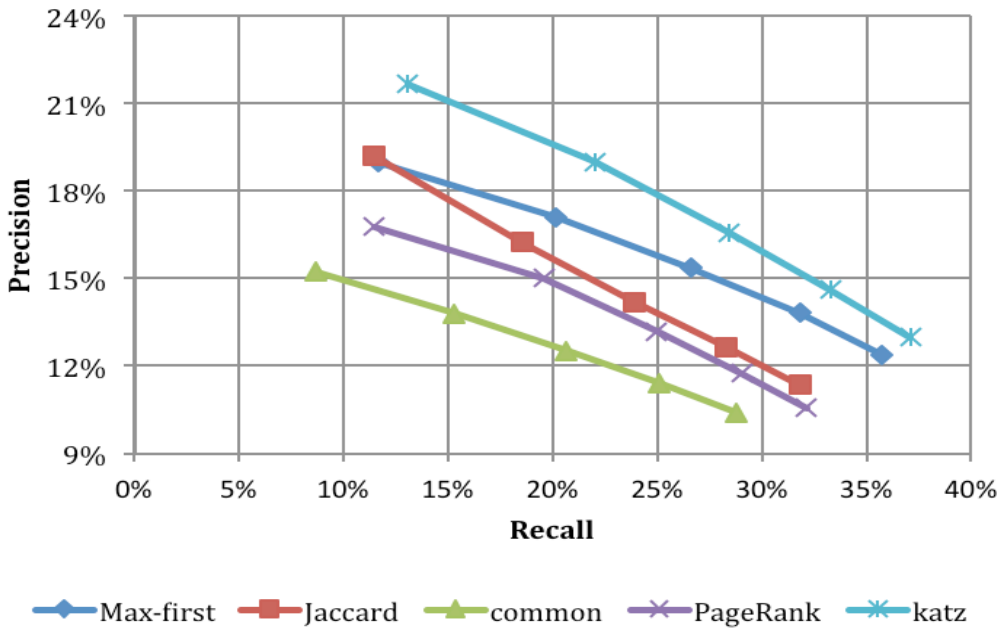

Figure 5.13: Precision and recall of baseline algorithms and Capacity-first at different N values when m=6.

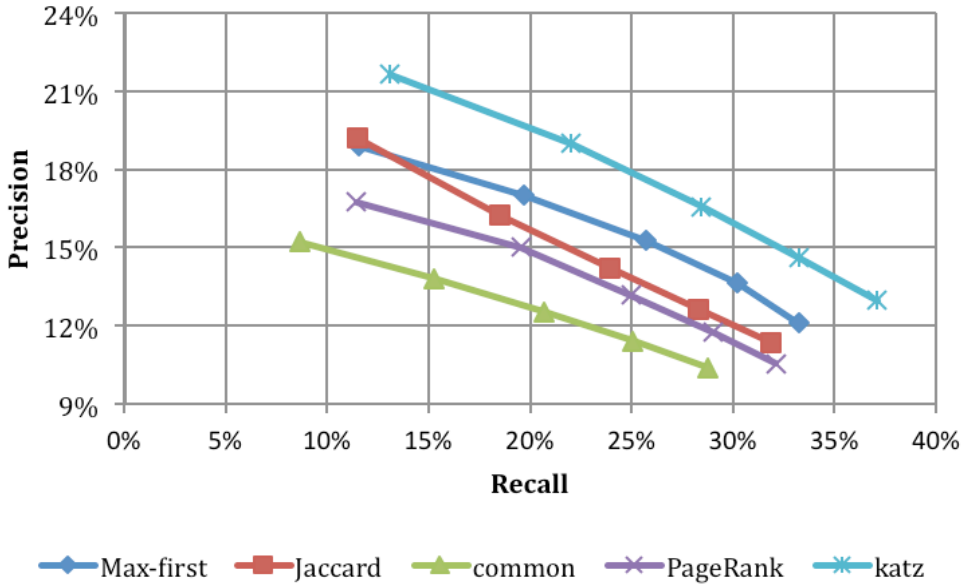

Figure 5.14: Precision and recall of baseline algorithms and Capacity-first at different N values when m=1.

II. Error-hit at Top-*N*

To provide a comparison between algorithms we used the error-hit metric to evaluate the accuracy of our algorithm in preventing unreliable users from accessing the network. Our algorithm concentrates on finding trusted people while blocking unauthenticated people from impinging with a chain of connected users. Decisively, the Capacity-first method achieved the lowest value of error-hits among all the comparable methods. This is clearly depicted in figures 5.15 and 5.16. Although Katz obtained the highest recall and precision, it obtained the third highest error rate after PageRank and Common Neighbour. In Top-10, Capacity-

first achieved 1.16%, and PageRank, Common Neighbour, and Katz obtained 3.24%, 2.82%, and 2.44% respectively. Additionally, Jaccard obtained a 1.31% error-rate, which is higher than Max-first by 0.15%. From the figures, we see that Capacity-first has the lowest error rate compared to the other methods, which accomplishes our goal of including a minimal number of unauthenticated users.

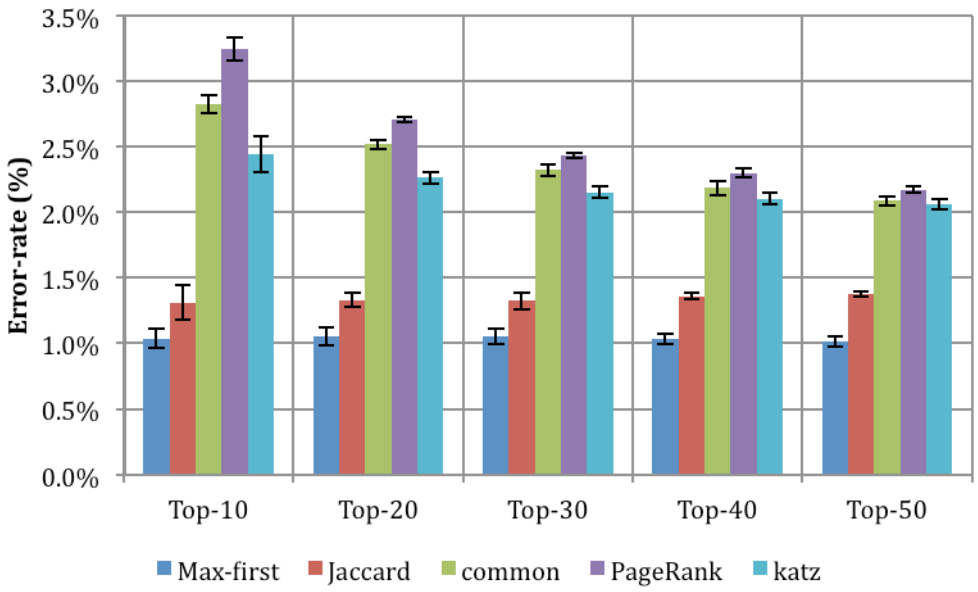

Figure 5.15: Error-rate of baseline algorithms and Capacity-first at different N when m=1.

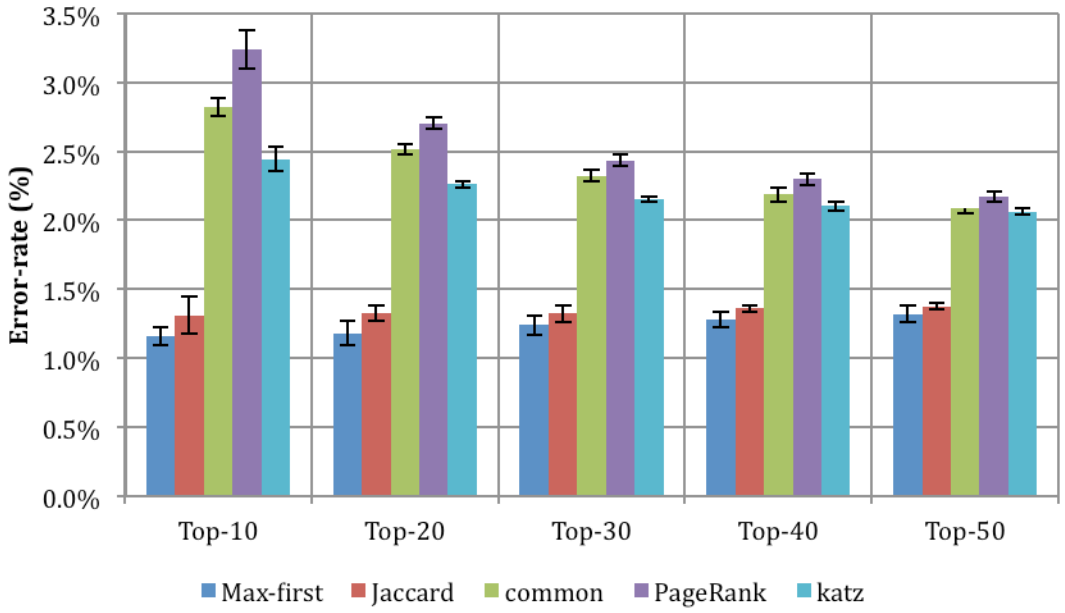

Figure 5.16: Error-rate of baseline algorithms and Capacity-first at different N when m=6.

Our objective in this work is to find as many as possible trustworthy users within a network taking into consideration the blocking of untrustworthy users from the network. Thus, even though Katz achieved better results in uncovering trusted users, the high number of untrustworthy users included in the results affects the accuracy of the results in a negative way. We conclude from all comparison experiments proposed in this chapter that our algorithm provides the best quality results not only in terms of identifying trustworthy neighbours, but also in dealing with blocking unreliable users.

Chapter 6 Conclusion and Future Work

People in online social networks perpetuate existing relationships while looking to establish new ones and exchange information with individuals who they consider to be valuable users. Because of the rapid growth of these online social networks and other social media, there is a new and significant concern for privacy issues. Participants in these websites want to protect their information from malicious users, so there is a need to identify reliable users within a network and prevent the untrustworthy ones from misusing the information. In this thesis, a new method to form a group of trust based on the user perspective is introduced. It uses the strength of relationships in the user's network to adapt the Advogato trust metric. This involves propagating a capacity along the chain of social connections. By applying the Capacity-first maximum algorithm, it is possible to identify local, trusted users and rank them based on their trust level.

In order to evaluate the Capacity-first maximum flow algorithm's performance, we compare it against five algorithms: Advogato, Jaccard coefficient, Common Neighbour, Random Walk with Restart (personalized PageRank), and Katz. Firstly, we started the evaluation by measuring the effectiveness of the algorithm against the Advogato approach since it is the adaptive metric. Then, we compared the algorithm performance with the rest of the baseline methods. In our experiments, we used precision and recall metrics as performance measurements, and error-hit metrics to indicate the error rate of existing non-trusted users within a list of recommended reliable users obtained by algorithms. The results of testing our

algorithm's performance against Advogato show that the Capacity-first method outperforms Advogato in terms of both precision and recall. Moreover, our algorithm achieves a lower error-rate than Advogato. In comparing precision and recall achieved by Capacity-first, Jaccard, Common Neighbour, RWR, and Katz, we conclude that Capacity-first exceeds all the other algorithms, excluding Katz. Katz records a higher precision and recall than Capacity-first which means that it identifies more reliable users. Katz, however, shows a high error rate, which is undesirable since our goal is to identify trustworthy people while preventing malicious users from accessing a network. On the contrary, our proposed algorithm attains the lowest error rate over all comparable algorithms. Overall, the experimental results demonstrate that our proposed algorithm outperforms other algorithms and provides significant advantages in finding a trustworthy group of people and in blocking anonymous unauthenticated users.

Social network sites provide users with features for communicating with trusted users and encourage them to establish new relationships with compatible people. Thus, in the future, social network sites will become an even more important tool for sharing information and media content. For future work, we intend to embed an inferring social relationship mechanism in order to classify the ties that connect users within networks, and we are considering large-scale networks for further experiments. In addition, we plan to apply our algorithm to specific software in order to control access to the resources in online social networks.

Bibliography

[1] Retrieved 12,28, 2011, from www.facebook.com/press/info.php?statistics

[2] Retrieved 01,08, 2012, from http://www.facebook.com/press/info.php?timeline

[3] Retrieved 01,01, 2012, from http://www.csoonline.com/article/527970/facebook-twitter-social-network-attacks-tripled-in-2009-

[4] Retrieved 01,01, 2012, from http://www.sophos.com/en-us/

[5] Retrieved 10,13, 2011, from http://dictionary.cambridge.org/dictionary/british/trust_1?q=trust

[6] Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as classification: Using social and content-based information in recommendation. In Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, AAAI '98, pages 714–720, Menlo Park, CA, USA. American Association for Artificial Intelligence.

[7] Blomqvist, K. (1997). The many faces of trust. *Scandinavian Journal of Management, 13*(3), 271-286.

[8] Boyd D.M., and Ellison, N.B. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication13, pp. 210-230.

[9] Caverlee, J., and Webb, S. (2008). A large-scale study of MySpace: Observations and implications for online social networks. In *Proceedings of* the 2nd International Conference on Weblogs and Social Media, *ICWSM, 8*

[10] Chen, J., Geyer, W., Dugan, C., Muller, M., and Guy, I. (2009). Make new friends, but keep the old: Recommending people on social networking sites. *Proceedings of the 27th International Conference on Human Factors in Computing Systems,* pp. 201-210.

[11] Deutsch, M. (1958). Trust and suspicion. Journal of Conflict Resolution 2, 265-279.

[12] Dwyer, C., Hiltz, S. R., and Passerini, K. (2007). Trust and privacy concern within social networking sites: A comparison of facebook and MySpace. *I*n Proceedings of the Thirteenth Americas Conference on Information Systems.

[13] Ellison, N. B., Steinfield, C., and Lampe, C. (2007). The benefits of facebook "friends:" social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication, 12*(4), 1143-1168.

[14] Fix, C. L. (2009). *An Analysis of Trust in Deception Operations, Mater's thesis,* Naval Postgraduate School, MONTEREY, CALIFORNIA**.**

[15] Garton, L., Haythornthwaite, C., and Wellman, B. (1997). Studying online social networks. *Journal of Computer-Mediated Communication, 3*(1).

[16] Gefen, D. (2002). Reflections on the dimensions of trust and trustworthiness among online consumers. *ACM SIGMIS Database, 33*(3), 38-53.

[17] Gefen, D., & Straub, D. W. (2004). Consumer trust in B2C e-commerce and the importance of social presence: Experiments in e-products and e-services. *Omega, 32*(6), pp.407-424.

[18] Golbeck, J. (2009). Trust and nuanced profile similarity in online social networks. *ACM Transactions on the Web (TWEB), 3*(4), 12.

[19] Golbeck, J., & Hendler, J. (2004). Accuracy of metrics for inferring trust and reputation in semantic web-based social networks. In *International Conference on Knowledge Engineering and knowledge Management (EKAW)*, pp.116-131.

[20] Golbeck, J., & Hendler, J. (2006). Inferring binary trust relationships in web-based social networks. *ACM Transactions on Internet Technology (TOIT), 6*(4), pp.497-529.

[21] Golbeck, J. (2005). Computing and applying trust in web-based social networks. Ph.D. thesis, University of Maryland, College Park, MD.

[22] Golbeck, J. (2009). Computing with social trust. Paper presented at the *Human-Computer Interaction Series,* pp. 335.

[23] Gong, Y., Yang, F., Su, S., & Zhang, G. (2009). Improve peer cooperation using social peer-to-peer networks. *1st International Conference on Information Science and Engineering (ICISE),* pp. 253-257.

[24] Hangal, S., MacLean, D., Lam, M. S., & Heer, J. (2010). All friends are not equal: Using weights in social graphs to improve search. *4th ACM Workshop on SONAM.*

[25] Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS), 22*(1), pp. 5-53.

[26] Hong, D., & Shen, V. Y. (2008). Setting access permission through transitive relationship in web-based social networks. *Weaving Services and People on the World Wide Web,* pp.229-253.

[27] Huang, C., Chen, Y., Wang, W., Cui, Y., Wang, H., & Du, N. (2010). A novel social search model based on trust and popularity. *3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT),* pp. 1030-1034.

[28] Huang, Z., Li, X., & Chen, H. (2005). Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries,* pp. 141-142.

[29] Jamali, M., & Ester, M. (2009). TrustWalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 397-406.

[30] Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web,* pp. 640-651.

[31] Katz, Y., & Golbeck, J. (2006). Nonmonotonic reasoning with web-based social networks. *Proceedings of the Workshopon Reasoning on the Web, WWW06, Edinburgh, UK.*

[32] Kazemi, A., & Nematbakhsh, M. (2011). Finding compatible people on social networking sites, a semantic technology approach. *Second International Conference on Intelligent Systems, Modelling and Simulation (ISMS), 2011,* pp. 306-309.

[33] Kini, A., & Choobineh, J. (1998). Trust in electronic commerce: Definition and theoretical considerations. *In Proceedings of the Thirty-First Hawaii International Conference on System Sciences, \4.* pp. 51-61 vol. 4.

[34] Kuter, U., & Golbeck, J. (2007). Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In *Proceedings of the National Conference on Artificial Intelligence* (AAAI)*, 22.* (2).

[35] Lenhart, A., Madden, M., Smith, A., & Macgill, A. (2007). Teens and social media: An overview. *Pew Internet and American Life Project, Washington, DC.* Retrieved June 15,2011,from http://www.pewinternet.org/pdfs/PIP_Teens_Social_Media_Final.pdf

[36] Levien, R. (2009). Attack-resistant trust metrics. In *Computing with Social Trust,* Human-Computer Interaction Series. pp.121-132.

[37] Levien, R., & Aiken, A. (1998). Attack-resistant trust metrics for public key certification. *In Proceedings of 7th USENIX Security Symposium*, Jan 26-29, San Antonio, Texas.pp. 229-242.

[38] Levien, R. L. (2002). An attack-resistant, scalable name service. Draft submission to the Fourth International Conference on Financial Cryptography.

[39] Li, H., & Singhal, M. (2007). Trust management in distributed systems. *Computer,* IEEE Computer Society, *40*(2), 45-53.

[40] Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology, 58*(7), 1019-1031.

[41] March, S. (1994). Formalising trust as a computational concept. *Ph.D. Dissertation,* Department of Mathematics and Computer Science, *University of Stirling.*

[42] Pan, J. Y., Yang, H. J., Faloutsos, C., & Duygulu, P. (2004). Automatic multimedia cross-modal correlation discovery. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* pp. 653-658.

[43] Rosenblum, D. (2007). What anyone can know: The privacy risks of social networking sites. *IEEE Security and Privacy,* vol. 5, no. 3. PP. 40-49.

[44] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce.* pp. 158-167.

[45] Shi, J., v. Bochmann, G., & Adams, C. (2005). A trust model with statistical foundation. In proceedings of 2nd International workshop on Formal Aspects in Security and Trust, Toulouse, France. *PP.*145-158.

[46] Song, H. H., Cho, T. W., Dave, V., Zhang, Y., & Qiu, L. (2009). Scalable proximity estimation and link prediction in online social networks. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC),* pp. 322-335.

[47] Sztompka, P. (1999). *Trust: A sociological theory*. Cambridge University Press, Cambridge, UK..

[48] Taherian, M., Amini, M., & Jalili, R. (2008). Trust inference in web-based social networks using resistive networks. In the *Third International Conference on Internet and Web Applications and Services, ICIW'08.*pp. 233-238.

[49] Ten Kate, S. (2009). Trustworthiness within social networking sites: A study on the intersection of HCI and sociology. *Unpublished Business Studies Master, University of Amsterdam, Amsterdam.*

[50] Tong, H., Faloutsos, C., & Pan, J. Y. (2008). Random walk with restart: Fast solutions and applications. *Knowledge and Information Systems, 14*(3). PP. 327-346.

[51] Volakis, N. ,(2011). Trust in Online Social Networks. Master thesis, School of Informatics, University of Edinburgh.

[52] Wang, C., Jing, F., Zhang, L., & Zhang, H. J. (2006). Image annotation refinement using random walk with restarts. In *Proceedings of the 14th Annual ACM International Conference on Multimedia,* pp. 647-650.

[53] Wang, E. A survey of web-based social network trust. ITEC 810 final report.

[54] Wang, J., Robertson, S., de Vries, A. P., & Reinders, M. J. T. (2008). Probabilistic relevance ranking for collaborative filtering. *Information Retrieval, 11*(6). PP.477-497.

[55] Wang, Q., Wang, W., Cui, Y., Du, N., & Wang, H. (2010). Improved trust path searching in mobile social networks. In the *3rd IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*. pp. 524-528.

[56] Wang, Y., & Vassileva, J. (2003). Trust and reputation model in peer-to-peer networks. *In Proceedings of the Third International Conference on Peer-to-Peer Computing .(P2P 2003)*. pp. 150-157.

[57] Wang, Y. D., & Emurian, H. H. (2005). An overview of online trust: Concepts, elements, and implications. *Computers in Human Behavior, 21*(1). PP.105-125.

[58] Wang, Y., & Vassileva, J. (2003). Bayesian network trust model in peer-to-peer networks.In *Proceedings of IEEE/WIC International Conference on Web Intelligence* (WI), pp. 372– 378.

[59] Xiang, R., Neville, J., & Rogati, M. (2010). Modeling relationship strength in online social networks. In *Proceedings of the 19th International Conference on World Wide Web(* WWW '10). pp. 981-990.

[60] Zheleva, E., Getoor, L., Golbeck, J., & Kuter, U. (2010). Using friendship ties and family circles for link prediction. *Advances in Social Network Mining and Analysis, Springer PP.*97-113.

[61] Ziegler, C. N. (2005). Towards decentralized recommender systems. PhD thesis, Albert-Ludwigs-Universit¨at Freiburg, Freiburg i.Br., Germany, June 2005.

[62] Ziegler, C. N., & Lausen, G. (2004). Spreading activation models for trust propagation. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service*. pp. 83-97.