# ABSTRACT

Title of dissertation: COMPUTING AND APPLYING TRUST IN WEB-BASED SOCIAL NETWORKS

Jennifer Ann Golbeck, Doctor of Philosophy, 2005

Dissertation directed by: Professor James Hendler
Department of Computer Science

The proliferation of web-based social networks has lead to new innovations in social networking, particularly by allowing users to describe their relationships beyond a basic connection. In this dissertation, I look specifically at trust in web-based social networks, how it can be computed, and how it can be used in applications. I begin with a definition of trust and a description of several properties that affect how it is used in algorithms. This is complemented by a survey of web-based social networks to gain an understanding of their scope, the types of relationship information available, and the current state of trust.

The computational problem of trust is to determine how much one person in the network should trust another person to whom they are not connected. I present two sets of algorithms for calculating these trust inferences: one for networks with binary trust ratings, and one for continuous ratings. For each rating scheme, the algorithms are built upon the defined notions of trust. Each is then analyzed theoretically and with respect to simulated and actual trust networks to determine how accurately they calculate the opinions of people in the system. I show that in both rating schemes the algorithms presented can be expected to be quite accurate.

These calculations are then put to use in two applications. FilmTrust is a website that combines trust, social networks, and movie ratings and reviews. Trust is used to personalize the website for each user, displaying recommended movie ratings, and ordering reviews by relevance. I show that, in the case where the user's opinion is divergent from the average, the trust-based recommended ratings are more accurate than several other common collaborative filtering techniques. The second application is TrustMail, an email client that uses the trust rating of each sender as a score for the message. Users can then sort messages according to their trust value.

I conclude with a description of other applications where trust inferences can be used, and how the lessons from this dissertation can be applied to infer information about relationships in other complex systems.

# COMPUTING AND APPLYING TRUST IN WEB-BASED SOCIAL NETWORKS

by

Jennifer Ann Golbeck

Dissertation Submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2005

Advisory Committee:

Professor James Hendler, Chair/Advisor
Professor Ashok Agrawala
Professor Mark Austin
Professor Benjamin Bederson
Professor Lise Getoor
Professor Ben Shneiderman

To my parents

# ACKNOWLEDGEMENTS

First, I would like to thank James Hendler, my advisor. He gave me great intellectual freedom to pursue my interests and provided encouragement and guidance throughout this work's lifetime. Thanks also to my committee for their challenges, assistance, and support:  Ben Bederson, Ben Shneiderman, Ashok Agrawala, Lise Getoor, and Mark Austin.

I received what seemed like endless help from members of MINDSWAP in many capacities. Thanks to Yarden Katz, Mike Grove, Aditya Kalyanpur, Evren Sirin, Ron Alford, Amy Alford, Debbie Heisler. Aaron Mannes, Denise Cross, and others I may have forgotten. Special thanks to Bijan Parsia who has been a tireless advocate and supportive colleague for the life of this work, and who also co-authored the work that appears as section 10.3. Thanks also to the FOAF community for their support and participation.

Many colleagues around the world have helped me develop this work into what it is now. Thanks to Cai-Nicolas Ziegler, Paolo Massa, Matthew Richardson, Morten Frederiksen, Chris Bizer, and Sep Kamvar. Thanks also to Stuart Kurtz, my former advisor at the University of Chicago, who helped set me on my way toward this goal.

My family, of course, has been very supportive and encouraging. Thanks to brother Tom Golbeck and his wife and my friend Michelle, Jeanne Mitchell, and the rest of my large Catholic family who would require several pages to be fully enumerated. As always, bones to π and K who were present throughout the dissertation process.

Of course, a very special thanks to my husband, Dan Golbeck. He has endless wells of patience and support, and always knows the right moment to tell me that I'm brilliant so I'll keep going. He deserves some sort of degree for putting up with me while I completed this dissertation.

Finally, thanks to Irene and John Golbeck, my mom and dad. They have encouraged me every step of my life to be a strong, independent thinker, to work hard, and to keep trying at difficult things. I inherited a different kind of insanity from each of them, and the combination has served me well through all my years of education. I am eternally grateful to them for all the opportunities they made available to me, and for the support they have given me along the way.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# Chapter 1

# Introduction

The vast public interest in social networks has opened up many new spaces of possible research in computing. This research adopts web-based social networks as the foundation for studying trust. The goal of this work is twofold: First, find ways to utilize the structure of social networks and the trust relationships within them to accurately infer how much two people that are not directly connected might trust one another, and second, show how those  trust inferences can be integrated into applications. The ultimate goal is to create software that is intelligent with respect to the user's social preferences such that the user's experience is personalized, and the information presented to them is more useful.

Tens of millions of users participate in web-based social networking. The web-based nature of these networks means that the data is publicly available; the websites that are taking advantage of Semantic Web technologies, such as FOAF, have even taken this a step further, making the social network information easily available to any system that wants to incorporate it. Similarly, the role of social trust in computing is becoming a prominent topic for research on the Semantic Web, within human-computer interaction, and in the larger computing community as a whole.

In this work, I look at instances where trust is integrated into a social network. The first step to facilitate that integration is to have a definition of trust that captures the social features while being narrow enough to function in the environment of a social network. Given two people, Alice and Bob, I define trust as follows: *Alice trusts Bob if she commits to an action based on a belief that Bob's future actions will lead to a good outcome*. From that definition, functional properties of trust can be extracted, including transitivity, composability, asymmetry, and personalization.

This definition has allowed for the development of two naturally-evolved trust networks that are used in this research. The first has nearly 2,000 members and is entirely based on the semantic web. Using an ontology I created to extend the Friend of a Friend (FOAF) vocabulary, the network is created by spidering files on the semantic web and building a centralized model. The second network is also available on the semantic web, but has a more typical web-based social network structure, with user accounts and a central website. This trust network backs the FilmTrust website, and has over 300 members.

Using these foundations of trust in web-based social networks, and real networks as testbeds, I move toward inferring trust within the network. If two individuals are not directly connected, a trust inference uses the paths that connect them in the social network, and the trust values along those paths, to come up with a recommendation about how much one person might trust the other. I present algorithms for inferring trust in networks where trust is assigned on a binary scale, and when it uses a continuous range of values. In both cases, I show that trust can be inferred quite accurately.

The natural question that follows regards how the inferred trust values can be used. I demonstrate this, and their benefit, through two applications. The first is FilmTrust, a web-based social network that is integrated into a movie rating and reviews website. The trust values are used to personalize the user experience. Reviews are ordered according to the trustworthiness of the author, as calculated from the users perspective. Trust values are also used to create personalized recommended movie ratings for the user. When the user looks at a specific movie, they are shown the overall average rating, as well as the recommended rating calculated using trust values as weights. I show that when the user's opinion is divergent from the average opinion, that the trust-based recommendations significantly outperform both the average rating and the ratings generated by traditional collaborative filtering algorithms. The second application to use trust values is TrustMail, and email client that displays the trust rating of the sender next to each message. Users can sort their inboxes according to the trustworthiness of the sender, with the goal of identifying useful and important messages that might otherwise be missed.

The contributions of this work are relevant within the space of trust and social networks, but also as a general technique within complex systems. The analysis of the type of network and functional properties of the relationship (trust in this case) is what lead to algorithms for inferring indirect relationships in the system. That same type of analysis can be used to develop algorithms for other complex systems. I envision carrying this work into other spaces to show this connection. One project where I have already begun this work is with food webs, an ecological network illustrating which species eat which species in an ecosystem. The same sort of analysis used for trust can be applied there, utilizing phylogenic, taxonomic, and known trophic relationships to infer possible trophic links that have not been observed. The promise of such techniques is to help the users of a system – be it web users reviewing movies, or scientists interacting with their own specific system – to better understand a layer of the complexity and thus help them make more informed and better decisions.

## *1.1 Contributions*

The main focus of this dissertation is to illustrate how an analysis of the trust relationships in web-based social networks can lead to methods for inferring relationships, and that those inferred values, when integrated into applications, can enhance the user experience.

My contributions can benefit research in online communities, the semantic web, recommender systems, and complex systems analysis. Through this work I have shown that using inferred trust relationships in web-based social networks offers some real benefits to the user. In order to accomplish this:

- I present a formalization of trust as a computational concept within web-based social networks, by presenting a definition and describing the functional properties of trust that follow from the definition.
- I present a set of algorithms for inferring trust relationships in social networks that are shown to be quite accurate.
- I show that using trust inferences to make predictive recommendations can offer significant improvements when the user's opinion is divergent from the average.

## *1.2 Organization*

For this dissertation, I have chosen trust in web-based social networks as a very specific area to study the larger issue of trust, reputation, and relationships in social networks. The decision to work with web-based social networks is enforced by the fact that they form a large, publicly available dataset with tremendous interest from the general public. Chapter 2 specifically defines what qualifies as a web-based social network, and then presents the results of an exhaustive survey of websites. Over 133,000,000 user accounts spread across 127 websites were found, with subjects ranging from the deeply religious to the fringes of alternative sexual lifestyles. The description of the size and scope of websites is followed by a explanation of how users are able to add information to their social relationships. In fact, several large social networks already contain a notion of expressed trust between users, strengthening the choice to work with these datasets. Chapter 2 also introduces the Friend-Of-A-Friend(FOAF) Project, a Semantic-Web based technology that allows users to combine information about themselves from a variety of websites, and make statements about their friends even if those statements are not supported by the sites of which the user is a member.

Before making any computations with trust in social networks, it is vitally important to know what trust is and the properties it has. Within computer science, trust has been co-opted by many subfields to mean many different things. It is a descriptor of security and encryption (Kent, Atkinson, 1998); a name for authentication methods or digital signatures (Ansper, et al., 2001); a measure of the quality of a peer in P2P systems (Lee, et al., 2003); a factor in game theory (McCabe et al., 2003); a model for agent interactions (Jonker, Treur, 1999); a gauge of attack-resistance (Wallach, et al., 1997);a component of ubiquitous computing (Shankar, Abraugh, 2002); a foundation for interactions in agent systems (Maes, Kozierok, 1994; Barber, Kim, 2000); and a motivation for online interaction and recommender systems (Abdul-Rahman, Hailes, 1997). It is only in the last few years that the computing community has begun to look at the more social aspect of trust as a relationship between humans.

The difficulty of combining trust with algorithms and mathematical analyses is that trust is difficult to define, let alone to pin down in a quantifiable way. Intense theoretical analysis and complex models are one way to address this issue. However, the real boost to the union of these two topics has come in the form of web-based social networks that *force* people to quantify trust.

Beginning with work in the philosophical, sociological, and psychological communities, I present a definition that captures the nature of social trust relationships and also remains clear and simple enough to be used in social networks on the web. As is justified in chapter 3, the following definition of trust is used throughout this work: *Alice trusts Bob if she commits to an action based on a belief that Bob's future actions will lead to a good outcome*. This definition pervades this work. The first application is as a justification for the properties of trust that forms the foundation of the algorithms: transitivity, composability, and asymmetry.

These definitions allow for the creation of algorithms that utilize the trust networks. Chapters 5 and 6 present algorithms for working with networks where trust is expressed in binary values and over a continuous range, respectively. When a person has not expressed a trust rating for another person, the paths that connect them in a network can be used to infer how much one (the *source*) should trust the other (the *sink*). These chapters introduce the algorithms and present a theoretical and experimental analysis of their accuracy.

In chapter 5, I introduce two variations on an algorithm for inferring trust relationships in networks with binary trust ratings ("trusted" or "not trusted"). Using a statistical analysis and generated social networks for simulations, it is shown that these algorithms can produce highly accurate inferences about how much one person should trust another when as few as half of the trust ratings in the network agree with the source node's opinions. The relies on the binomial distribution created with the binary system. Chapter 6 moves to ratings over a continuous scale. Because the properties of the binary networks are not available to help increase accuracy, chapter 6 begins with an analysis of how trust rating and path length affect the agreement between a source and other nodes. This analysis is then used to develop an algorithm for calculating trust in networks with continuous ratings. Two naturally developed trust networks are used to show that my method is significantly better than several other algorithms taken from the literature.

To demonstrate the usefulness of these calculated ratings, two applications are introduced that personalize the users' experiences based on their trust ratings and social connections. The first is FilmTrust, a website that integrates social networks, trust, and movie ratings and reviews. Users build social connections to other people in the network. For each connection, they rate how much they trust their friend's opinion about movies. Users also rate movies and write reviews. The two are combined when a user visits a page about a movie. They are shown the average rating for the film and a personalized "recommended rating" calculated from the most trusted people who have rated the movie. As users' opinions deviate from the average, the personalized rating remains a relatively accurate estimation of their opinions. Reviews of the movies are also sorted according to the trustworthiness of the author so users see the most relevant ones first.

The second application is TrustMail, an email client that uses the calculated trust rating of each email's sender as a score for the message. Users are able to sort messages according to their trust value. Because ratings are drawn from an extended social network, they help users identify potentially important messages from otherwise unknown senders. The ratings can also be used for sorting spam folders so that improperly classified messages could be pulled out more easily.

Ultimately, trust and social preferences can be integrated into any number of potential applications. However, there is a larger application of this work. Social networks are only one type of complex system, and trust is only one type of relationship. The analysis in this work shows how an understanding of the properties of relationship-types within systems can lead to effective algorithms for understanding the relationships implicit in those systems. There are many other projects where these types of analyses can lead to results. Chapter 10 approaches this broader application with brief introductions to ongoing projects extending the work in this dissertation, as well as ecoinformatics, and information filtering.

This work is both the beginnings for future work and a case study. The applications presented here represent only a small portion of how the specific results of trust in social networks can be applied, and a still smaller fraction of how such techniques can be applied to complex systems analysis. This work should complement the growing body of work that is integrating social network analysis and trust into the user experience. Furthermore, the results here seem to suggest that a deeper understanding of the relationships in complex systems and methods for inferring information about them has the potential to lead to new discoveries in the social, biological, and physical sciences.

# Chapter 2

# Web-Based Social Networks

## *2.1 Introduction*

Web-based social networks (WBSN) have grown quickly in number and scope since the mid-1990s. They present an interesting challenge to traditional ways of thinking about social networks. First, they are large, living examples of social networks. It has rarely, if ever, been possible to look at an actual network of millions of people without using models to fill in or simulate most of the network. The problem of gathering social information about a large group of people has been a difficult one. With WBSNs, there are many networks with millions of users that need no generated data. These networks are also much more complex with respect to the types of relationships they allow. Information qualifying and quantifying aspects of the social connection between people is common in these systems. This means there is a potential for much richer analysis of the network.

As interest in social networking has grown, the term "social network" has become looser. Many sites promote themselves as social networks when they do not maintain any data that would be useful for a network analysis. This chapter presents a set of criteria for qualifying a system as a WBSN and another set for determining when information can be considered part of a relationship. Those principles guided an exhaustive survey of existing WBSNs followed by a discussion of trends in social network data sharing on the Semantic Web.

## *2.2 Previous Work*

This survey is motivated by the large body of work in social network analysis and in the study of online communities. While it would be impossible to cite all of the influential work related to social network analysis, the range of interest in the topic across nearly every academic field is impressive. These are just a few examples to give a sense of that scope.

Much of the foundational work in the analysis of social networks, and the major advances in the 20th century have been carried out in the fields of sociology, psychology, and communication (Barnes, 1972),(Wellman, 1982),(Wasserman & Faust, 1994). With a goal of understanding the function of relationships in social networks, and how they affect the social systems in which the networks exist, the research has been both theoretical and applied. Labor markets (Montgomery, 1991), public health (Cattell, 2001), and psychology (Pagel, et al., 1987) are just a few of the spaces where social network analysis has yielded interesting results.

In the last five to ten years, a new interest has developed in the structure and dynamics of social networks to complement the work already being done in social network theory. Though one of the first, and most popular papers in this area – Milgram's "Six Degrees of Separation" study (1967) – was conducted by a social scientist, the topic is of increasing interest to physical scientists. Their studies have addressed issues such as mathematical analyses of the structure of small world networks (Watts, 1999), community structure (Girvan, Newman, 2002), and how social network structure affects the spread of disease (Dezo, et al., 2002), (Jones, et al. 2003), (Newman, 2002).

As the web emerged, online communities and social networks supported by the internet became a source of interesting data. Garton, et al. (1997) presented an excellent introduction to how traditional methods of social network analysis could be applied to these online communities. Work in this space was also embraced by the interdisciplinary field of human-computer interaction, which produced interesting work on designing and supporting online communities (Preece, 2000), their application to problems such as collaborative filtering (Kautz, et al., 1997) and electronic commerce (Jung, Lee, 2000).

The promise of social networks on the web is that they offer new opportunities to researchers across the board. With network topologies that can be automatically extracted from the web, the social networks provide a new, large source of data for the more mathematical and structural types of analysis. At the same time, users are participating in rich social environments online while building these networks. That holds promise for scientists interested in the general function of social interactions, and because the contexts of these social networks is often very restricted (e.g. business networking among Asian-Americans), they can serve as a window into specific communities

## *2.3 Definitions*

There are many ways in which social networks can be automatically derived on the web: users connected through transactions in online auctions, users who post within the same thread on a news group or message board, or even members of groups listed in an

HTML document can be turned into a social network. Many online communities claim to be or support social networks, but lack some of the properties one may expect of a social network. This work uses a very specific definition. A web-based social network must meet the following criteria:

1. It is accessible over the web with a web browser. This excludes networks where users would need to download special software in order to participate and social networks based on other technologies, such as mobile devices.
2. Users must explicitly state their relationship with other people qua stating a relationship. Although social networks can be built from many different interactions, a WBSN is more than just a potential source of social network data; it is a website or framework that has the development of an explicit social network as a goal. This criteria rules out building social networks from auction transactions, co-postings, or similar events that link people when a connection is created as a side effect of another process.
3. The system must have explicit built-in support for users making these connections. The system should be specifically designed to support social network connections. This means that a group of friends who each maintain a simple HTML page with a list of his or her friends would not qualify as a WBSN because HTML itself does not have explicit built-in support for making social connections. There must be some greater over-arching and unifying structure that connects the data and regulates how it is presented and formatted.
4. Relationships must be visible and browsable. The data does not necessarily have to be public (i.e. visible by anyone on the web) but should be accessible to at least the

registered users of a system. Websites where users maintain completely closed lists of contacts are not interesting for their social networking properties – neither to users or people performing a network analysis – and are thus ignored for these purposes. For example, some websites allow users to bookmark the profiles of other users and others allow users to maintain address books. Even when these lists are explicit expressions of social connections, they would not qualify a system as a WBSN if they cannot be seen and browsed by other users. One important note here is that the system itself does not need built-in browsing support. Rather, each user's data must be made accessible with unambiguous pointers to each social connection.

These criteria qualify most of the major social networking websites like Tickle, Friendster, Orkut, and LinkedIn while ruling out many dating sites, like Match.com, and other online communities that connect users, such as Craig's List or MeetUp.com. Sites that require users to pay for membership are included as long as they meet the criteria above.

Within these social networks, users are often able to say more about their relationships than simply stating they exist. However, it is easy to confuse functionality of a WBSN with actual information about a relationship. Again, it is helpful to have a set of criteria that establish when an action or datum qualifies as information about a relationship in the social network.

1. A basic social networking connection between individuals must exist before additional information can be added. Sites that allow users to rate others, such as rating someone's appearance, often do not require that users have a connection –

anyone can rate anyone else. In order to be used as additional information about a relationship, there must be a relationship between people in the first place. Thus, simple rating systems that do not require users to be socially connected are not counted.

2. The information must be persistent. Many websites allow users to send messages or mini-messages (such as "winks" or "smiles"[1] on dating-related sites). Since these are sent and do not persist as a label on the relationship, they are not a piece of information about a relationship. On the other hand, comments or testimonials about a person do persist on the website and are considered as free text descriptions of a relationship.
3. The information must be visible and modifiable by the user who added it. At the same time, the information does not have to be publicly visible. Some data, like trust ratings, are personal and users would not want this shared with others.

### *2.4 A Survey of Web-based Social Networks*

The goal of this survey was to profile every social network on the web that met the criteria above. The number of users and primary purpose of each website, along with what additional relationship information they support, if any, was gathered from each website.

[1] "Winks" or "smiles" are usually sent by clicking an icon on a member's page. That member then receives a small message letting them know that someone sent them a "wink" or a "smile". These mimic their real-world counterparts in that they are small indications of interest without requiring the sender to say much.

This list grows daily and certain sites are not included because they are accessible by invitation only or in languages that could not be translated. An up to date list is maintained at http://trust.mindswap.org.

As of January 15, 2005 the survey encompassed 125 social networks with over 115 million members.

### 2.4.1 Size

The size of the networks varied greatly. Eighteen sites have over one million members, as shown in Table 2.1.

Table 2.1: Million-member WBSNs

| | Website | URL | Number of Members |
|---|---|---|---|
| 1 | Tickle | http://tickle.com | 18,000,000 |
| 2 | Friendster | http://friendster.com | 17,000,000 |
| 3 | Adult Friend Finder | http://adultfriendfinder.com | 15,700,000 |
| 4 | Black Planet | http://blackplanet.com | 14,000,000 |
| 5 | Hi5 | http://hi5.com | 6,100,000 |
| 6 | Asia Friend Finder | http://asiafriendfinder.com | 6,000,000 |
| 7 | My Space | http://myspace.com | 6,000,000 |
| 8 | LiveJournal | http://livejournal.com | 5,700,000 |
| 9 | Friend Finder | http://friendfinder.com | 3,600,000 |
| 10 | Amigos | http://amigos.com | 3,500,000 |
| 11 | Orkut | http://orkut.com | 3,000,000 |
| 12 | gradFinder | http://www.gradfinder.com/ | 3,000,000 |
| 13 | Alt.com | http://alt.com | 2,600,000 |
| 14 | LinkedIn | http://linkedin.com | 1,500,000 |
| 15 | Zero Degrees | http://www.zerodegrees.com/ | 1,300,000 |
| 16 | Out Personals | http://outpersonals.com | 1,050,000 |
| 17 | The Face Book | http://thefacebook.com | 1,000,000 |
| 18 | Fotolog | http://fotolog.net | 1,000,000 |

Figure 2.1 shows the membership of sites ranked according to size. There is an exponential decrease in the membership of the sites moving from the largest to the smallest. At the bottom of the membership list were sites with only a few hundred members.

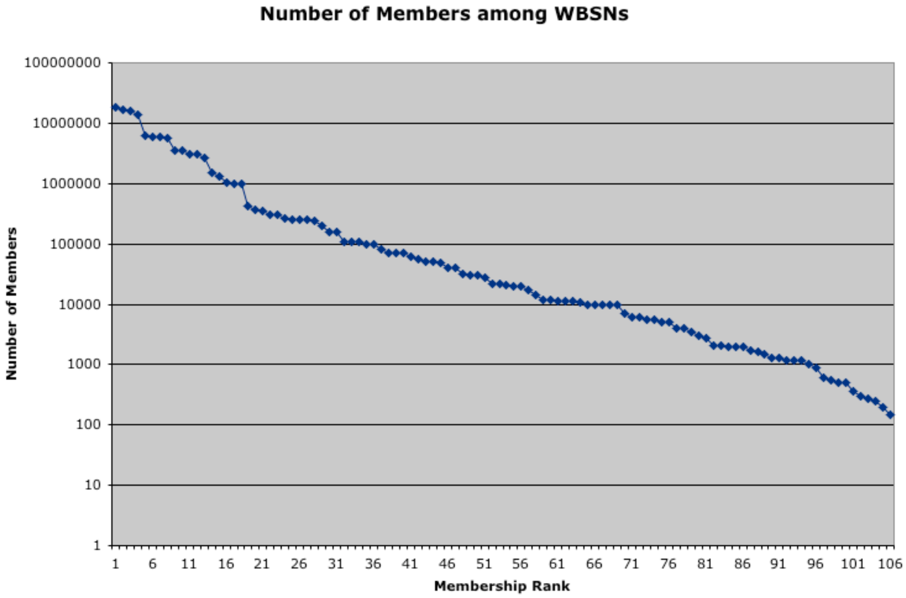


Figure 2.1: WBSN membership for sites ranked by population. Note that the y-axis is a logarithmic scale.

### 2.4.2 Categorization

With only a few exceptions, WBSNs fell into a small group of categories, shown in Table 2.2.

Table 2.2. Categories of WBSNs

| Purpose | Number of Sites | Number of Members |
|---|---:|---:|
| Blogging | 5 | 5,700,000 |
| Business | 16 | 3,300,000 |
| Dating | 23 | 49,000,000 |
| Pets | 6 | 80,000 |
| Photos | 7 | 1,015,000 |
| Religious | 11 | 650,000 |
| Social/Entertainment | 55 | 70,000,000 |

Some sites fall into multiple categories (e.g. there are several sites that are both "Religious" and "Dating" sites), so member and site totals in Table 2.2 add up to more than the overall numbers.

Perhaps it is not surprising that sex and love play prominent roles among these websites. The "Dating" group is second in number of sites and membership only to the more general "Social/Entertainment" category. Seven of the eighteen million-member sites list dating or personals as one of their explicit purposes. Two of the most explicit dating sites fall into the million-member club: Adult Friend Finder (a.k.a. Passion.com), "The World's Largest Sex & Swinger Personals site" with over 15.5 million members, and Alt.com, "the World's Largest Bondage, BDSM & Alternative Lifestyle Personals." At the same time, there is a continuum within these categories. At the opposite end of the spectrum is HotSaints.com, a site for single Mormons whose motto is "Chase and be chaste." Religion and romance were actually tightly coupled among sites surveyed; half of the "religious" sites stated dating or personals as one of their primary goals.

### 2.4.3 Relationship Data

One of the primary questions motivating this survey was to see how WBSNs allowed users to add information about their relationships. Of the 125 sites found, fifty-four had some method for describing relationships.

On twenty-nine sites, the only method of describing relationships was through free-text comments or testimonials. With the exception of LinkedIn (a business site), all of those were dating or social/entertainment sites where testimonials generally took the form of friends writing about their friends. A random sampling from some of these pages included the following comments. Names have been changed to protect users' privacy.

> User X is my absolute favorite Pittsburgher. I refuse to go home to Pittsburgh unless User X is there to ease the pain of the awful big-haired reality that Pittsburgh is. I love User X a ton and am always interested to see what this girl is up to.

---

> User Y, you can't have User Z. I love him too much. I too have hugged him and I never want to let go. He is my teddy bear and I want to have his babies. A little piece of me dies every time that I call and you don't answer. You know it was meant to be, you can't avoid the truth... Come back to me schnookums!!!!

---

> Well everyone..this is my OLDER sister User M...wat can i say about her? hmmmz..well first of all...shes scary even tho shes like a million times shorter than me! =P..and shes really really really EMBARASSING! and she says im boy crazy..YOU ARE ER! *wink wink* newaiz gonna go now..b4 u can read this @ home.. Bubiez...love ya er..see ya @ home....dont hurt me =)

These examples are fairly representative of the set of free-text testimonials out there. They are amusing and offer entertainment, but from a computational perspective they are not useful.

The other twenty-five sites that allowed users to describe relationships in a more restricted way. Twenty of them include options for users to categorize their relationships. Relationship types can be user-created labels in a few cases, but generally users choose from an enumerated list. Table 2.3 shows a few examples of these options.

These relationship types are much more useful when attempting to gain a deeper understanding of the dynamics within social network. Even when only a few options are offered, such as those from Naseeb seen in Table 2.3, the ability to approximately rank the strength of connections between people is greatly increased.

Other sites offer users the ability to rate aspects of their relationships on a numeric scale. Table 2.4 has a sampling of the features and rating scales available from some sites. From the perspective of someone performing a social network analysis, these numbers open up many new possibilities.

Table 2.3: A sampling of websites with options they offer for describing relationships.

| Website | URL | Relationship Type Options |
|---|---|---|
| Naboe | http://naboe.com | Friend, Lover, Neighbor, Brother, Sister, Cousin, Daughter, Son, Granddaughter, Grandson, Grandparent, In-law, Aunt, Uncle, Relative, Father, Mother, Spouse, Niece, Nephew, Employee, Business Associate, Co-worker, Boss, Vendor, Customer |
| Multiply | http://multiply.com | Family, Wife/Husband, Mother/ Father, Mother-in-law/Father-in-law, Daughter/ Son, Daughter-in-law/Son-in-law, Sister/Brother, Sister-in-law/ Brother-in-law, Grandmother/ Grandfather, Granddaughter/Grandson, Cousin, Second-cousin, Aunt/Uncle, Niece/Nephew, Step-mother/step-father, Step-sister/Step-brother, Step-daughter/Step-son, Ex-wife/Ex-husband, Friend of family, Distant relative, Other relative, Life partner |
| People Aggregator | http://peopleaggregator.com/ | Know of, Don't know of but want to, Know in passing, Know by reputation, Acquaintance of, Friend of, Close friend of, Relative |
| Naseeb | http://naseeb.com | Online Friend, Acquaintance, Friend, Good Friend, Best Friend |

Table 2.4: A sampling of websites that allow relationship features to be rated on a numeric scale.

| Website | URL | Relationship Characteristic | Rating Scale |
|---|---|---|---|
| Orkut | http://orkut.com | Trust | 0-3 |
| | | Sexy | 0-3 |
| | | Cool | 0-3 |
| Overstock | http://auctions.overstock.com | Business Rating | -2 - +2 |
| | | Personal Rating | 0-5 |
| RepCheck | http://repcheck.com | Social Trust | 0-5 |
| | | Business Trust | 0-5 |
| Trust Project | http://trust.mindswap.org | Trust | 1-10 |

Analysis begins with the graph structure of the social network and using the rating numbers as labels on the edges. It is then essential to understand the functional properties of the relationship characteristic. Knowing whether the characteristic is symmetric between individuals, if it is transitive or composable, and other such qualities lead to the types of algorithms and mathematical methods that could be used to gain a deeper understanding of the indirect relationships between people in the social networks.

## *2.5 The Semantic Web and Friend Of A Friend (FOAF)*

The 115,000,000 members of the social networks discovered in this survey do not represent 115,000,000 unique people; about one hundred accounts in that total belong to the author. In fact, many people maintain accounts at multiple social networking websites. It is desirable, for example, to keep information intended for business networking separate from information about dating. People's bosses or colleagues certainly do not need to know they you enjoy long walks on the beach. At the same time, users put significant effort into maintaining information on social networks. Multiple social network accounts are not just for compartmentalizing parts of their lives. A person may have one group of friends who prefer Orkut, another group on Friendster, like the quiz features of Tickle, and have an account on one or two religious websites to stay connected to that community.

At the same time, from the perspective of managing an entire set of social connections that are spread across sites, it is advantageous to merge all of those connections together into one set of data. In a merged social network, friends who have multiple accounts would be represented as a single person. Information about the user that is distributed across several sites also would be merged. The Friend-of-a-Friend (FOAF) Project (Dumbill, 2002) is a potential solution to sharing social networking data among sites, and this section introduces how that is being done.

### 2.5.1 Background

Rather than a website or a software package, FOAF is a framework for representing information about people and their social connections. The FOAF Vocabulary (Brickley, Miller, 2004) contains terms for describing personal information, membership in groups, and social connections. Table 2.5 lists the concepts and properties of the FOAF vocabulary. The property "knows" is used to create social links between people (i.e. one person *knows* another person).

Table 2.5: FOAF Vocabulary summary. More detail about each term and its use can be found at http://xmlns.com/foaf/0.1. The term "Person" and "knows" have been highlighted because they represent the basics needed to represent a social network.

| FOAF Basics | Personal Info | Online Accounts / IM |
|---|---|---|
| | | |
| Agent | weblog | OnlineAccount |
| Person | knows | OnlineChatAccount |
| name | interest | OnlineEcommerceAccount |
| nick | currentProject | OnlineGamingAccount |
| title | pastProject | holdsAccount |
| homepage | plan | accountServiceHomepage |
| mbox | based_near | accountName |
| mbox_sha1sum | workplaceHomepage | icqChatID |
| img | workInfoHomepage | msnChatID |
| depiction (depicts) | schoolHomepage | aimChatID |
| surname | topic_interest | jabberID |
| family_name | publications | yahooChatID |
| givenname | geekcode | |
| firstName | myersBriggs | |
| | dnaChecksum | |

| Projects and Groups | Documents and Images |
|---|---|
| | |
| Project | Document |
| Organization | Image |
| Group | PersonalProfileDocument |
| member | topic (page) |
| membershipClass | primaryTopic |
| fundedBy | tipjar |
| theme | sha1 |
| | made (maker) |
| | thumbnail |
| | logo |

The FOAF Vocabulary is represented as a Semantic Web ontology. The Semantic Web is an extension to the current web and is designed to encode information in a way that is machine readable. Like the current web of hypertext documents, Semantic Web information is maintained in documents stored on servers. Instead of using HTML, the Semantic Web uses a hierarchy of languages, including the Resource Description Framework (RDF) and Web Ontology Language (OWL). These languages are used to create ontologies, comprising classes (general categories of things) and their properties. The concepts from those ontologies are then used to describe data. There are several forms that data modeled with RDF and OWL can take. The examples presented here are shown in the N3 language. This shows the subject listed with each of its properties and their values.

In Table 2.5, terms with initial capital letters are classes, and terms in all lower-case are properties. A FOAF file will generally contain a Semantic Web-based description of at least one person with some personal information and who that person knows. The following code example  contains a simple FOAF description of a person

```
:Joe a foaf:Person;
     foaf:depiction <http://example.com/me.jpg>;
     foaf:firstname "Joe";
     foaf:lastname  "Blog";
     foaf:knows     :Dan,
                         :K,
                         :Pi.
```

From this snippet, a program that understands OWL and RDF will be able to process the information. Using the FOAF vocabulary , it can recognize that there is a person named "Joe Blog" with a picture online who knows Dan, Pi, and K.

The Semantic Web acts much like a large distributed database. There may be information about a person stored in many places. Using the basic features of RDF and OWL, it is easy to indicate that information about a person is contained in several documents on the web and provide links to those documents. Any tool that understands these languages will be able to take information from these distributed sources and create a single model of that person, merging the properties from the disparate sites.

### 2.5.2 FOAF and Current WBSNs

If a website builds FOAF profiles of its users, it allows the users to own their data in a new way. Instead of having their information locked in a proprietary database, they are able to share it and link it. Some WBSNs are already moving in this direction. Six of the sites in this survey generate FOAF files for each user, and they are shown in Table 2.6.

Table 2.6: WBSNs that provide FOAF profiles of users' social networks.

| Website | URL | Number of Members |
|---|---|---|
| LiveJournal | http://livejournal.com | 5,700,000 |
| eCademy | http://ecademy.com | 72,000 |
| Trust Project | http//trust.mindswap.org | 1,700 |
| Tribe | http://tribe.net | 250,000 |
| Buzznet | http://www.buzznet.com | 52,000 |
| Zopto | http://zopto.com | 10,500 |
| Total | | 6,086,200 |

With this information, a user with accounts on all of these sites can create a small document that points to the generated files. A FOAF tool would follow those links and compile all of the information into a single profile. The code example below shows a file that would link a person to the files maintained at each of the sites listed in Table 2.6.

```
:Joe  a foaf:Person;
rdfs:seeAlso
<http://trust.mindswap.org/trustFiles/385.owl>,
<http://www.livejournal.com/users/joeblog/data/foaf>,
<http://www.tribe.net/FOAF/6bed4755-a467-4fa9-844d-
e9bfc786e570>,
<http://ecademy.com/module.php?mod=network&op=foafrdf&uid=7
1343>,
<http://joe.buzznet.com/user/foaf.xml>
<http://www.zopto.com/foaf.asp?id=10088>;

     = <http://trust.mindswap.org/trustFiles/385.owl#me>.
```

These simple nine lines of code makes it possible to join potentially hundreds of pieces of information distributed across six sites together into one single description of the person.

Aside from the benefit to users who are able to merge their data, websites are also able to benefit from FOAF data on the web. For example, a website could suggest connections to other users in their system if FOAF data from another site shows a connection between the two people. Some user information could be pre-filled in if it is contained in a FOAF file somewhere else. By enhancing the user experience, a site becomes easier and more attractive to use.

### 2.5.3 Extensions to FOAF

While FOAF does have a long list of properties about people, many WBSNs have ways of describing people and relationships that are not part of the FOAF Vocabulary. One of the benefits of the Semantic Web is that ontologies and data can be extended by anyone, and thus it is easy to create properties that work with FOAF.

The Trust Project has created a Trust Module for FOAF that allows people to rate how much they trust one another on a scale from 1 – 10. Trust can be assigned in general or with respect to a particular topic. There is also FOAF Relationship Module (Davis,Vitiello, 2002) with over thirty terms for describing the relationships between people, including "lost contact with", "enemy of", "employed by", "spouse of", and others along those lines. A WBSN could co-opt these terms, or define its own set of relationship terms or personal characteristics to include in the FOAF data about its users.

## *2.6 Conclusion and Future Directions*

This survey of WBSNs was designed to provide a snapshot of the current state of web-based social networks, their number, size, and complexity. With this information, there are two clear fronts on which to progress: the computational and the analytical. The FOAF Project presented here is useful on both fronts in that it allows separate networks to be merged into one larger network model.

From the perspective of analysis, web-based social networks offer a look at a real living, evolving network. Users add, remove, and change connections frequently within these networks. The growth rate is exceptional, with larger sites gaining literally thousands of members each day. Tracking new members and their connections to the existing network at a regular interval would provide a window into how social networks grow and evolve. The information about relationships stored in many of these networks can provide an even deeper source of information, since the type of friends added to a person's network can be tracked as well as if and when those relationship types change.

Computationally, there are also tremendous opportunities. Particularly with information about relationships, there is space to develop new and useful algorithms for analyzing connections within the graph structure of the social network, making recommendations about indirect connections, and understanding the structure of relationships. Because many of the networks are open data sources, there is also the possibility of integrating users' social preferences into applications. This rich web-based data source will form the foundation of this work in personalization and social intelligence within software.

# Chapter 3

# Trust: Definition and Properties

Functioning societies rely heavily on trust among members (Fukuyama, 1998; Cook, 2001; Uslaner, 2002), and it is natural to expect the same to be true in online communities. If one is building a system online, there are several strategies for increasing the user's trust in the system and in its users overall (Shneiderman, 2000). In web-based social networks where users are making explicit statements about their trust relationships, the goal is not necessarily to build trust in the site or its members, but to make computations with the data that have already been made available. In human society, trust depends on a host of factors which cannot be easily modeled in a computational system. When deciding whether or not to trust a person, we are each influenced by past experiences with the person and with his or her friends, our opinions of actions the person has taken, our own predisposition to trust that is linked to psychological factors impacted by a lifetime of history and events (most of which are completely unrelated to the person we are deciding to trust or not trust), rumor, influence by others' opinions, and motives to profit by extending our trust, just to name a few. Putting a computationally usable notion of trust into social networks requires a clear, narrower definition of the term that still preserves the properties of trust with which we are familiar in our social lives.

For trust to be used as a rating between people in social networks, the definition must be focused and simplified. Individuals need a clear definition so they know how to describe their trust for others, and additional features of trust must be understood if trust relationships are to be used in computation.

### *3.1 A Definition of Trust*

Trust plays a role across many disciplines, including sociology, psychology, economics, political science, history, philosophy, and computer science. As such, work in each discipline has attempted to define the concept. The problem with defining trust is that there are many different types of trust and it means something different to each person, and potentially in each context where it is applied (Deutsch, 1973, Shapiro 1987).

Because the goal of this work is to perform computations with trust, it is natural to turn to the computer science literature. One of the most widely cited works is Marsh's PhD dissertation from the University of Stirling, "Formalising Trust as a Computational Concept" (1994). In this work, Marsh gives careful attention to many facets of trust, from the biological to the sociological, in order to develop a model for trust among agents interacting in a distributed way. His model is complex and highly theoretical. Aside from the difficulties with implementation, it is particularly inappropriate for use in social networks because his focus was on interacting agents that could maintain information about history and observed behaviors. In social networks, users assign a trust as a single rating describing their connection to others, without explicit context or history. Thus much of the information necessary for a system like Marsh's is missing. Furthermore, the intended application of this work is different, and that information is not necessary.

Web-based social networks are tools for the average web user. The definition of trust must be uncomplicated and straightforward enough that average web users understand what they are expressing, so they can express it accurately. For a simple, clear definition of trust, the sociological and psychological literature has more to offer.

Deutsch (1962) contains a frequently referenced definition of trust. He states that trusting behavior occurs when a person (say Alice) encounters a situation where she perceives an ambiguous path. The result of following the path can be good or bad, and the occurrence of the good or bad result is contingent on the action of another person (say Bob). Furthermore, the negative impact of the bad result is greater than the positive impact of the good result. This further motivates Alice to make the correct choice. If Alice chooses to go down the path, she has made a trusting choice. She trusts that Bob will take the steps necessary to ensure the good outcome. The requirement that the bad outcome must have greater negative implications than the good outcome has positive implications has been countered in other work (Golombiewski and McConkie, 1975), which does not always require disparity.

Sztompka (1999) presents and justifies a simple, general definition of trust similar to that of Deutsch: "Trust is a bet about the future contingent actions of others." There are two main components of this definition: belief and commitment. First, a person believes that the trusted person will act in a certain way. The belief alone, however, is not enough to say there is trust. Trust occurs when that belief is used as the foundation for making a commitment to a particular action. These two components are also present in the core of

Deutsch's definition: we commit to take the ambiguous path if we believe that the trusted person will take the action that will produce the good outcome.

Taking the main social aspects of the definitions above, the following definition is used in this work: *Alice trusts Bob if she commits to an action based on a belief that Bob's future actions will lead to a good outcome*.

The action of the trusted person and commitment by the truster do not have to be significant. If we are looking at trust in the context of movies, we can say Alice trusts Bob if she decides to see a movie (commits to an action) based on Bob's recommendation (based on her belief that Bob will not waste her time).

The justification for the belief component of the definition will vary from person to person. People may base their belief on pervious experiences in their own lives, a history of interacting with the person, or information gathered from an outside source.

One important note about trust is that it is not actually a single value. Take a specific topic, like movies. Users may be able to form a general opinion about how much they trust others about movies, but it would be more accurate and specific to break that trust down by genre. We may trust a friend about comedies, but not about dramas. However, that is not specific enough. Within each genre, trust could be broken down by period: we may trust someone's opinion of classic horror of the 50s and 60s, but not about modern horror films. There are an infinite number of ways that trust can be broken down, and when trust is used in a web-based social network, it is necessary to limit the complexity of the expression. Thus, the applications that use trust, including those I will present here, will always have a notion of trust that could be broken down more specifically. The nature of social networks requires it, and the results to follow will demonstrate that this approach seems to be sufficient to achieve good results.

This definition forms the foundation for explaining the properties of trust, identifying where trust exists in social networks, and how it can be used in computation.

## 3.2 Properties of Trust

### 3.2.1 Transitivity

The primary property of trust used in this work is *transitivity*. Trust is not perfectly transitive in the mathematical sense; that is, if Alice highly trusts Bob, and Bob highly trusts Chuck, it does not always and exactly follow that Alice will highly trust Chuck. There is, however, a notion that trust can be passed between people. When we ask a trusted friend for an opinion about a plumber, we are taking the friend's opinion and incorporating that to help form a preliminary opinion of the plumber. Generally, when encountering an unknown person, it is common for people to ask trusted friends for opinions about how much to trust this new person.

There are actually two types of trust one must express: trust in a person, and trust in the person's recommendations of other people. Alice may trust Bob to recommend a plumber, but not trust him at all to recommend other people whose opinion about plumbers is worth considering. Despite this dichotomy, in social networks it is preferable to let a single value represent both of these ideas. A single rating system is also more compatible with the traditional way users participate in social networks. Users are rarely, if ever, asked to express opinions of others with such subtle differences. As the following examples will show, the definition of trust supports a single value for both concepts.

The definition of trust supports the idea of transitivity. Recall that trust involves a belief that the trusted person will take an action that will produce a good outcome. To add context to the discussion, consider the aforementioned case of finding a plumber. If Alice asks Bob whether or not Chuck is a good plumber, she is going to use Bob's answer to support her action to use Chuck as a plumber or not because she believes Bob will give her information that will lead to a good plumbing outcome. Thus, if Bob says Alice should trust Chuck, Alice relies on her trust in Bob to develop some trust for Chuck. Bob's recommendation becomes a foundation for the belief component of Alice's new trust for Chuck. She will have some trust in him because, based on Bob's information, she believes he will take the steps necessary to produce a good outcome.

This same argument can be extended to longer chains of trust. In the situation above, perhaps Bob does not know about Chuck. Bob may ask a trusted friend (say Denise) about Chuck, and report back to Alice what the trusted friend has said. This adds a step in the chain: Alice->Bob->Denise->Chuck. Alice trusts Bob to give her information that will lead to a good outcome. Bob may decide the best way to give Alice that good information is to talk to his trusted friends, namely Denise. By the definition of trust, Bob trusts Denise because he expects her to give him good information about Chuck so he can obtain a good result (in this case, giving Alice reliable information). Since Alice trusts Bob to give her the good information, she can expect that the steps he goes through to obtain that information are also trustworthy. Thus, trust can be passed along a chain of trusting people. This logic also supports the use of a single value to represent trust in a person and trust in their recommendations about other people.

Because trust is not perfectly transitive, we could expect that it degrades along a chain of acquaintances. Alice is likely to have more trust in Chuck if Bob knows him directly and says he is trustworthy, than if a chain of people pass the information back to her. The computational aspect of this work will address how to compose those trust relationships down a chain to accurately how much the person asking for trust information (in this case, Alice) should trust the person at the end of the chain (Chuck). The precise method for propagating trust effectively is a computational issue, but the point here is to show that the definition of trust supports making these computations.

Computationally, this idea of propagating trust along chains of connections (thus exploiting some form of transitivity) has been widely studied and implemented (Gray, et al., 2003, Guha, Kumar, 2004, Jøsang, 1996, Jøsang et al., 2003, Richardson, et al., 2003, Ziegler, Lausen, 2004a).

### 3.2.2 Composability

Transitivity describes how a trust rating can be passed back through a chain of people. This is depicted in part (a) of Figure 3.1. Recommendations about the trustworthiness of an unknown person are used to support a belief about the actions of the unknown person, and thus lead to some amount of trust. What about the case in part (b) of Figure 3.1, where many people are making recommendations about how much to trust Chuck? In that situation, Alice must compose the information to decide whether or not to trust Chuck. This *composability* of trust is another important feature for making trust computations.

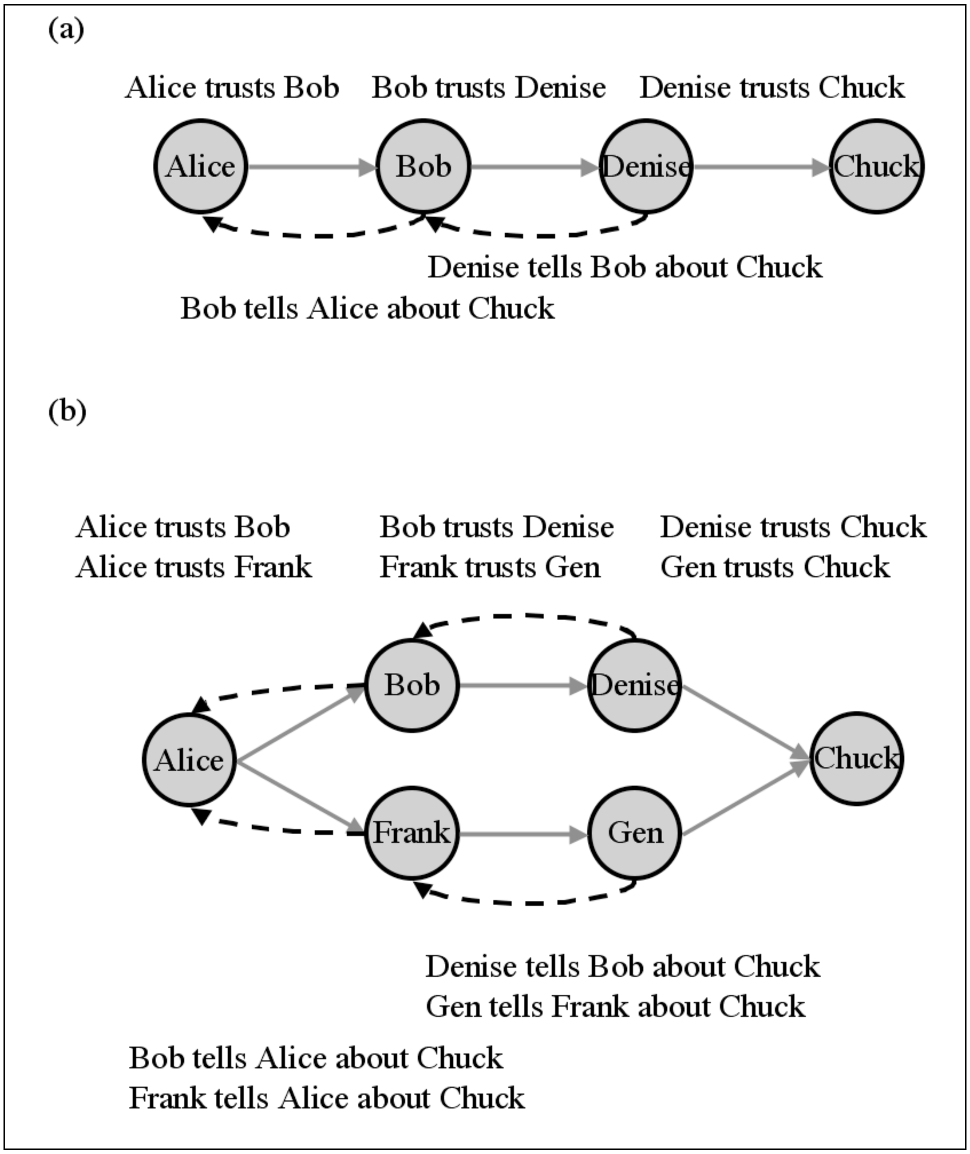


Figure 3.1: Network Path Structures for Finding Trust. Part (a) shows a simple chain of people where the transitive features of trust allow Alice to form an opinion of Chuck based on the information Denise gives to Bob and Bob, in turn, gives to Alice. Part (b) shows a more complex structure where Alice receives information from two people and she must come up with an opinion of Chuck by composing the information she has.

Composability makes sense if we look at trust recommendations as evidence used to support the belief component of trust. With information from many people, there is simply more reasoning and justification for the belief. Exactly how Alice should compose the trust values from many sources is another question. The trust values of each neighbor, and their recommendations about Chuck, all flow into a composition function that can vary from situation to situation and person to person. Richardson et al. (2003) used the concept of an openly defined composition function in their work. In later chapters, this work will use an analysis of the structure of trust relationships to produce a function that should return accurate results.

### 3.2.3 Personalization and Asymmetry

One property of trust that is important in social networks, and which has been frequently overlooked in the past, is the *personalization* of trust. Trust is inherently a personal opinion. Two people often have very different opinions about the trustworthiness of the same person. For an example, we need only look to politics. In the United States, when asked "do you trust the current President to effectively lead the country?" the population will be split – some will trust him very highly, and the others will have very little trust in his abilities.

The definition of trust includes a belief that the actions of the trusted person will lead to a *good* outcome. What qualifies as a good outcome varies from one person to another. What is a good outcome when the Red Sox are playing the Yankees? The answer depends strongly on where a person is from[2]. For a more immediate example, consider two sales people bidding on the same contract. What is the "good" action for the contract manager to take? Again, depending on which sales person is asked, the "good" action will be different. Since we all have interests, priorities, and opinions that can conflict with the interests, priorities, and opinions of others, when and how much we trust people will vary. Since there is rarely an absolute truth, a universal measure of the trustworthiness of a person is also rare. Calculations about trust must be made from the perspective of the individual to be of use to them and reflect their interests.

The *asymmetry* of trust is also important, and it reflects a specific type of personalization. For two people involved in a relationship, trust is not necessarily identical in both directions. Because individuals have different experiences, psychological backgrounds, and histories, it is understandable why two people may trust each other different amounts. For example, parents and children clearly trust one another at different levels, since the children are not capable of many tasks. This strong asymmetry can occur in other relationships where the people involved are on close social levels. This can be carried out fully to "one-way trust" where circumstances force one person to trust the other, but there is no reciprocal trust (Hardin, 2002; Cook, 2001). However, most asymmetry is not as extreme as any of those circumstances. Most trust is mutual (Hardin, 2002) in that each party has some trust for the other, but there are still often differences in how much they trust one another. For example, employees typically say they trust their supervisors more than the supervisors trust the employees. This is

[2] Actually, I assert that it's always better for the Red Sox to win. There *are* some absolute truths in this world.

seen in a variety of hierarchies (Yaniv, Kleinberger, 2000). Asymmetric trust can arise in any relationship, and representations of trust relationships in models of social networks must allow for these differences.

### *3.3 The Values of Trust*

Trust is information about a social relationship and, as such, in a web-based social network it must be represented as a label on that relationship. There is still much freedom as to what form that label takes, and this section addresses some of the possible options for the values representing trust.

In the survey of social networks presented previously, six social networks allow users to express trust in one way or another. One of them – eCademy – uses the simplest possible representation of trust. Users have two options: do not make any statement about trust, or state that a friend is "trusted". This does not allow for any range of trustworthiness or an expression of untrustworthiness. It simply lets users indicate which people they trust.

There are some types of relationships that easily fit in this paradigm of simply existing or not existing. For example, whether or not we are related to a person, if we have met a person, or if we are co-workers, is a relationship that exists or does not exist. Trust, however, is not this simple. It is generally established that social trust has a range of strength (Gambetta, 2000, Marsh, 1992, Marsh 1994). Five WBSNs, shown in Table 3.1, have some notion of trust that is expressed over a range of values. Overstock.com auctions has been included in this list, although the their "Business Rating and "Personal Rating" are not explicitly ratings of trust. Ratings in the context of business are similar to trust in that they provide information about how much one can trust a person to produce a good outcome with respect to a business transaction.

Table 3.1: The range of values available for rating trust in web-based social networks.

| Website | URL | Relationship | Trust Values |
|---|---|---|---|
| Overstock Auctions | http://auctions.overstock.com | Business Rating | -2 - +2 |
| | | Personal Rating | 0-5 |
| Orkut | http://orkut.com | Trust | 0-3 |
| RepCheck | http://repcheck.com | Business Trust | 0-5 |
| | | Personal Trust | 0-5 |
| The Trust Project and FilmTrust | http://trust.mindswap.org<br>http://trust.mindswap.org/FilmTrust | Trust | 1-10 |

There are other schemes for representing levels of trust, including scales with more values (such as Richardson et al., (2003) that used a continuous 0-1 range) or with labels rather than numbers (e.g. "very low trust," "low trust," "moderate trust," "high trust," and "very high trust"). While there are no web-based social networks currently using them, there are other possibilities for creating trust information. For example, ranking systems as opposed to explicit ratings could be used. These could be combined with preference elicitation mechanisms (Keeney, Raiffa, 1976; Boutilier, et al., 1997) to build a profile of user's trust. However, the direct rating adopted by all web-based social networks generally puts less burden on the user and extracts that information more quickly. The algorithms presented in chapters 4-6 are designed to work with the standard, explicit rating of trust.

### *3.4 Conclusions*

This chapter presents a formalization of trust for use in computations in web-based social networks. Beginning with sociological and psychological background, I present a definition of social trust tailored for use in web-based social networks. The functional properties of trust follow from that definition. Computing with trust in social networks is a burgeoning area of research, and the results of this chapter contribute a foundation to tie the algorithmic aspects of this work in with the nature of trust.

# Chapter 4

# Inferring Trust: Background and Related Work

With an understanding of web based social networks and how trust functions within those networks, the goal of this work is to develop algorithms for inferring a trust value from one person to another when there is not a direct link between them in the network.

In this chapter, I introduce the fundamental elements of how trust will be calculated within social networks. The exact algorithms are presented in chapters 5 and 6, but this discussion shows how the properties of trust and social networks from chapters 2 and 3 relate to and justify the basic elements of the algorithms. This is followed by a description of the major trust algorithms currently found in the literature, and some applications that rely upon trust.

## *4.1 From Trust Properties to Trust Algorithms*

Given a social network, information about trust can be provided to users in many ways. The goal is generally the same: recommend to one node how much to trust another node in the network. The way this is done varies (see section 4.2). The goal of the algorithms in this dissertation is to recommend what trust rating one person might want to give another, unknown person if there were a connection. The trust recommendations are very much like predictive recommendations made by a recommender system.

Fitting an algorithm to the task of inferring trust values must be guided by the properties of trust. Those properties, as presented in Chapter 3, are transitivity, composability, and personalization.

Trust algorithms can be divided into *global* and *local* algorithms. Global algorithms compute a universal trust value for each person in the network. Regardless of who asks for a trust recommendation, the same answer is given. On the other hand, local trust algorithms calculate trust from the perspective of the person asking for the trust recommendation. Essentially, the results are personalized for each user.

Because trust is personal, and beliefs vary between two people, personalization (through a local algorithm) should improve the accuracy of the results. If a person wants a recommendation about how much to trust the President, an algorithm that simply composes *all* of the values in the system can be expected to give an answer that falls almost directly in between "very low trust" and "very high trust". Since most people have an opinion that leans toward high trust or low trust, this middle rating will not mean much. It reflects the opinion of the population, and is not a recommendation to the *individual*. Our algorithm is based on the perspective of the user. It looks at friends who the user trusts about their opinions on a topic, the people whom those friends trust, and so on. Thus, the opinions of people whom the user does not trust much are given very little consideration, and the opinions of people whom the user trusts highly are given more consideration.

Figure 4.1 depicts a sample social network. The solid lines indicate relationships with trust, and the dashed lines indicate no trust. The node for which we are determining a trust rating, called the *sink*, is trusted by two nodes (8 and 9) and not trusted by two nodes (6 and 7). If a trust rating for the sink were calculated by averaging all of the direct ratings of the sink, every node would get the same recommendation. However, if we take into account the information that we know about the structure of the network from the perspective of each node, a much more informative recommendation can be made.

This will rely on the properties of transitivity and composability. Transitivity allows information to be passed along paths back to the sink, and composability means that the source can combine information from many sources. Just how trust is passed along paths and composed is left out here and will be presented in the next two chapters. Here, I present only the structural fundamentals of how the process will work.

In the sample network in Figure 4.1, Node 1 can choose to accept information only from it's trusted neighbors. In the end, only the trust ratings given by Nodes 6 and 7 will propagate back to Node 1. Both 6 and 7 do not trust the sink, and only their opinion will be passed back to Node 3 and then to Node 1 who will calculate that the sink is not to be trusted. Similarly, Node 2 also only considers trusted paths. At the end of those paths, Nodes 8 and 9 both have directly rated the sink to be trustworthy. Their values are passed back along the network paths through Nodes 4 and 5 to Node 2. Node 2 will conclude that the sink is to be trusted Thus, if perspective is taken into account, Node 1 and Node 2 can each receive relevant and accurate information about how much to trust the sink, even though their opinions are diametrically opposed and the information in the network is mixed.

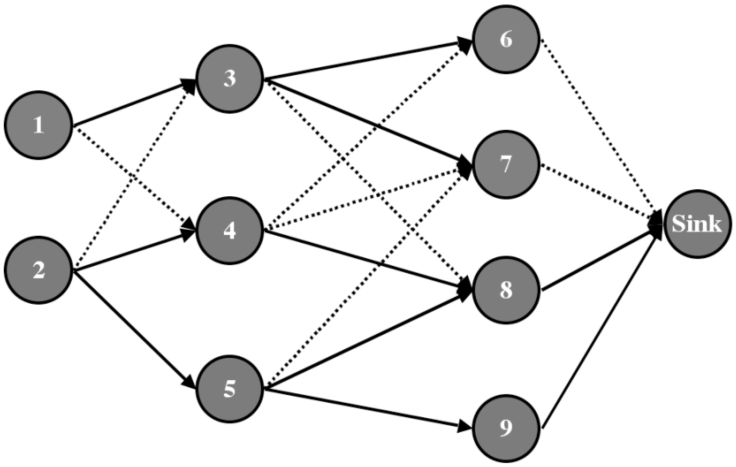


Figure 4.1: Finding Trusted Paths to the Sink. Nodes consider ratings from the people they trust highly (indicated by solid edges). Nodes with low trust ratings (indicated with dashed edges) are only considered when they are a direct rating of the sink, but are not used in finding paths to the sink. The ratings made by trusted nodes that directly rated the sink are used in coming up with a recommendation about how much the source should trust the sink.

This knowledge about personalization will guide the development of a trust inference algorithm and the properties of transitivity and composability will form its core.

### *4.2 Previous work*

Determining trust for an unknown entity is  a common problem, even when it is not directly related to social networks. In this section, I present related work on calculating trust with an eye toward how the work relates to and influences the algorithms that will appear in the following chapters.

### 4.2.1 Game Theory

Although game theory does not play a large role in this work, it makes up one of the largest bodies of work on understanding and predicting trust. It would be remiss not to mention some game theoretic background here.

Game theorists, and particularly those interested in iterated games (games where players meet many times), are interested in how players can maximize their strategies based on their knowledge of the previous actions of others. When a game is repeated, reciprocity becomes an important part of trust.

There are many reciprocity strategies proposed by game theoreticians. The most widely known of these is the Tit-For-Tat strategy that has been extensively studied in the context of the Prisoner's Dilemma game (Axelrod, 1984; Pollock and Dugatkin, 1992; Nowak and Sigmund, 2000, Golbeck, 2002). The Prisoner’s Dilemma is a decision model. Anecdotally described, two people, indicated here as player 1 and player 2, are arrested for espionage and placed in separate interrogation rooms. Each player has the option to cooperate with his peer, or to defect against him. Based on what each player does, a payoff is awarded (see Table 4.1). When players play the game many times (the *iterated* Prisoner's Dilemma), they gain information about the past actions of the other player, and develop a strategy based on this. If both players trust one another to cooperate, they earn higher scores. The Tit-For-Tat strategy states that in iteration *n*, a player takes the action taken by the opponent in iteration *n-1*. If the opponent cooperated in the previous round, the player will cooperate in the current round. If the opponent defected, then the player will defect.

Table 4.1: Payoff Matrix for the Prisoner's Dilemma. The payoffs are listed as (player 1, player 2).

| | | Player 1 | |
|---|---|---|---|
| | | Cooperate | Defect |
| Player 2 | Cooperate | 3,3 | 5,0 |
| | Defect | 0,5 | 1,1 |

When deciding whether or not to cooperate, it is a benefit to know that your partner in a game is willing to participate in reciprocative strategies. This leads to cooperation which, in turn, leads to better outcomes. When deciding whether or not to trust another person, trustworthy individuals tend to trust others who have a reputation for being trustworthy and eschew those with weaker reputations (Cosmides and Tooby, 1992). Developing a reputation as someone who is trustworthy is an asset because trust affects how willing people are to participate in reciprocative interactions (Dasgupta, 2000; Tadelis, 1999), which in turn lead to higher payoffs.

In game theory, learning about the reciprocativeness of a person, and, as a result, determining the trustworthiness of an individual, is largely based on looking at past behavior. Using game theoretic principles and models as a background, Miu et al. (2002) proposed a model for  calculating trust and reputation scores in social networks. A deeper analysis of to trust in social networks from the game theory perspective is covered in Buskens (2002).

The problem with game theoretic models is not their accuracy; indeed, knowledge about historical interactions can lead to very accurate assumptions about the trustworthiness of an individual. Rather, the game theoretic approaches rely on that information about past behavior. That is available in some social networks, but in the web-based social networks that form the foundation of this dissertation, it is not. The algorithms used here rely on *explicit statements* of how much one individual trusts another, with essentially no knowledge of why that trust was developed, because that is how relationship information is made available in the medium of WBSNs.

### 4.2.2 Peer-to-Peer Systems

Trust is also an important issue in peer-to-peer (P2P) file sharing network. P2P systems are similar to social networks in that each peer is connected to the peers with which it has interacted, and this is a subset of the total number of peers.

Some work (Nejdl, et al., 2004) has adopted the term to describe access control policies; an agent would be trusted to  access information if it can provide information showing that it meets the requirements set forth in the access control policy. However, the applications of trust that are more relevant to this work are those that use it for reputation management. For a P2P system to work, each node must correctly implement the network protocols and provide access to uncorrupted files. If a node is not reliable, it can degrade the usefulness of the entire network. Thus, the "trustworthiness" of a node can be measured as how well it participates in the P2P network.

Several projects have addressed the issue of inferring trust for an unknown node in P2P systems, so bad nodes can be filtered out of the active group of peers.

The EigenTrust algorithm (Kamvar et al., 2003) considers trust as a function of corrupt vs. valid files that the node provides. A peer maintains information about the trustworthiness of peers with which it has interacted based on the proportion of good files it has received from that peer. For one peer to determine the trustworthiness of another with which it has not interacted, it needs to gather information from the network and infer the trustworthiness. The EigenTrust algorithm calculates trust with a variation on the PageRank algorithm (Page et al., 1998), used by Google for rating the relevance of web pages to a search. A peer creates a direct trust rating for another peer based on its historical performance. In its simple form, the algorithm uses a matrix representation of the trust values within the system and over a series of iterations it converges to a globally accepted trust rating of each peer. Because of safeguards built into the system, EigenTrust has been shown to be highly resistant to attack.

Another approach to reputation and trust management in Peer-2-Peer systems determines trust through the past behavior of a peer. The focus of that work is on how to share trust assessments in a distributed way (Aberer and Despotovic 2001, Lee et al., 2003).

There is a fundamental difference between trust in P2P networks and trust in social networks. P2P trust is based on the reliability of a node to adhere to absolute correct parameters. A file is either corrupted or it is not. There is not a "sort of corrupted" file. A node properly implements a protocol or it does not. Again, there is no in-between. In social networks, though, trust is not based on this absolute truth. Two people may hold vastly different opinions about a topic (look only to religion or politics for extreme examples), and there is no absolute truth to determine which one should be trusted and which one should not. A person decides how much to trust another based on personal opinion. The universal truth in P2P networks is a benefit for making trust calculations because the information provided by one peer reflects the truth of all peers. The need for a trust rating personalized to each node is minimized because one peer can expect to have the same experience as every other peer.

### 4.2.3 Calculating Trust on the Web

On the web, "trust" has largely been an issue of security, authentication, and digital signatures. However, work has also focused on using the more social aspects of trust.

Advogato is a website, at http://advogato.org, that serves as a community discussion board and resource for free software developers. It also is the testbed for Raph Levin's trust metrics research (Levin, 1998). Each user on the site has a single trust rating calculated from the perspective of designated *seeds* (authoritative nodes). Trust calculations are made using a network flow model. His metric composes certifications between members to determine the trust level of a person, and thus their membership within a group. Users can be certified at three levels: apprentice, journeyer, and master. Access to post and edit website information is controlled by these certifications. Like EigenTrust, the Advogato metric is quite attack resistant. By identifying individual nodes as "bad" and finding any nodes that certify the "bad" nodes, the metric cuts out an unreliable portion of the network. Calculations are based primarily on the good nodes, so the network as a whole remains secure. Because of its use of groups to determine who can post messages, Advogato is called a *group trust metric*. It is also a global trust algorithm because the same seeds are used to make calculations for every user. A common alteration to Advogato sets the user as the single seed, thus converting it to a local metric with personalized calculations.

Ziegler and Lausen (2004a) propose a trust algorithm called Appleseed. Like Advogato, it is a group trust metric. However, instead of using maximum flow, the basic intuition is motivated by spreading activation strategies. Like EigenTrust, Appleseed is based on finding the principal eigenvector. It is a local trust metric, and given a network and a source it returns a ranking of all the nodes in the network.

Richardson et al.(2003) use social networks with trust to calculate the belief a user may have in a statement. This is done by finding paths (either through enumeration or probabilistic methods) from the source to any node which represents an opinion of the statement in question, concatenating trust values along the paths to come up with the recommended belief in the statement for that path, and aggregating those values to come up with a final trust value for the statement. Current social network systems on the Web, however, primarily focus on trust values between one user to another, and thus their aggregation function would require some modification to be applied in these systems. Their paper intentionally does not define a specific concatenation function for calculating trust between individuals, opting instead to present a general framework as their main result. To test their algorithms, they do choose a concatenation function (multiplication) and show accuracy results using the Epinions network. Grishchenko (2004) discusses some issues and potential applications related to the work from Richardson and others.

One problem that arises in algorithms that are based on finding the principal eigenvector, like Kamvar (2003), Zeigler and Lausen (2004), and Richardson et al., (2003), is that trust must first be normalized to work within the matrix. This means that the normalized trust value from a person who has made many trust ratings will be lower than  if only one or two people had been rated. However, socially, trust is not a finite resource; it is possible to have very high trust for a large number of people, and that trust is not any weaker than the trust held by a person who only trusts one or two others.

Some work has also looked at whether *distrust* can be propagated and inferred like trust (Guha, et al. 2003). They convert continuous ratings to binary values representing trust and distrust. They observed a relatively low error rate in their calculations. In Chapter 5, Inferring Trust in Binary Trust Networks, I presents some theoretical foundations that speak to why the binary system may be partially responsible for these results.

It is worth mentioning that Richardson, et al. (2003) and  Guha, et al. (2004) both used Epinions as a testbed network. This network does not qualify as a web-based social network based on the definition given in Chapter 2 because the trust ratings are not an expression of the trust in a social relationship. The Epinions web of trust has no expectation that a social relationship exists between trusted people, but simply that one person values the opinions of another. Even without this unsatisfied criterion, there are other issues[3] with the Epinions network that make it less than optimal for the type of analysis presented in this dissertation.

[3] These include the fact that distrust is not displayed, so the public network is only made up of "trust" relationships or no relationships.

### 4.2.4 Public Key Infrastructure

Work in the Public Key Infrastructure (PKI) uses trust in a similar way to social networks. Mapping a name to a public key or, conversely, to finding the public key of a particular user is an important task for executing secure transactions. When there is no centralized authority to map keys and names, this sort of authentication can be done by combining information from a path of authorities. These chains can suffer if any of the intermediate authorities have poor information. Trust values between authorities can be combined over paths to determine confidence in the authority at the end point. While the concept of trust in these systems does not match exactly with the definition presented in chapter 3, the approaches to combining trust are very similar. Metrics for calculating trust over paths have been presented in Tarah and Huitema (1992), Beth et al. (1994), Mendes and Huitema (1995), Maurer (1996), and Reiter and Stubblebine (1998) to name a few.

The inputs and outputs of these metrics vary, and some do not translate well to use in social networks. For example, Maurer (1996) makes calculations using several features in addition to trust. Confidence ratings are combined with explicit statements of trust and authenticity measures to infer authenticity information. On the other hand, some metrics can be directly applied to trust in social networks. Beth, Borcherding, and Klein's metric (1994) is more directly applicable to web-based social networks. It takes a network with trust values on a [0,1] scale, a source, and a sink as input, and produces a calculated trust value as output. In chapter 6, their algorithm will be compared with TidalTrust, my algorithm for inferring trust values.

### *4.3 Conclusions*

With an understanding of social networks on the web and the definition and functional properties of trust, we can move toward creating algorithms to infer trust relationships in web-based social networks. That first requires an understanding of how the properties of trust translate into algorithms. In this chapter, I have given insights into how personalization guides the decision to create a local algorithm, as opposed to a global one, and provided a general overview of how trust will need to be calculated over paths and composed into a single value.

Making calculations about trust are important in many spaces, within and outside of social networks. I presented an overview of the trust-related literature in game theory, peer-to-peer networks, web-based social networks, and the public key infrastructure, and described where lessons can be borrowed for inclusion in the algorithms that will be presented in the next chapters, and when information needed in other areas of research prevents a direct translation to use in web-based social networks.

With this information as a background, it is possible to start developing algorithms for computing trust. The next two chapters will introduce two algorithms for making these calculations in networks with binary and continuous trust values.

# Chapter 5

# Inferring Trust in Binary Trust Networks

There is a wide range of values that can be used for rating how much one person trusts another. The simplest of these rating schemes is a voting system; a person would rate someone as either "trusted" or "not trusted". While many of the subtleties of trust are lost with such a coarse rating system, it also simplifies the analysis of the algorithms. As such, it is a good starting place for developing the foundations of algorithms that will be used with finer grained rating systems.

This chapter presents two variations on a simple algorithm for inferring trust in binary-valued trust networks. The accuracy of the algorithms, determined by how frequently the inferred values agree with the actual trust values assigned by a node, are analyzed both theoretically and in simulation. The results show that in this system, the algorithms described here can produce highly accurate results. This will serve as the foundation for eventually developing algorithms that work with more complex trust ratings.

## *5.1 Generating Social Networks*

To test algorithms for inferring trust, there must be networks for testing. Naturally occurring networks take a long time to develop, and the topological properties are fixed. To experiment with making trust inferences on networks with various properties, it is necessary to be able to automatically generate network models.

### 5.1.1 Building Networks with Correct Topology

It has been widely documented that social networks have the properties of small world networks (Watts, 1999). The properties of graph structure that define a small world network are connectance and average path length. Connectance (indicated by the variable $\gamma$) is the property of clustering in neighborhoods: given a node $n$, connectance is the fraction of edges between neighbors of $n$ that actually exist compared to the total number of possible edges. Small world graphs have strong connectance. The average shortest path length between nodes(indicated with variable $L$) grows logarithmically with the size of the graph in small world networks.

The one difference between the networks in this research and traditional complex systems is that our network has directed edges. Although the $L$ and $\gamma$ values are usually calculated with undirected edges, they can easily be calculated with directed edges. The shortest path is calculated by following edges and respecting their direction. Connectance is calculated with twice as many possible edges, since any pair of nodes has two possible directed edges than can connect them.

The work by Watts and Strogatz (Watts, Strogatz, 1998) showed that graphs with small world properties can be generated by randomly rewiring a small number of nodes in a regular graph, like a lattice. The variable $p$ indicates what percentage of edges should be randomly selected, removed, and randomly reconnected As *p* increases the average path length drops off quickly. The average connectance, on the other hand, remains high until *p* gets too large. Creating a lattice and choosing a *p* that produces a graph with high connectance and low average path length will produce a small world graph. This model, called the ß-model (Watts, 1999) has been shown to successfully emulate the structure of several common social networks, including the co-authorship graph and co-actor graph (Davis et al., 2003; Foster et al., 1963; Newman, 2001;Watts, 1999).

We used a the ß-model to create graphs for use in the analysis of the accuracy of algorithms that are presented in section 4. The *p* value varied depending on the size and average degree of our graphs. We verified for each graph size and average degree that the *p* value produced graphs with *L* and *γ* values consistent with small world graphs.

### 5.1.2 Adding Trust Ratings to Graphs

The edges in the generated graphs represent connections between individuals. To generate a trust network, those edges must be augmented with values representing the trust relationship between individuals. This section describes the process of adding trust ratings into a generated graph.

One node in the network is randomly chosen as the source. We then give each remaining node a "true" rating. This rating says whether the node is trustworthy (good) or not trustworthy (bad) according to the source. The number of good and bad nodes are assigned at a pre-determined ratio, and have two properties. First, this true value is treated as *the source's opinion* of the node, as though an all-knowing oracle could tell whether or not the source would consider this node was good or bad if there were an established relationship. The second property determined by the good/bad rating is the node's behavior. Good nodes agree with the source with a certain probability, while bad nodes always vote incorrectly; the bad nodes will say every good node is bad and every bad node is good.

It is important to note here that we are not studying the *behavior* of the nodes in the network to try to identify "good" or "bad" nodes. Clearly, with bad nodes always voting opposite the source, they would be relatively easy to track down. The bad nodes represent attackers – nodes that may be incorrectly called trustworthy, and can then corrupt the system. While the behavior in these simulations – *always* assigning incorrect values – is worse than we would expect from an attacker in an actual network, it allows us to perform a worst-case analysis of our algorithm since the true value allows us to determine if our inference was correct or incorrect.

Once each node has been given its true value, the trust ratings on the edges are assigned. Bad nodes rate every neighbor opposite its true value. Good nodes rate each neighbor correctly with a certain probability. For example, if we specify that the good nodes are accurate 70% of the time, each neighbor is rated independently with a rating corresponding to the true value with probability 0.7

## *5.2 Making Trust Inferences*

In developing the networks described in the previous section, several properties have been identified: the trust value from one node to another, "good" and "bad" nodes, the notion of a "true" value for a node from the perspective of another, and the accuracy of ratings. In this section, two variations on a simple algorithm are introduced. A statistical analysis of their performance is complemented by an experimental analysis on generated social networks. These results show that in reasonable conditions the algorithms are able to produce accurate inferences, often improving on the initial accuracy in the system.

### 5.2.1 A Rounding Algorithm

In this algorithm, the source polls each of the neighbors to which it has given a positive reputation rating. Neighbors with zero ("no trust") ratings are ignored, since their reputation means that they give unreliable information. Recall that a zero, or "no trust" rating does not mean *distrust*; it simply means that information from the non-trusted person cannot be relied upon to be accurate. Since we do not want information from nodes that are not trusted, paths through these nodes are not included. Only paths through trusted neighbors are considered.

Each of the source's trusted neighbors will return their rating for the sink. The source will then average these ratings and round the final value. This rounded value is the inferred reputation rating from source to sink.

Each of the source's neighbors will use this same process to come up with their reputation ratings for the sink– if there is a direct edge connecting them to the sink, the value of that edge is used; otherwise, the value is inferred. As shown in Figure 5.1, if a node is encountered in two paths from source to sink, it is considered in each path. Node B and C will both return ratings calculated through D. When a reputation rating from D is first requested, D will average the ratings from E and F. The value is then cached at D, so that the second time D's reputation rating is needed, no calculations are necessary.

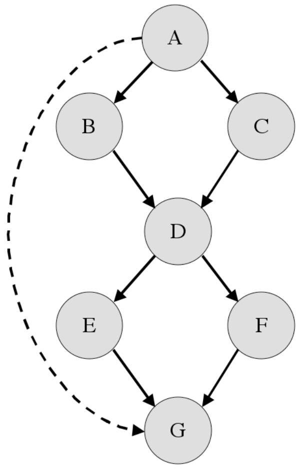


Figure 5.1: An illustration of how nodes are used in the inference from node A to node G

### 5.2.2 Non-Rounding Algorithm

We altered the algorithm presented above by removing the rounding performed by each node before it returns a value. The only rounding is made in one final step, added to the end of the algorithm, where the original source rounds the average of the values returned by its neighbors, so the final inferred value is 0 or 1. Ratings are still assigned as 1 or 0 values (trusted or not trusted). With the algorithmic change, however, intermediate nodes on the path from source to sink return values in the range of [0,1] instead of returning rounded {0,1} values. Accuracy was determined by taking the difference between the rounded final inferred value and the true value.

### 5.2.3 Analysis of the Algorithms

There are two variables in the network – percentage of good nodes, $g$, and the accuracy, $p_a$, of good nodes. When a node gives an inaccurate rating, it may rate a good node as "not trusted", or rate a bad node as "trusted". This causes a problem because good nodes may be ignored, since information from nodes with zero-ratings is ignored, or, more profoundly, a large amount of incorrect information can be introduced into the calculations when a bad node is incorrectly rated as "trusted".

The overall accuracy of the direct ratings initially assigned in the network is given by $g*p_a$. We call this the *initial accuracy* in the network and represent this with the variable $a$. Figure 5.2 illustrates the initial accuracy as these two parameters are varied. Note that the initial accuracy is a measure of how frequently the direct ratings of people in the network agree with the true value *according to the source*. It is not a measure of how accurate a person's ratings are with respect to their own beliefs.

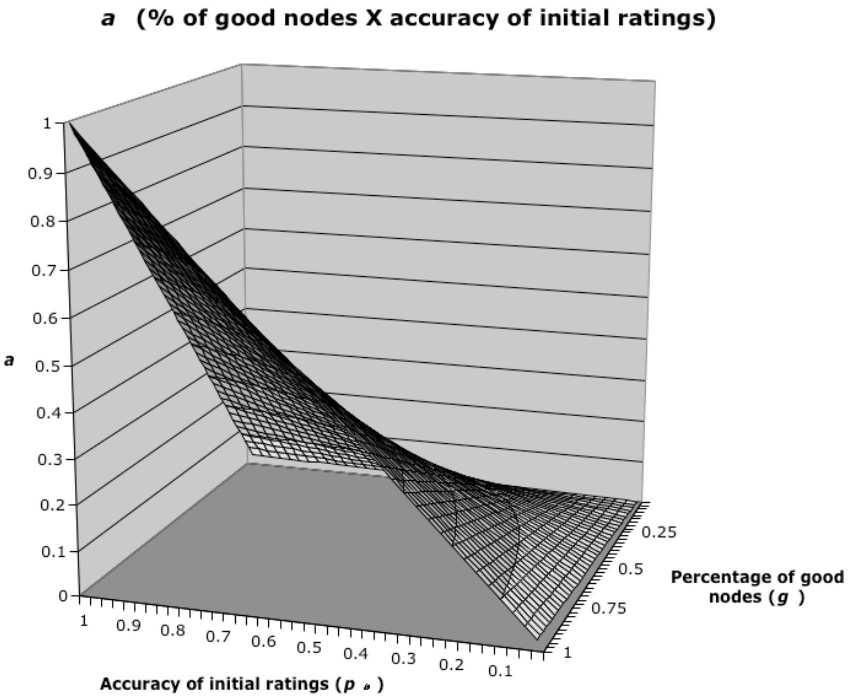


Figure 5.2: A map of how the initial accuracy in the system changes with $g$ and $p_a$.

By design in these algorithms, a node will make a correct inference if the majority of its neighbors return the correct rating for the sink. Since the bad nodes are always incorrect, the accuracy of the good nodes must compensate to obtain a correct inference from a majority vote. Thus, to obtain a correct inference, the initial accuracy must be at least 0.5.

Let $a = g * p_a$. For a given graph with $n$ nodes, the probability that the majority of the nodes will correctly rate the sink is given by a binomial distribution.

$$\sum_{i=\left\lceil \frac{n}{2} \right\rceil}^{n} \binom{n}{i} a^i (1-a)^{n-i}$$

**1**)

The binomial distribution can be approximated by a normal distribution with a mean centered at $a$. The Central Limit Theorem says that as $n$ increases, the binomial distribution gets closer and closer to a normal distribution. That is, the binomial probability of any event approaches the normal probability of the same event. As $n$ increases, the standard deviation of the normal distribution decreases, making a narrower curve, and thus the probability that a majority of nodes make correct recommendations is closer to the mean $a$. Thus, for $a > 0.5$, the probability that the mean is greater than 0.5, and ergo the inference is correct approaches 1. Similarly, for $a$<0.5, the probability of a correct inference goes to 0.

$$\lim_{n \to \infty} \sum_{i=\left\lceil \frac{n}{2} \right\rceil}^{n} \binom{n}{i} a^i (1-a)^{n-i} \to 1 \qquad \text{for a>0.5} \tag{2}$$

Thus, if nodes are accurate at least half of the time, the probability that the recommendation is correct goes to 1. This is a critical point. As long as $g * p_a$ is greater than half, we can expect to have a highly accurate inference.

This analysis describes one step of rounding. With the non-rounding algorithm, where only the final inferred value is rounded, this analysis applies. In the rounding algorithm, where the average trust value is rounded at each step, the accuracy increases at each step. As the algorithm moves up from the immediate neighbors of the sink toward the source, *a* will vary from node to node, but it will increase at each level. Figure 5.3 illustrates this point where the network starts with all good nodes, accurate in 70% of their classifications. After moving up three levels from the sink, the accuracy of the inference will be approximately 96%.

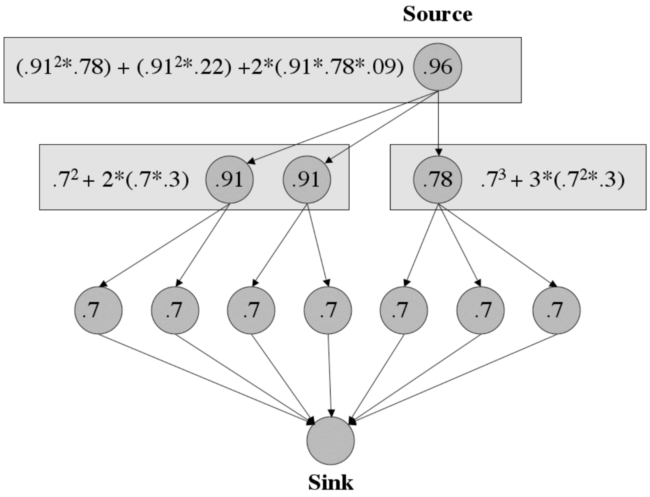


Figure 5.3: The increasing probability of a correct trust inference. This figure shows a simple network and demonstrates the increasing probability of accuracy. Beginning with a uniform accuracy of 0.7, the probability of properly classifying the sink increases as we move up the search tree from sink to source. After only three levels, this source has a 96% chance of properly classifying the sink.

This analysis suggests that the rounding algorithm will outperform the non-rounding algorithm. The non-rounding algorithm is an important intermediate step between a system with binary ratings and a system with continuous values. Since continuous values are used in the intermediate steps between source and sink, this algorithm nearly replicates what would be used when users assign values in a broader range. The analysis here not only shows that the final-step rounding gives good results, but also that accuracy is not lost when the internal rounding is eliminated. Future work will address the shift to a system with continuous values and how a slight variation on the non-rounding algorithm can be effective in such a network.

### 5.2.4 Simulations

When inferring the trust rating for a node, the accuracy of the inference is determined by comparing the inferred value to the true value. The simulations presented in this section support the theoretical analysis presented in section 4.3; for both the rounding and non-rounding algorithms, the inferred trust rating is more accurate than the accuracy of the initial trust ratings in the network.

Starting at 0.025 and using increments of 0.025, there are 1,600 pairs $(g, p_a)$. For each $(g, p_a)$ pair we generated 1,000 small-world graphs using the ß-model described above. In those graphs, the source and sink were randomly chosen. The trust inference from source to sink made on each graph was checked against the true value of the sink. This experiment was repeated for graphs with 400, 800, and 1600 nodes. Similar results were found for each graph size.

Beginning with the rounding algorithm, experiments showed that the accuracy of the inferred rating was significantly higher than then initial accuracy in the network from the good nodes ($p_a$) and the percentage of good nodes ($g$). Figure 5.4 shows data for the inferred accuracy using the rounding algorithm on a set of graphs with 400 nodes and an average degree of 16. While the initial accuracy of ratings decreases linearly, the accuracy of the inferred ratings remains higher.

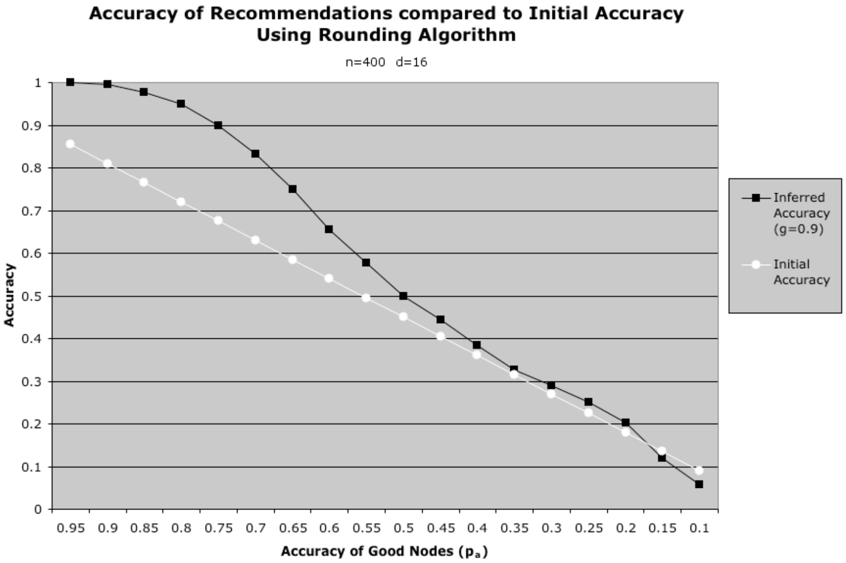


Figure 5.4: A comparison of the initial accuracy of trust ratings with the accuracy of inferred ratings using the rounding algorithm for $n$=400, $d$=16, $g$=0.9, and a variable $p_a$.

In most simulations, the accuracy of a recommendation remains high relative to the initial accuracy until $p_a*g$ nears 0.5. This is shown in Figure 5.5.

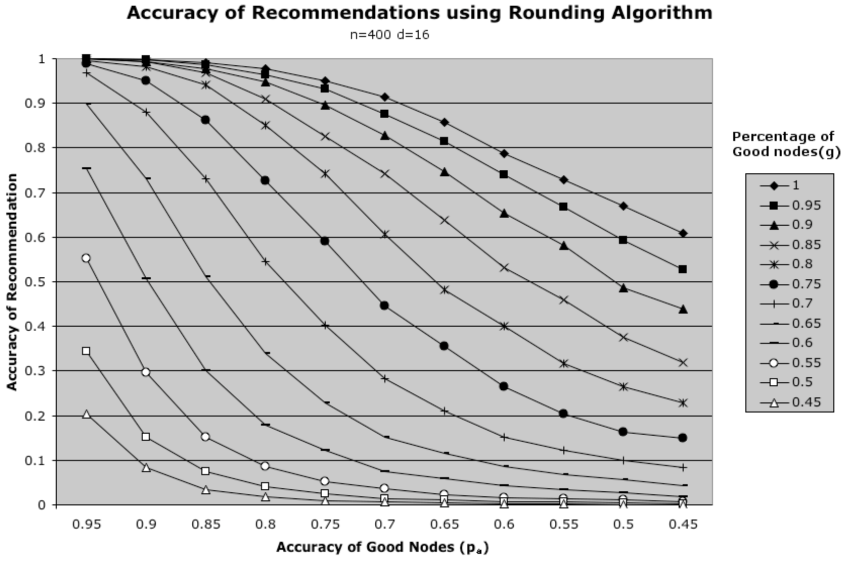


Figure 5.5: The accuracy of inferred ratings are shown for various initial percentages of good nodes.

The non-rounding algorithm produced results inline with the theoretical analysis. Even without the intermediate rounding, the inferred values were more accurate than the initial accuracy in the system. The results are less dramatic than for the rounding algorithm, but Figures 5.6 and 5.7 shows that the increased accuracy is still present.

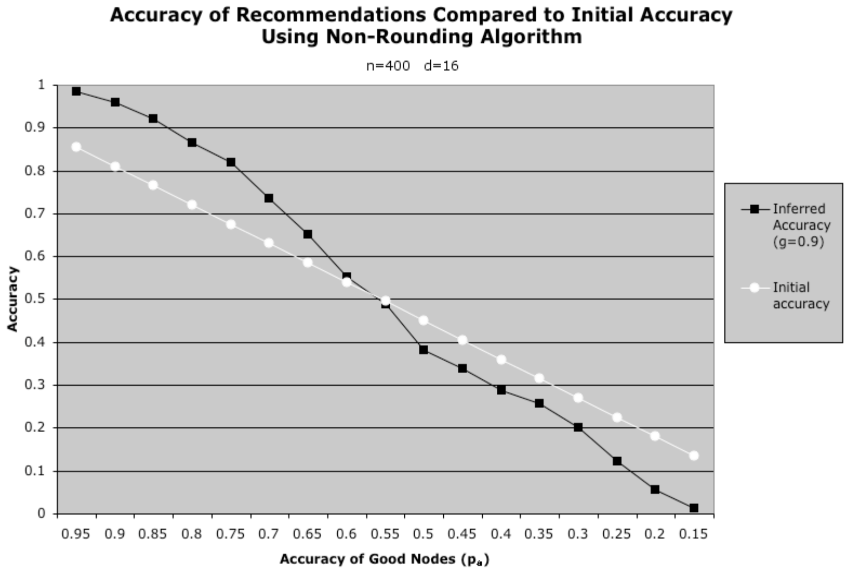


Figure 5.6: Accuracy of Recommendations Compared to Initial Accuracy Using Non-Rounding Algorithm. This figure shows that for *a*>0.5, the inferred accuracy using the non-rounding algorithm is higher than the accuracy of the initial ratings. This is the same effect seen in Figure 5.4 for the rounding algorithm.

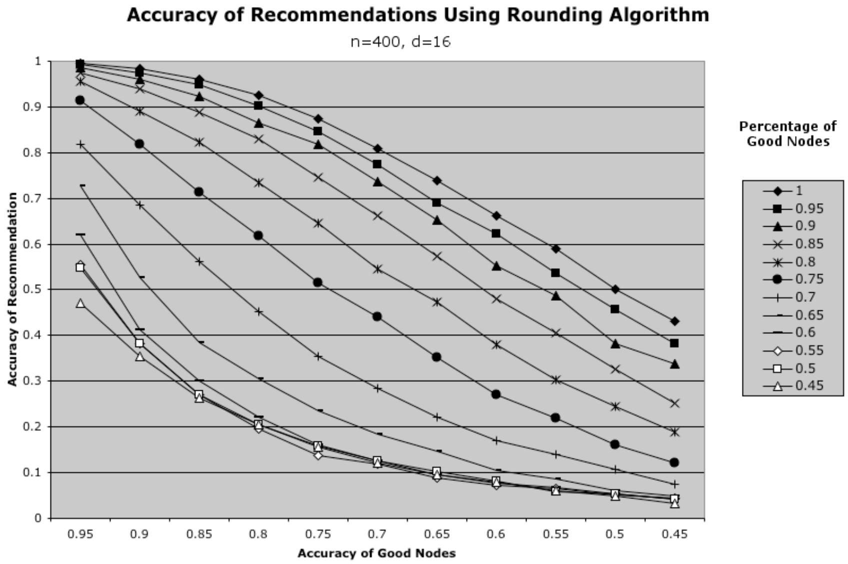


Figure 5.7: Accuracy of Recommendations Using Rounding Algorithm. This figure shows the accuracy of inferred trust values using the non-rounding algorithm for a range

of *g* and $p_a$ values. Though less pronounced than the results for the rounding algorithm shown in Figure 5.5, we can see that the results follow a similar pattern of remaining higher than the *a* value for *a*>0.5 and lower for *a*<0.5.

To directly compare the two algorithms, Figure 5.8 shows the accuracy of both the rounding and non-rounding algorithms together for *g=1* and *g*=0.9 with $p_a$ >0.5.

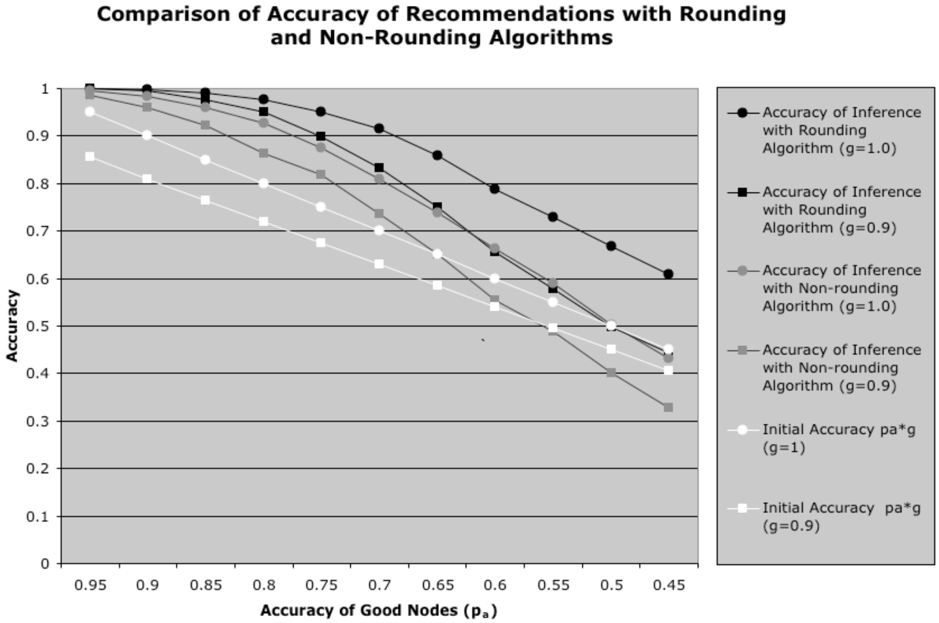


Figure 5.8: A comparison of the accuracy of trust inferences made with the rounding and non-rounding algorithms.

The results in Figure 5.8 are drawn from simulations on 400 node networks with average degree of sixteen. The performance follows a similar pattern for both algorithms and both *g* values. As the theoretical results would indicate, the rounding algorithm outperforms the non-rounding algorithm for both *g* values because the rounding at each step removes more error than is rounded out in the final step of the non-rounding algorithm.

### *5.3 Conclusions*

The analysis and experimental results show that the algorithms presented in this chapter can produce accurate trust inferences when the initial accuracy in the social network is greater than 0.5. These algorithms could be directly applied in social networks that use binary trust ratings, such as AllConsuming.net or Epinions. They also provide a foundation for extending the algorithms to continuous rating systems, which is presented in Chapter 6.

# Chapter 6

# Inferring Trust in Continuous Trust Networks: TidalTrust

Although some systems may choose to use a binary trust rating system, a larger range of values more accurately reflects the nuances of social trust. Most web-based social networks that implement trust ratings use a range of values. To move from an algorithm that works well in a binary rating system to one that works well with a more continuous set of ratings, an understanding of how these values are patterned in actual social networks is necessary. This chapter presents two naturally developed social networks with trust ratings with an analysis of how trust behaves in that network. This information is used to ground the development of new functions for composing trust values to replace the simpler algorithms used in the previous chapter. An algorithm for calculating these trust recommendations in networks with continuous values, called TidalTrust, is described and analyzed in detail in section 6.3.

### *6.1 Experimental Networks*

The quality of trust inference algorithms in this work is determined by measuring their accuracy. To do that, it is necessary to have real networks on which to test the data. As part of the trust-related projects in this work, two separate trust networks have been grown from scratch.

The first network is part of the Trust Project at http://trust.mindswap.org/. This network is built up from distributed data maintained on the Semantic Web. Within their FOAF files, users include trust ratings for people they know using the FOAF Trust Module, a simple ontology for expressing trust developed as part of this project. The ontology has vocabulary for rating people on a scale of 1 (low trust) to 10 (high trust). These ratings can be made in general or with respect to a specific topic. In the network built up for study in this research, users assigned general ratings to one another.

Because the network is built up from files that are distributed across the web, the size varies depending on which files can be accessed at the time a model is built. On average, there are about 2,000 members with over 2,500 connections. Figure 6.1 shows the current structure of this network.

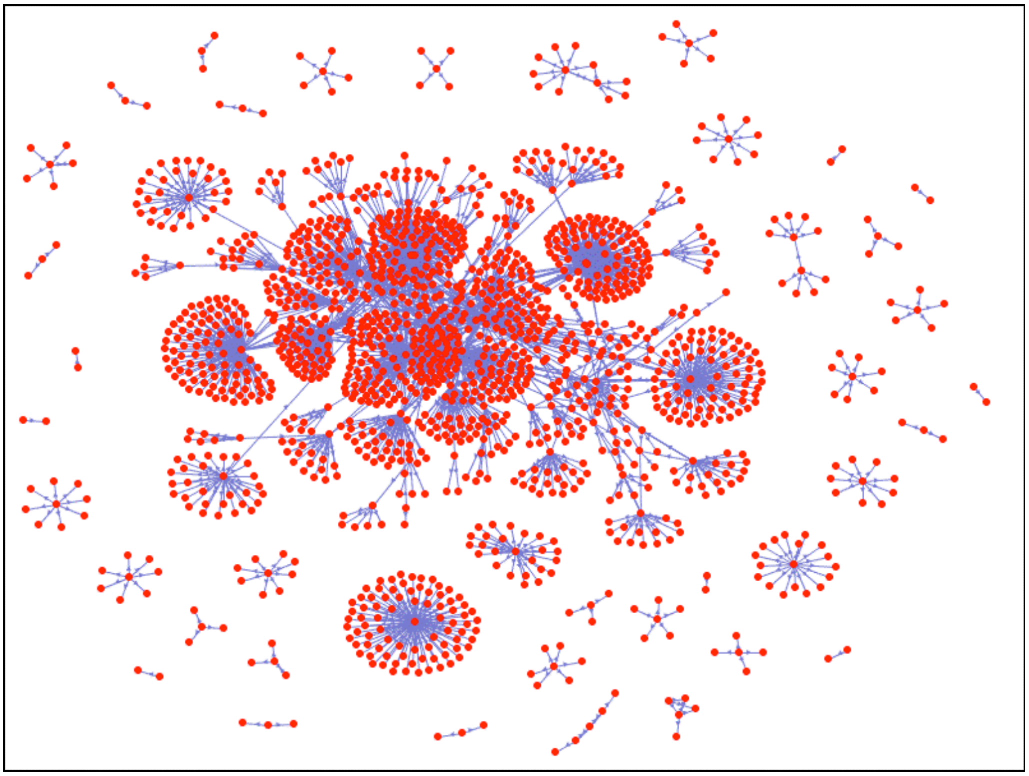


Figure 6.1: The structure of the Trust Project's network.

The second network is part of the FilmTrust project, a website that combines social networks with a movie ratings and reviews site. More information about the network is described in Chapter 7.

The site currently comprises 300 members who have rated each others' trustworthiness on a scale of 1-10. Figure 6.2 illustrates the topology of the network.

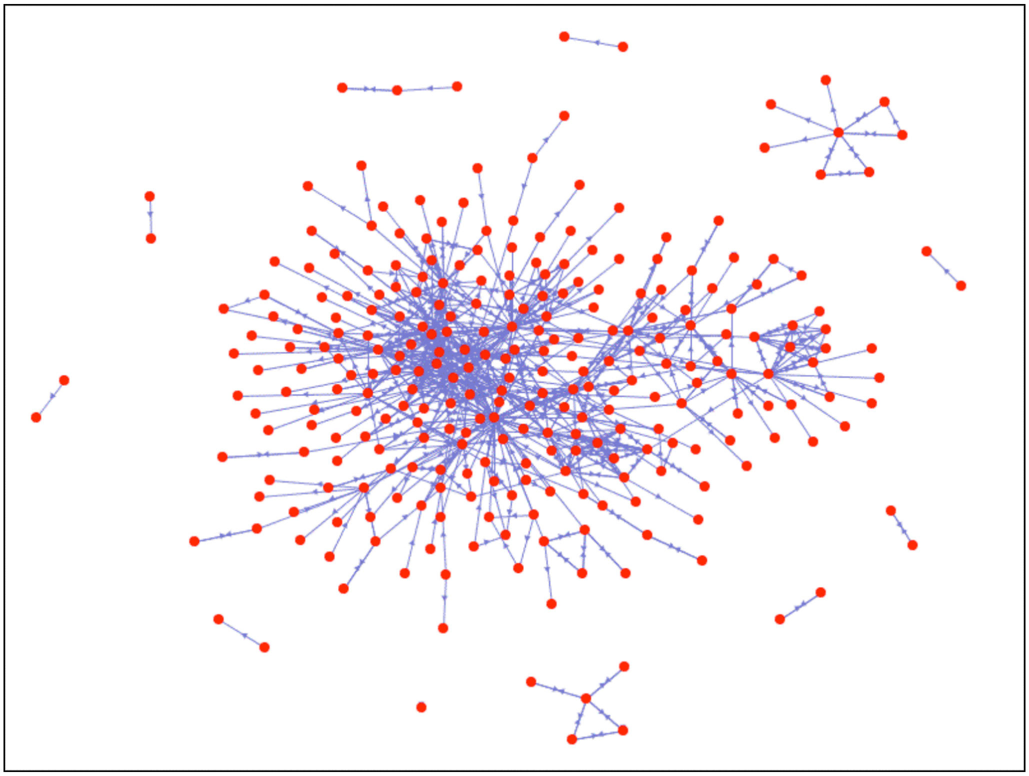


Figure 6.2: The structure of the FilmTrust social network.

In this network, users rate how much they trust people about movies. Since *all* of the ratings are specific to movies, they can be composed in the same way as the generalized ratings of the Trust Project. This would not be possible if some trust ratings were assigned in general and others were specific to movies, because the arguments about transitivity and composability would not hold.

Both of these networks use a scale of 1-10, which comprises 10 discrete values, rather than a continuous range of values. This is an artifact of the human interface. The average web user is more comfortable working in a scale like this as opposed to assigning ratings, say, in the range of 0 to 1. These discrete values also make it easier to categorize and analyze results. While it is not totally continuous, it offers a range of values that approximates a continuous system closely enough that we do not expect it will affect the results.

## *6.2 Patterns of Trust Values*

In the previous chapter, two variations on an algorithm for inferring trust were introduced. Some of the choices in this algorithm were simple to make.

- Composing Values: Because the ratings were binary, neighbors were either trusted or not trusted. Information from the untrusted neighbors could just be ignored, since it is not logical to incorporate information from an untrusted source. On the other hand, all of the trusted neighbors are trusted equally, so all of their information can be accepted equally. Since there is nothing to distinguish one trusted neighbor from another, each neighbor's information is given equal weight when composing the values into an average.
- Transitivity and Paths in the Network: The social network models in the previous chapter were uniform in how likely it was that a good node would give a rating that agreed with the source node. This essentially took the worst case for accuracy and applied it uniformly in the network. Nodes that were closer to the source were not considered to be any more accurate than nodes far from the source. This worst-case analysis was easier to conduct, and still yielded good results. If it were the case that nodes closer to the source were more accurate, the quality of the results would only improve.

When working with a continuous scale, these facets of the algorithm need to be adjusted in a way that will ensure the accuracy of the results. This requires an analysis of the structure and trust topology of the network. The results in this section are primarily drawn from the larger Trust Project network.

### 6.2.1 Distribution of Trust Values

When discussing binary trust networks, the variable *g* indicated the proportion of trusted nodes in the network. When using a continuous scale of trust values, there is a distribution that measures the percentage of nodes for a given trust value.

In the networks used for these results, users employed a 1-10 scale with discrete integer values. As such, there is not a smooth distribution. Although one might expect the average trust rating to be 5, the middle value, this turns out not to be the case. The average trust rating is 7.25, with a standard deviation of 2.30 in the Trust Project network (used in the following examples), and 6.8 with a standard deviation of 2.1 in the FilmTrust network . Figure 6.3 shows the full distribution of trust values in the Trust Project network. In the network, there were 2,231 ratings assigned. The distribution has been scaled to show what proportion of the total number was comprised by each individual rating.

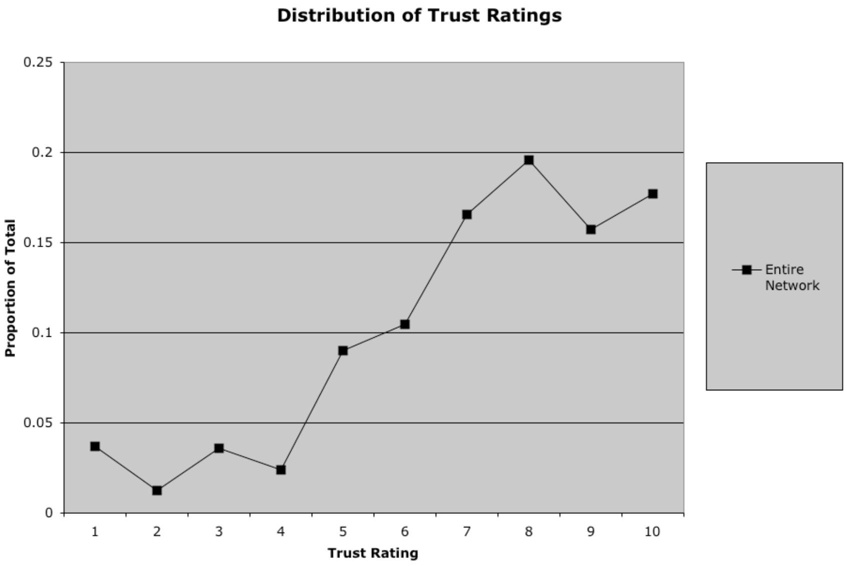


Figure 6.3: The distribution of trust ratings in the Trust Project network.

The distribution of the trust ratings is skewed toward the higher values which is logical in the social context. Users choose the people with whom they form social connections, and it is reasonable that people are more likely to connect with people they trust highly than people for whom they have little trust.

### 6.2.2 Correlation of Trust and Accuracy

The networks used in the previous chapter were designed to have untrusted nodes behave in a way that was incongruous with the opinions of the source node, and trusted nodes to behave reliability with some probability. In networks with continuous ratings, it is less obvious how trust ratings relate to accuracy.

To investigate this, experiments were performed on the Trust Project network. The goal was to ascertain if individuals with higher trust ratings were more likely to be accurate. This was determined repeating the following process for each node. First, a node was chosen as the source. For each neighbor of the source, $n_i$, a list of common neighbors of the source and $n_i$ was compiled. For each of those common neighbors, the difference between the source's rating and $n_i$'s rating was recorded as a measure of accuracy. A smaller difference means a higher accuracy. This difference was recorded along with the source's rating of $n_i$. Figure 6.4 illustrates one step in the process.

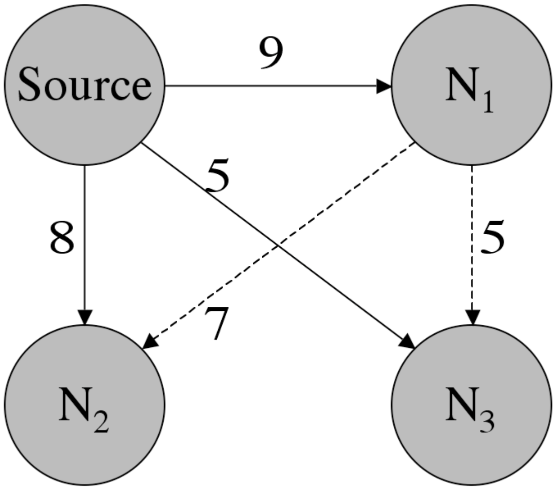


Figure 6.4: Finding points of comparison in the network. In these experiments, this network would produce two data points: the difference between the source and $N_1$'s ratings of $N_2$ (in this case, 1) and the difference between the source and $N_1$'s ratings of $N_3$ (in this case, 0)

These experiments produced a pair of numbers for each data point: the trust value from the source to its neighbor ($n_1$), and the difference between the ratings of a common neighbor. The number of data points for each trust value indicates how frequently common neighbors are shared between pairs of nodes at each trust level.

As shown in figure 6.5, the frequency of common neighbors among pairs of nodes with high trust levels is much higher than the frequency of those ratings in the original network. These indicate that people with stronger trust connections share more common social connections than is proportional. In fact, over 40% of the common neighbors were found between nodes that shared a high trust rating. If the experimental results show that the results are more *accurate* when there is more trust, this distribution means that a the increased accuracy will be reinforced by the increased frequency of common neighbors among pairs with high trust.

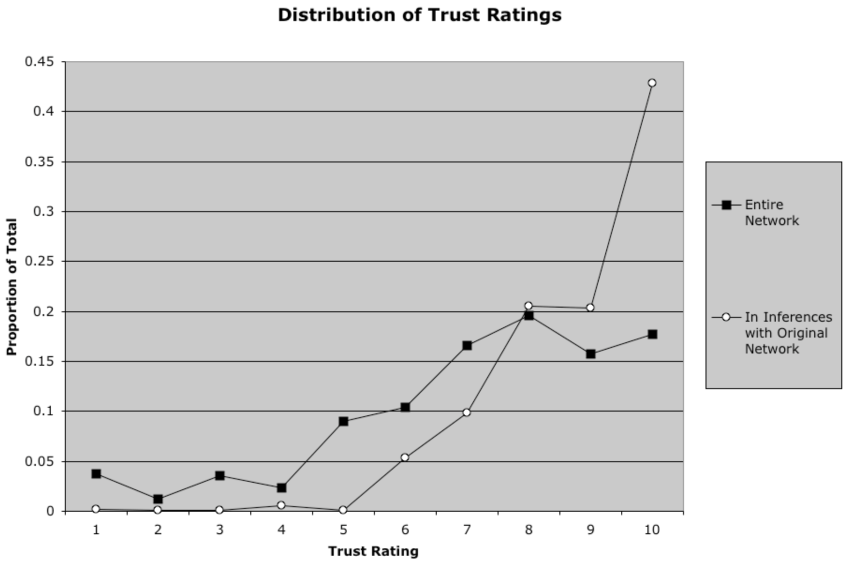


Figure 6.5: Distribution of trust ratings in the original network, and in the data points of the experiments.

If individuals with higher trust ratings agree with the source more, we would expect average difference (Δ) would decrease as trust ratings increase. Table 6.1 presents the actual data from these experiments. Note that there are very few comparisons for Trust Ratings 1-5. Because the number of comparisons for the lower trust ratings is so small, and the margin of error so large, these data points were not included in the analysis here. Instead, we focused on the comparisons made for trust values 6-10.

Table 6.1: Data from experiments run to determine accuracy.

| Trust Rating | Number of Comparisons | Average Difference (Δ) |
|---|---|---|
| **1** | **2** | **0** |
| **2** | **1** | **6** |
| **3** | **1** | **9** |
| **4** | **5** | **3.6** |
| **5** | **1** | **5** |
| 6 | 46 | 1.7 |
| 7 | 85 | 1.5 |
| 8 | 178 | 1.25 |
| 9 | 176 | 1.05 |
| 10 | 371 | 0.95 |

As Figure 6.6 and Table 6.1 indicate, there appears to be a strong linear relationship between trust value and Δ. This is confirmed by the statistics; the Pearson's correlation is greater than –0.991, indicating that there is an almost perfect negative linear relationship between the variables.

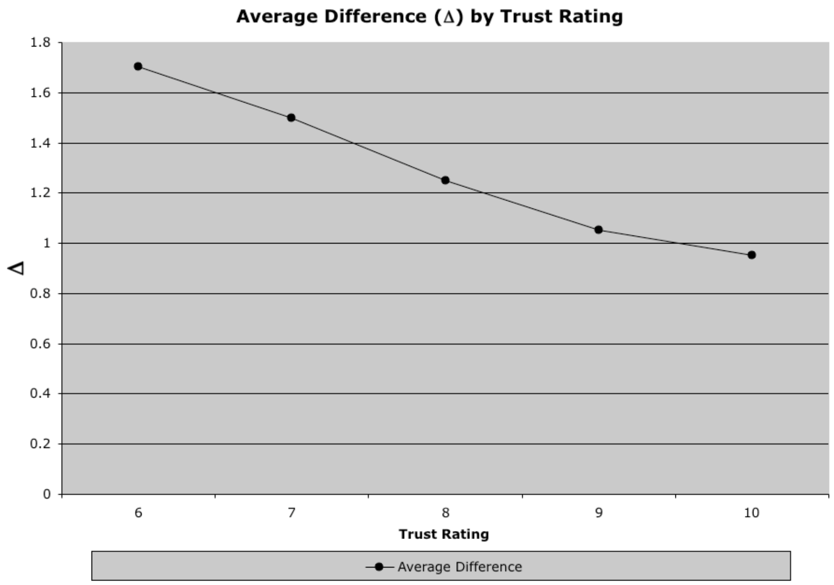


Figure 6.6: The relationship between Δ and Trust Rating.

An ANOVA (shown in Table 6.2) using the full set of data points confirms that accuracy changes significantly as trust changes. The null hypothesis is that there is no significant difference in accuracy among the categories of trust values; probability of this ANOVA result assuming the null hypothesis is less than 0.001. Thus, we can conclude that accuracy is significantly better as trust values increase.

Table 6.2: ANOVA Analysis of the results in Table 6.1.

| Source of Variation | Sum of Squares | d.f. | Mean | F |
|---:|---:|---:|---:|---:|
| **between** | **39.49** | **4** | **9.871** | **6.475** |
| **error** | **1229** | **806** | **1.525** | |
| **total** | **1268** | **810** | | |

Does the social trust between individuals lead to the smaller Δ as trust ratings increase? It is possible that the correlation could it be a result of the underlying structure of the network or the distribution of trust values. To test this, the trust values in the network were randomized. The topology of the network was not altered – each person was connected to the same people – and the distribution of trust ratings was also not changed. The values were just assigned to random edges. This maintains the features of the network while removing the social trust component; the values on the edges no longer represent the actual trust between people.

Several new networks were created with different randomizations of the values within the network. The same experiment described above was conducted on each randomized network. Figure 6.7 shows the average Δ for each trust rating in the randomized networks, as well as the results from the original network. It shows that for each trust value, the Δ value is much higher in the randomized networks. Furthermore, the relationship between trust rating and Δ is less apparent. An ANOVA was run on these data, and the probability of the result shown in Table 6.3 assuming the null hypothesis – that there is no statistically significant difference among the trust values – is 0.42. Thus, the null hypothesis cannot be rejected.

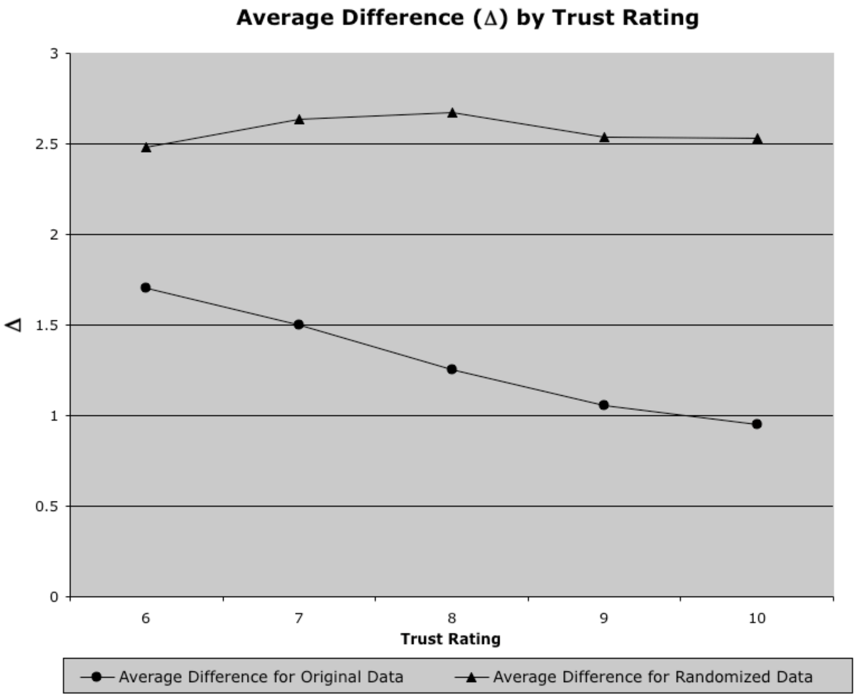


Figure 6.7: Average Δ by trust value for the original social network data and the randomized data.

Table 6.3. ANOVA Analysis of Randomized Network Results

| Source of Variation | Sum of Squares | d.f. | Mean | F |
|---|---|---|---|---|
| between | 17.66 | 4 | 4.415 | 0.9722 |
| error | 8328 | --- | 4.541 | |
| total | 8345 | --- | | |

As one final piece of evidence, the proportion of common neighbors found for each trust value is quite different than is seen in the original network with actual data. Figure 6.8 shows trust distributions overall in the network, among pairs of nodes with common neighbors in the experiments with the original network, and among pairs of nodes with common neighbors in the experiments with the randomized networks.

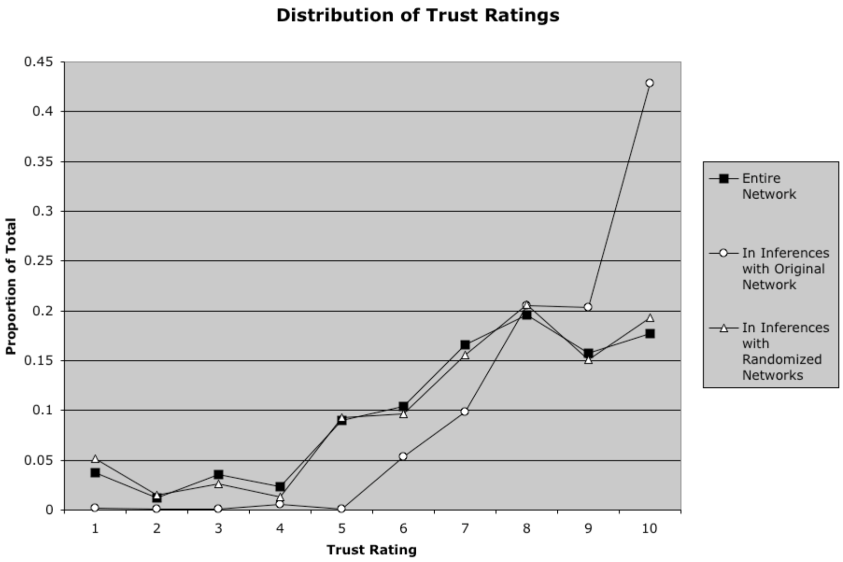


Figure 6.8: Trust distributions in the original network and experiments on the original and randomized networks.

Unlike in the experiments on the original network where there was a sharp increase in the frequency of high trust values among the data points, there is no such increase in the randomized networks. In fact, the frequency of trust values among the pairs with common neighbors in randomized networks follows the distribution in the network as a whole quite closely. Figure 6.9 shows this more closely.

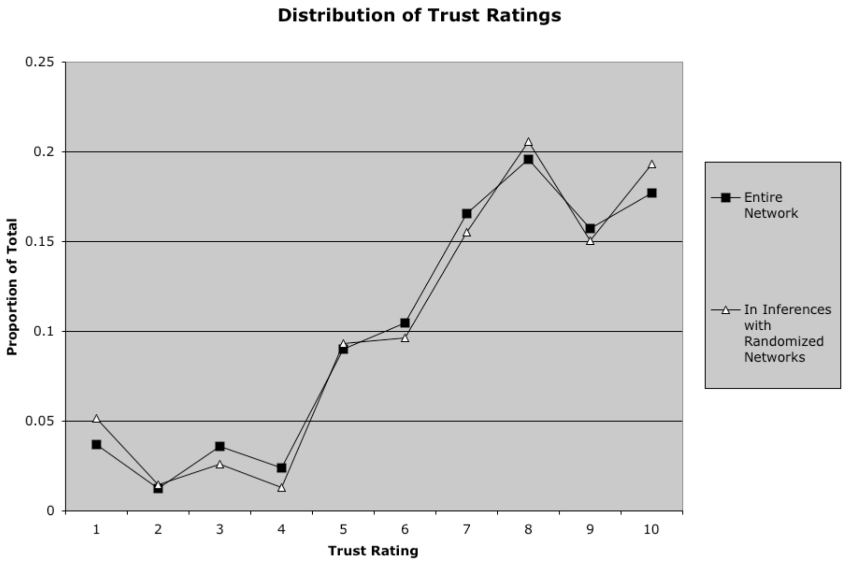


Figure 6.9: Distribution of trust ratings in the original network and among the pairs with common neighbors in the randomized networks.

These analyses show that in this trust network, there is more agreement between nodes connected by high trust ratings than nodes connected by lower trust ratings. Furthermore, common neighbors are found more frequently among pairs of nodes with higher trust ratings. Thus, the increased accuracy among highly trusted neighbors is amplified by the increased frequency of receiving data from those highly trusted neighbors.

If this analysis is carried out fully, we can predict the overall expected accuracy. The Δ for each trust value is multiplied by the frequency that value appears in the experiments. As Table 6.4 shows, the expected Δ in the actual network is almost half what it is in the randomized networks. The lower Δ is synonymous with higher accuracy, and these results show that the structure and trust topology of the network lead to higher accuracy among highly trusted neighbors, and higher accuracy overall.

Table 6.4: Expected Δ in original and randomized networks

| **Trust Rating** | **Δ * Frequency of occurrence in Original Network** | **Δ * Frequency of occurrence in Randomized Networks** |
|---|---|---|
| 1 | 0.000 | 0.125 |
| 2 | 0.007 | 0.034 |
| 3 | 0.010 | 0.072 |
| 4 | 0.021 | 0.029 |
| 5 | 0.006 | 0.257 |
| 6 | 0.090 | 0.240 |
| 7 | 0.147 | 0.410 |
| 8 | 0.257 | 0.551 |
| 9 | 0.214 | 0.383 |
| 10 | 0.408 | 0.490 |
| Overall Expected Δ | 1.117 | 2.073 |

These elements will become a critical in the development of the new trust inference algorithm.

### 6.2.3 Path Length and Accuracy

The length of a path is determined by the number of edges the source must cross before reaching the sink. For example, source→$n_1$→sink has length two. This is length used to determine the correlation between trust value and accuracy. For that analysis, the common neighbors approach was a natural fit. To study the relationship between path length and accuracy, it is reasonable to try to follow a similar approach. Paths of length, 2, 3, 4…$n$ from the source to a node that is also a neighbor of the source would be compiled, and the difference between the source's rating, the rating from the last node in the path would be analyzed with respect to the trust values along that path. For example, paths where each node is connected to the next with a trust rating of 10 may be expected to be more accurate than paths where nodes are connected with lower trust values.

However, certain properties of social networks make this a more complex analysis. Figure 6.10 depicts the types of paths that would be used for this analysis. The path through Node A has only one intermediate node, and this is length of the paths used in the previous section. The path with two intermediate nodes goes through nodes $B_1$ and $B_2$. It is important in this path that the source must *not* be connected to node $B_2$. If there is a connection there, then the path really only has one intermediate node (source → $B_2$ → sink), with $B_1$ as a meaningless placeholder. Similarly, for the path with three intermediate nodes represented in Figure 6.10 by the $C_i$s, the source cannot be connected to $C_2$ or $C_3$ because it would essentially shorten the actual path length being analyzed. Also, $C_1$ cannot connect to $C_3$ for the same reason because it would skip the intermediate node that completes the path.

The small world properties of the network affect the effectiveness of this analysis. Recall that *connectance* of a node is the ratio of how many of the nodes neighbors are connected to one another versus how many possible connections can be made. Essentially, this is a coefficient of how closely connected members of a neighborhood are. In small world networks, including most social networks, connectance is quite high relative to what would be expected in a random network. In the Trust Project social network, the connectance is greater than 0.66, meaning about 2/3 of possible connections between neighbors of a node exist.

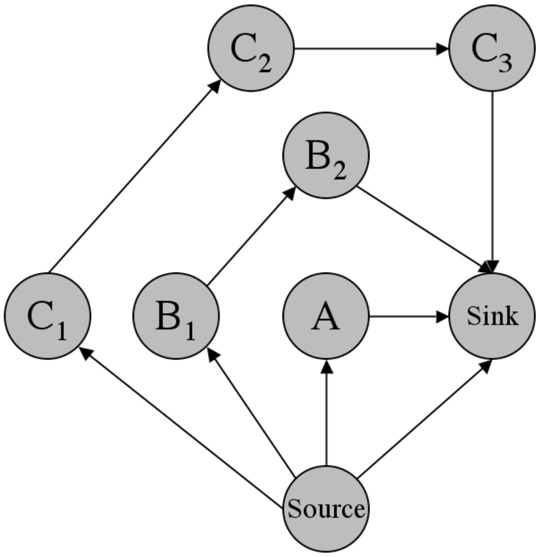


Figure 6.10. Paths from the source to sink of length two (through Node A), three (through the B nodes), and four (through the C nodes). The source must not have connections to any nodes in this figure other than those shown because it would create a shorter path through the additional edge.

When looking for longer paths, the source must choose two of its neighbors (call them $n_1$ and $n_2$) who are not directly connected. The connectance statistic of 0.66 for this network means that, on average, only 1/3 of the possible neighbor pairs are not directly connected. To make a path of length 3, there must be one node (say $n_3$)that connects $n_1$ and $n_2$. Since $n_3$ and the source are unconnected neighbors of $n_2$, they are one of the 1/3 of $n_2$'s neighbor pairs. If $n_1$ and $n_2$ are also used to find a path of length 4, there will have to be another node adjacent to $n_2$ that is not adjacent to the source, and not adjacent to $n_3$. That adds another two unconnected pairs to the neighborhood of $n_2$. Similarly, the number of unconnected neighbor pairs of $n_1$ will also increase.

However, because of the high connectance, there can only be a limited number of unconnected neighbor pairs. This limits the number of paths that can be found, and as the length of the path increases, the number of unconnected neighbor-pairs increases linearly. This is illustrated in figure 6.11.

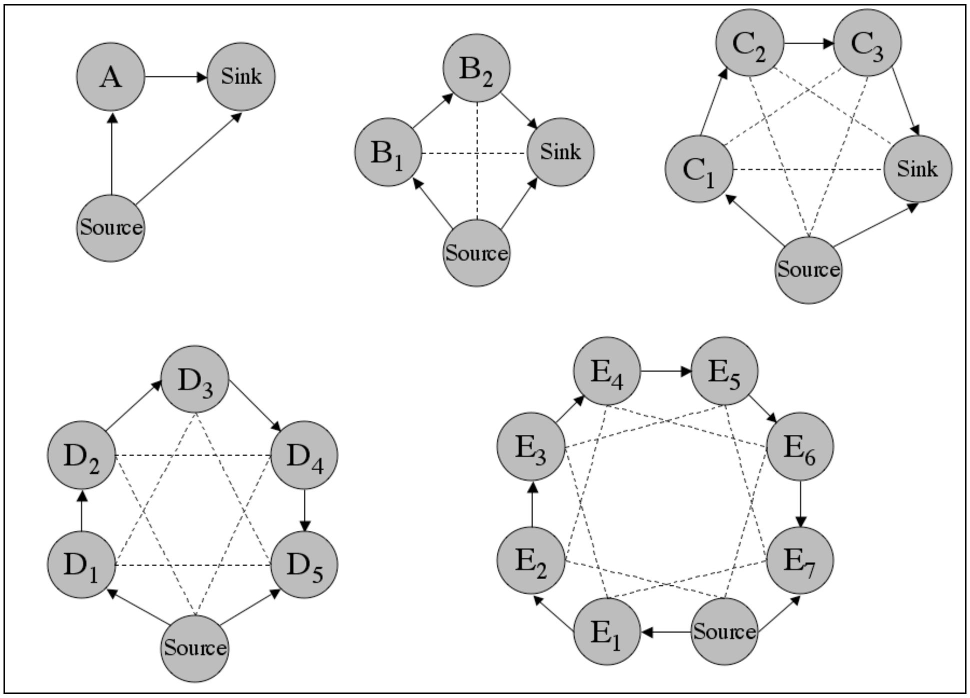


Figure 6.11. Paths of length 2, 3, 4, and 5,. The dashed lines indicate edges that cannot exist because they would shorten the path. As the number of nodes in the path increases, so do the number of edges that must be excluded.

The average degree in the network is 3.19. With an average connectance of 0.66, that means that in the average case only about one pair of neighbors can be lacking an edge between them. Of course, there is variance within the network; some nodes have many more neighbors (standard deviation on the average degree is 4.28) and others have a different connectance (standard deviation is 0.508). However, these properties limit the length and number of paths that can be used for a common neighbor's style analysis.

Data from the experiments to calculate the effect of trust provide an opportunity to estimate the effect of length on accuracy. As described in the previous section, for every pair of nodes with a common neighbor in the network, the difference between their ratings of the common neighbor was recorded along with the trust rating between nodes in the pair.

To approximate the effect of length, we use those datasets. Consider a path of length three: source→$n_1$→$n_2$→sink. There are two variables that must be controlled for a path this length: the trust from the source to $n_1$ ($t_{s,n1}$)and the trust from $n_1$ to $n_2$($t_{n1,n2}$). Once those two values are fixed, the final rating, from $n_2$ to the sink, is the used to determine Δ. We will write this as a delta for the path $t_{s,n2}$-$t_{n1,n2}$. Based on the trust rating from the source to $n_1$, we can retrieve the set of Δ values calculated used in the analysis done in section 6.2.1. Similarly, for the trust rating from $n_1$ to $n_2$ there is a corresponding set of Δ values. Those can be combined to produce an estimated average Δ(or )for the longer path.

For each Δ in the data set corresponding to the trust value from source to $n_1$ (call this $\Delta_{s,n1}$) the value represents two possibilities: $n_1$'s rating for the common neighbor was higher than the source's rating, or it was lower. For each of those two we look at each Δ value corresponding to the trust value from $n_1$ to $n_2$ (call this $\Delta_{n1,n2}$ ) That leads to a total of four possibilities: $\Delta_{s,n1}$ and positive $\Delta_{n1,n2}$, positive $\Delta_{s,n1}$ and negative $\Delta_{n1,n2}$ , negative $\Delta_{s,n1}$ and positive $\Delta_{n1,n2}$ , and negative $\Delta_{s,n1}$ and negative $\Delta_{n1,n2}$. However, since Δ is positive, there are really only two $\Delta_{s,n2}$ values: $\Delta_{s,n1} + \Delta_{n1,n2}$ and the absolute value of $\Delta_{s,n1} - \Delta_{n1,n2}$. The frequency of each of these values is recorded for each pair of Δ values drawn from the original data. The result is an approximation of $\Delta_{s,n2}$ for each pair of source→$n_1$→$n_2$ trust values (10-10, 10-9, 10-8, 10-7, 10-6, 9-8, 9-7, 9-6, 8-7, 8-6, 7-7, and 7-6). Notice that the operations performed – addition and absolute difference – are commutative, so the ordering of pairs in this analysis does not matter; the average Δ for i-j is the same as if it were j-i.

Figure 6.12 shows this same process carried out to a path with four steps: source→$n_1$→$n_2$→$n_3$→sink. In this case, there are four possible values for each $\Delta_{source,n3}$: two for each of the two $\Delta_{s,n2}$ values.

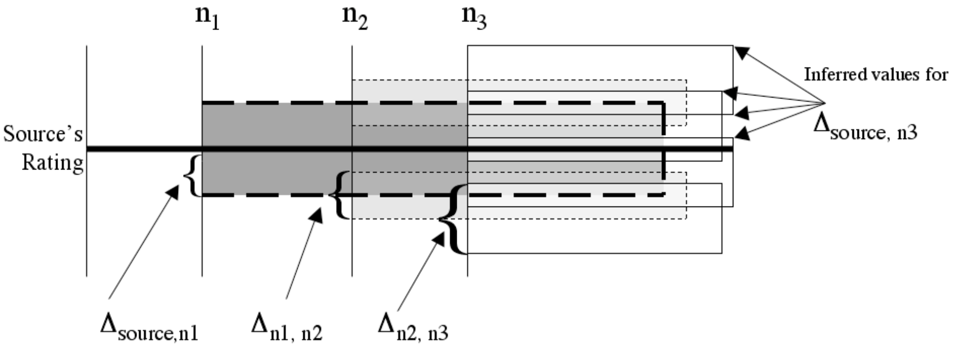


Figure 6.12: An illustration of how $\Delta_{s,n2}$ values are derived for a path length of four.

Although the results obtained by composing these values are only an approximation, they give a general sense of the impact path length has on accuracy. The numbers in section 6.2.1 indicate the measured Δ in paths of length 2 (source→$n_1$→sink). Table 6.5 shows the data for paths of length 2, 3, and 4.

The Path Length 2 data is the original data from section 6.2.1. For the path length 3 data, we can look at either the rows or columns to get a sense of the changes. As either one of the trust values in the pair decreases, the average Δ increases. For example, consider the first row. The average Δ rises steadily from a low of at 10-10 to its highest value in that row at 10-6. Similarly, choose the column headed by 6. The average Δ there is lowest at 6-10 and highest at 6-6. In fact, the lowest average Δ is 1.52 at the pair 10-10, and the highest is 2.497 at the pair 6-6. Not surprisingly, this shows that the correlation between trust value and average Δ is preserved in the longer path.

Table 6.5: Average Δ values for paths from source to sink of length 2, 3, and 4.

| | Source -> *n1* | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 10 | 9 | 8 | 7 | 6 | | |
| | 0.953 | 1.054 | 1.251 | 1.5 | 1.702 | | |
| | 10 | 9 | 8 | 7 | 6 | | |
| Length 3 | 1.520 | 1.588 | 1.698 | 1.958 | 2.076 | 10 | *n1->n2* |
| | | 1.652 | 1.756 | 2.017 | 2.129 | 9 | |
| | | | 1.837 | 2.098 | 2.188 | 8 | |
| | | | | 2.329 | 2.424 | 7 | |
| | | | | | 2.497 | 6 | |
| | 10 | 9 | 8 | 7 | 6 | | |
| Length 4 | 1.92 | 1.969 | 2.048 | 2.287 | 2.369 | 10->10 | *n1->n2->n3* |
| | | 2.017 | 2.092 | 2.327 | 2,406 | 10->9 | |
| | | | 2.16 | 2.396 | 2.464 | 10->8 | |
| | | | | 2.599 | 2.666 | 10->7 | |
| | | | | | 2.725 | 10->6 | |
| | | 2.063 | 2.136 | 2.367 | 2.443 | 9->9 | |
| | | | 2.202 | 2.341 | 2.499 | 9->8 | |
| | | | | 2.632 | 2.698 | 9->7 | |
| | | | | | 2.755 | 9->6 | |
| | | | 2.263 | 2.489 | 2.553 | 8->8 | |
| | | | | 2.687 | 2.746 | 8->7 | |
| | | | | | 2.801 | 8->6 | |
| | | | | 2.862 | 2.919 | 7->7 | |
| | | | | | 2.968 | 7->6 | |
| | | | | | 3.015 | 6->6 | |

Also of interest is that the lowest average Δ is still higher than the average Δ in path length 2. This is true for each value. With a path length of 2, the average Δ for trust rating 9 is 1.054. With a path length of 3, the lowest average Δ for a pair containing a 9 is the 9→10 value of 1.58.

Table 6.6 presents the data for paths of length  5. Since the same set of data points were being used, it is not surprising that these follow the same patterns as were seen in Table 6.6. For any row or column, there is a clear increase in Δ as a trust value decreases. At each step of the path, new error is introduced. If $n_1$ differs slightly from the source, and $n_2$ differs slightly from $n_1$, then *n2* can move closer to the source's value, or it can move further away. At each step, the range of possible error expands; this is visible in figure 6.12, as the space between the line representing the source's rating and the line for the most distant Δ increases at each node.

Table 6.6: Average Δ values for paths from source to sink of length 5.

| *source→n1→n2* | | | | | | | | | | | | | | | *n2->n3->n4* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10-10 | 10-9 | 10-8 | 10-7 | 10-6 | 9-9 | 9-8 | 9-7 | 9-6 | 8-8 | 8-7 | 8-6 | 7-7 | 7-6 | 6-6 | |
| 2.44 | 2.51 | 2.52 | 2.79 | 2.92 | 2.58 | 2.59 | 2.86 | 2.98 | 2.6 | 2.86 | 2.07 | 3.11 | 3.21 | 3.31 | 10-10 |
| | | | | | 2.65 | 2.66 | 2.92 | 3.03 | 2.66 | 2.93 | 3.03 | 3.16 | 3.27 | 3.36 | 10-9 |
| | | | | | | | | | 2.66 | 2.92 | 3.03 | 3.17 | 3.26 | 3.35 | 10-8 |
| | | | | | | | | | | | | 3.39 | 3.48 | 3.57 | 10-7 |
| | | | | | | | | | | | | | | 3.65 | 10-6 |
| | | | | | 2.71 | 2.72 | 2.98 | 3.09 | 2.73 | 2.98 | 3.09 | 3.22 | 3.32 | 3.42 | 9-9 |
| | | | | | | | | | 2.73 | 2.98 | 3.09 | 3.22 | 3.32 | 3.41 | 9-8 |
| | | | | | | | | | | | | 3.45 | 3.54 | 3.62 | 9-7 |
| | | | | | | | | | | | | | | 3.7 | 9-6 |
| | | | | | | | | | 2.73 | 2.98 | 3.08 | 3.23 | 3.31 | 3.4 | 8-8 |
| | | | | | | | | | | | | 3.44 | 3.53 | 3.61 | 8-7 |
| | | | | | | | | | | | | | | 3.69 | 8-6 |
| | | | | | | | | | | | | 3.65 | 3.73 | 3.81 | 7-7 |
| | | | | | | | | | | | | | | 3.89 | 7-6 |
| | | | | | | | | | | | | | | 3.96 | 6-6 |

For each path length, Table 6.7 shows the minimum average Δ for any path containing a given trust value. For example, for paths of length 5 (shown in Table 6.6), the minimum average Δ for a path containing a trust rating of 9 is 2.51 in the 10-9-10-10 path; all other paths containing a 9 have a higher average Δ. In fact, in every case, the path with the lowest average Δ was made up of all 10's with one instance of the lower trust value.

Table 6.7: Minimum average Δ for paths of various lengths containing the specified trust rating.

*Path Length*

| *Trust Rating* | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| 10 | 0.953 | 1.52 | 1.92 | 2.44 |
| 9 | 1.054 | 1.588 | 1.969 | 2.51 |
| 8 | 1.251 | 1.698 | 2.048 | 2.52 |
| 7 | 1.5 | 1.958 | 2.287 | 2.79 |
| 6 | 1.702 | 2.076 | 2.369 | 2.92 |

Figure 6.13 illustrates the relationships from Table 6.7.

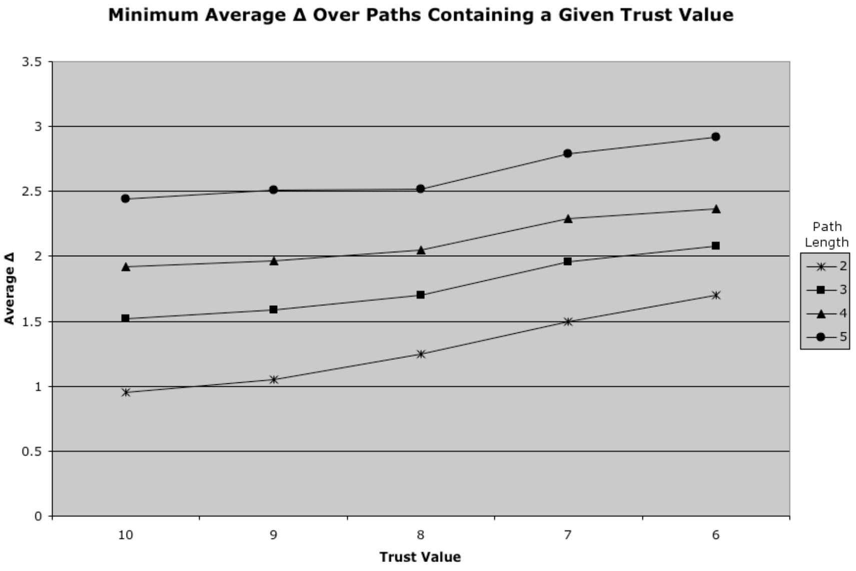


Figure 6.13: Minimum average Δ  from all paths of a fixed length containing a given trust value.

In Figure 6.13, the effect of path length can be compared to the effects of trust ratings. The points for the path length of 2 are the direct comparisons made in section 6.2.1. The average Δ for trust values of 7 on paths of length 2 is approximately the same as the average Δ for trust values of 10 on paths of length 3 (both are close to 1.5). The average Δ for trust values of 7 on paths of length 3 is about the same as the average Δ for trust values of 9 on paths of length 4. A precise rule cannot be derived from these values because there is not a perfect linear relationship, and also because the points in Figure 6.13 are only the minimum average Δ among paths with the given trust rating.

This relationship will be integrated into the algorithms for inferring trust presented in the next section.

## *6.3 TidalTrust: An Algorithm for Inferring Trust*

The in-depth look at the effects of trust ratings and path length in the previous section guided the development of TidalTrust, an algorithm for inferring trust in networks with continuous rating systems. The following guidelines can be extracted from the analysis of section 6.2:

1. For a fixed trust rating, shorter paths have a lower average Δ.
2. For a fixed path length, higher trust ratings have a lower average Δ.

The following sections describe how these features are incorporated into the final TidalTrust algorithm

### 6.3.1 Incorporating Path Length

The analysis in section 6.2 indicates that a limit on the depth of the search should lead to more accurate results, since the average Δ increases as depth increases. To test the effect of limiting path length, the resulting inferred value must be compared to a known value. The same problem occurs here as was described in section 6.2.3; the inferred value must be compared to a neighbor of the source, and the connectance features of the network limit the number of long paths that exist. In fact, performing this analysis showed no statistically significant difference between the accuracy of the inferred values generated when the path length is limited to 2 and the inferred values generated with an unlimited path length.

Although the properties of the network prevented a pure comparison-based analysis of long paths, there are other properties that guide the use of path length in the algorithm. If accuracy decreases as path length increases, as the earlier analysis suggests, then shorter paths are more desirable. However, the tradeoff is that fewer nodes will be reachable if a limit is imposed on the path depth. To balance these factors, the path length can vary from one computation to another. Instead of a fixed depth, the shortest path length required to connect the source to the sink becomes the depth. This preserves the benefits of a shorter path length without limiting the number of inferences that can be made.

### 6.3.2 Incorporating Continuous Trust

The algorithms presented in Chapter 5 had high accuracy because of the rounding that was possible, taking advantage of the properties of binomial distributions. When ratings are continuous instead of binary, the algorithms must be adjusted to take advantage of other properties that will increase their accuracy. This section presents three alternative designs for averaging information from neighbors, and analyzes them to decide which to use in the TidalTrust algorithm

The correlation between trust rating of the neighbor and accuracy of that neighbor's rating is clear (see section 6.2.1). In the binary network-based algorithms, when the source wants to infer a rating to the sink, it begins by asking all of its trusted neighbors for their opinion of the sink. It takes the responses from each node and averages them. To translate this to a system with continuous trust values, the simple average is changed to a weighted average where more weight will be given to nodes with higher trust values.

Let $t_{ij}$ represent the trust rating from node i to node j. The inferred trust rating from node i to node s is given by formula 1.

$$t_{is} = \frac{\sum_{j \in adj(i)} t_{ij} t_{js}}{\sum_{j \in adj(i)} t_{ij}} \qquad (1)$$

This formula respects the fact that more trusted neighbors are generally more accurate. Formula (1) uses trust ratings as the weights. This is the simplest weighted average, but it may not be the most accurate. Before settling on a final algorithm, there are several issues that need to be addressed regarding how trust values should be incorporated.

The results from section 6.2 indicate that the most accurate information will com from the highest trusted neighbors. As such, we may want the algorithm to limit the information it receives so that it comes from only the most trusted neighbors, essentially giving no weight to the information from neighbors with low trust. If the algorithm were to take information only from neighbors with the highest trusted neighbor, each node would look at its neighbors, select those with the highest trust rating, and average their results. However, since different nodes will have different maximum values, some may restrict themselves to returning information only from neighbors rated 10, while others may have a maximum assigned value of 6 and be returning information from neighbors with that lower rating. Since this mixes in various levels of trust, it is not an ideal approach. On the other end of possibilities, the source may find the maximum value it has assigned, and limit every node to returning information only from nodes with that rating or higher. However, if the source has assigned a high maximum rating, it is often the case that there is no path with that high rating to the sink. The inferences that are made may be quite accurate, but the number of cases where no inference is made will increase. To address this problem, we define a variable *max* that represents the largest trust value that can be used as a minimum threshold for every node such that a path can be found from source to sink. The method for calculating *max* will be described in section 6.3.3. The revised formula is given in formula 2.

$$t_{is} = \frac{\sum\limits_{j \in adj(i) \ni t_{ij} >= \max} t_{ij} t_{js}}{\sum\limits_{j \in adj(i) \ni t_{ij} >= \max} t_{ij}} \qquad (2)$$

On the other end of the extreme is the case were only the simple average of the neighbors is taken, and there is no consideration given to the trustworthiness of each neighbor. Although this goes against the previous data suggesting that trust will impact the accuracy of the inferences, it is worth considering this alternative. It is given in formula 3.

$$t_{is} = \frac{\sum\limits_{j \in adj(i)} t_{js}}{|adj(i)|} \qquad (3)$$

Algorithms using these three formulas were tested on both the Trust Project network with approximately 2,000 users, and on the smaller FilmTrust network with about 400 users. In each network, each node was chosen as the source. Each direct neighbor of the source was chosen as a sink, and the trust value was calculated from source to sink. The calculated value was compared to the source's known rating of that neighbor to come up with a Δ value. The average Δs are shown in Table 6.8. The first row shows the average for all Δs that were calculated. However, because of the high connectance and correspondingly high number of short paths, it was not uncommon for the Δ value for a given source-sink pair to be the same for each algorithm; this was the case about 62% of the time in the FilmTrust network, and 53% of the time in the Trust Project network. To highlight the differences among the algorithms, we selected only the cases where the three values were not identical. These are the values shown in the second row.

When looking at all data points (shown in the first row), it appears that the maximum trust average has a lower average Δ for both the FilmTrust and Trust Project network; however, this difference was not statistically significant. When the distinct points were separated out (as shown in the second row), the maximum trust average has a lower average Δ and the difference is significant with $p<0.01$.

We know that the Maximum Trust Average is superior. However, it would be reasonable to expect that between the other two algorithms, the Weighted Average would outperform the Simple average, since the former gives more weight to nodes that should be more accurate. In this analysis, the difference between the weighted average and simple average was not significant for the full set of points or for the distinct points. This may be because the networks are not large enough to show a difference, or it could be that the effect of the weighting is not strong enough to have an impact on accuracy.

Because the results from the maximum trust average are significantly better where there is a difference in the Δ values among the algorithms, this is the averaging mechanism that will be used in TidalTrust.

Table 6.8: Average Δ for the methods of inferring trust. The Distinct Points represent the data taken only when the values returned by the three methods were not all identical. This highlights specifically when one method is more effective than the others.

| | **FilmTrust** | | | **Trust Project** | | |
|---|---|---|---|---|---|---|
| | **Weighted Average** | **Maximum Trust Average** | **Simple Average** | **Weighted Average** | **Maximum Trust Average** | **Simple Average** |
| All Points | 2.136 | 2.062 | 2.242 | 1.400 | 1.250 | 1.429 |
| Distinct Points | 1.857 | 1.710 | 2.027 | 1.264 | 0.972 | 1.317 |

### 6.3.3 Full Algorithm for Inferring Trust

Incorporating the elements presented in the previous sections, the final TidalTrust algorithm can be assembled. The name was chosen because calculations sweep forward from source to sink in the network, and then pull back from the sink to return the final value to the source.

The source node begins a search for the sink. It will poll each of its neighbors to obtain their rating of the sink. Each neighbor repeats this process, keeping track of the current depth from the source. Each node will also keep track of the strength of the path to it. Nodes adjacent to the source will record the source's rating assigned to them. Each of those nodes will poll their neighbors. The strength of the path to each neighbor is the minimum of the source's rating of the node and the node's rating of its neighbor. The neighbor records the maximum strength path leading to it. Once a path is found from the source to the sink, the depth is set at the maximum depth allowable. Since the search is proceeding in a Breadth First Search fashion, the first path found will be at the minimum depth. The search will continue to find any other paths at the minimum depth. Once this search is complete, the trust threshold (the variable *max* in formula 2) is established by taking the maximum of the trust paths leading to the sink. This is illustrated in Figure 6.14.

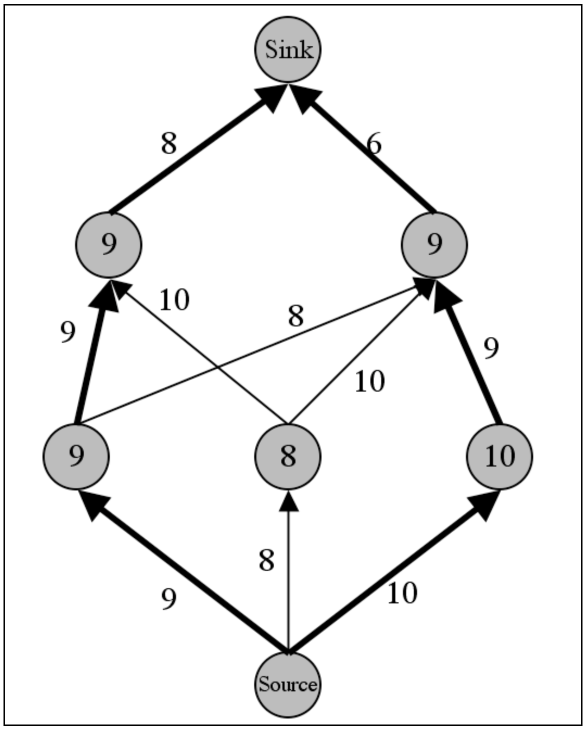


Figure 6.14: The process of determining the trust threshold. The label on each edge represents the trust rating between nodes. The label on each node indicates the maximum trust strength on the path leading to that node. The two nodes adjacent to the sink have values of 9, so 9 is the *max* value. The bold edges indicate which paths will ultimately be used in the calculation because they are at or above the *max* threshold.

With the *max* value established, each node can complete the calculations of a weighted average by taking information from nodes that they have rated at or above the *max* threshold. The pseudo code for TidalTrust, given a graph G, source node, and sink node, is given below:

```
for each n in G
 color(n) = white
 q = empty

TidalTrust (source, sink)
 push (q, source)
 depth=1
 maxdepth = infinity
 while q not empty and depth ≤ maxdepth
    n = pop(q)
    push (d(depth), n)
    if sink in adj(source)
       cached_rating(n,sink) = rating(n,sink)
       maxdepth = depth
       flow = min(path_flow(n), rating(n,sink))
       path_flow(sink) = max (path_flow(sink), flow)
       push (children(n), sink)
    else
       for each n2 in adj(n)
         if color(n2) = gray
            color(n2) = gray
            push (temp_q, n2)
         if n2 in temp_q
            flow = min(path_flow(n), rating(n,n2))
            path_flow(n2) = max (path_flow(n2), flow)
            push (children(n), n2)

    if q empty
       q = temp_q
       depth = depth +1
       temp_q = empty

 max = path_flow(sink)
 depth = depth-1

 while depth>0
    while d(depth) not empty
       n = pop(d(depth))
       for each n2 in children(n)
          if (rating(n,n2)>=max) and cached_rating(n2,sink)≥0
            numerator =
            numerator + rating(n,n2)* cached_rating(n2,sink)
            denominator = denominator +rating(n,n2)
       if denominator > 0
         cached_rating(n,sink) = numerator / denominator
       else
          cached_rating(n,sink) = -1
    depth = depth-1
 return cached_rating(source, sink)
```

The variable *q* is a queue of nodes at the current depth. Beginning at depth 1, there is only one node in the queue *q* – the *source*. We also begin with *maxdepth* equal to infinity because it is unknown at what depth the shortest path to the sink will be found. Line 8 begins a loop through the nodes at a given depth. For each node *n*, we begin by recording the depth at which it was visited. The variable *d(depth)* is a queue, recording all the nodes at a given depth. This will be used later in the algorithm. Then, we check to see if the selected node is adjacent to the *sink*. If the *sink* is in the adjacency list of the selected node, several parameters are set. First, the rating from *n* to the *sink* is cached. Then, the *maxdepth* variable is set. Since a breadth first search is being conducted, all the nodes at the current depth will be searched before proceeding to the next level. The *maxdepth* variable is used to stop the searches once the shortest path length has been discovered. Lines 14 and 15 are used to help find only the highest rated paths. A network flow-type model is used for finding the highest trust path. The lowest trust rating along a series of links in a path limits the overall flow of trust through that path. Line 14 takes the minimum of the flow currently seen from the source to the node, and then from the node to the sink. This records the minimum trust value along the path from source to sink. Line 15 then determines the maximum flow path to the sink. It will either come from a previously found path or from the current path. We assume that both the min and max functions handle undefined values for *path_flow*, and, if an undefined value is seen, the defined alternative will always be returned. Finally, the sink is added to *n*'s list of *children*. The *children* are a subset of all nodes adjacent to *n*. In general, the list of children will include either the *sink*, or all nodes except for those that have been discovered earlier in the search.

If the current node *n* does not have a link to the *sink*, then we proceed to the **else** portion of the branch on line 17. Within that branch, each node that has not been seen before (determined by checking the color) is colored grey and added to the *temp_q*. Then, any neighbor that has been first discovered at the current depth (determined by checking to see if it is in the *temp_q*) is added to the list of *children* and the flow calculations described above are repeated. Note that these steps differ from the standard breadth-first search. By checking for nodes in the *temp_q* instead of checking only for previously undiscovered nodes, the condition allows a node to include a neighbor in its list of children, even if that neighbor has been previously added to the *temp_q* by another node on the same level. Figure 6.15 illustrates this. If that section of code were missing, node B would not be able to include node C in its list of children because node C already would have been colored grey and added to the queue by node A.

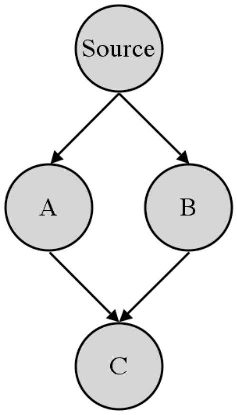


Figure 6.15: A network illustrating when the condition at line 22 will allow for more children. If node A is considered first, it will add node C to the *temp_q* (list of nodes to be searched at the next depth) and color it gray. TidalTrust will allow node B to also include node C in its list of children, even though it was previously discovered by node A.

After every node at the current depth has been checked, the queue *q* is set equal to the *temp_q*. This replaces the empty *q* with the set of nodes at the next depth. The depth variable is also incremented at this point. When the **while** loop at line 8 repeats, the queue *q* will not be empty. If no path was found to the sink at the previous depth, then the *maxdepth* will still be equal to infinity and the loop will continue.

Once the sink has been found or all nodes that can be reached from the source have been searched, the loop ends. At that point, the *max* variable, which stores the maximum trust flow along a path from the source to sink can be set. The depth is also decremented to lead into the **while** loop at line 32. This loop backtracks from the deepest nodes back to the source at depth 1. The loop at line 33 goes through all of the nodes seen at the current depth. For each node in the *children* list, the child's rating is weighted and added to a running sum only if the rating from the node to the child is at or above the *max* threshold, and if the child has a cached rating for the sink. After each child has been seen, the weighted average is cached. After all nodes at a given depth have been looked at, the depth is decremented and the steps are repeated for the next level.

Once the loop has completed, the rating from the source to the sink will be cached, and that value is returned.

The complexity of this algorithm is O(V+E). Within the loop at line 8, each node is colored gray when it is enqueued. Since only white nodes are enqueued and no node is ever whitened after it is grayed, each node will only be visited once. Enqueueing and dequeueing take O(1) time, the time for visiting the nodes is O(V). The adjacency list for each node is looked at only when the node is dequeued. It is used at line 11, when we check if the sink is contained in that list, and in the loop at line 18. In the worst case, searching for the sink at line 11 will require visiting each item in the adjacency list one time, and the loop at line 18 will definitely require searching the entire adjacency list. The sum of the length of all the adjacency lists is θ(E), and to search each line twice will still require θ(E) time since the constant is incorporated. Thus, at most O(E) time is spent scanning the adjacency lists, and making the calculations for path length (which runs in O(1) time).

Together, the loops at line 32 and 33 will visit each node in the network, for a total of O(V) steps. Each node is only visited once. Again, for each node, a subset of its adjacency list is visited only one time, in the loop at line 35. Each of the steps within these loops can be accomplished in O(1) time, so the second portion of the algorithm also has the complexity O(E).

The initialization step at lines 1-2 take O(V) time. Thus, the total time for the algorithm is O(V) + O(E) + O(E) = O(V+E) time. Thus, TidalTrust runs in time linear to the size of the adjacency list representation of the network. For the networks in this research, where E >> V, the algorithm is basically running in O(E) time.

## *6.4 Accuracy of TidalTrust*

### 6.4.1 Discussion of Trust and Accuracy

The accuracy of any trust algorithm will vary from network to network, simply as an effect of the number of users and their behavior within the social network. There are properties of trust within a social network that should be considered when making statements about accuracy.

Trust is expressed in web-based social networks as an explicit number assigned by the user to each friend. In calculating trust, the algorithm takes this personal perspective into account. When analyzing the results, the differences between people is also important. For example, consider the case where a user has rated a number of friends and given every one of them ratings of 9 or 10 on a 1-10 scale. What does this mean? There are several possibilities.

1. The user has a high propensity to trust (or is naive) and is likely to give high ratings to anyone. If this is the case, less trustworthy people may receive higher ratings because of this user.
2. The user feels bad about assigning low ratings, and gives high ones (even though the information is kept private).
3. The user only bothers to rate people that are very trustworthy, and does not spend time on anyone else. In this case, the user may be providing very accurate information about the people who have been rated.
4. The user does not understand the scale, and may be mis-assigning ratings. For example, if the user incorrectly treats the middle of the scale – a rating of 5 – as neutral and lower ratings as expressions of *distrust*, then the ratings from that user will all be skewed.

We cannot know which of these scenarios is responsible for the user's behavior, or if there is some other factor at work. For example, it is known that there is volatility in user ratings based; a user who has rated a string of untrustworthy people may be inclined to give a lower rating to the next person they rate than if the preceding people had been highly trustworthy (Cosely, et al., 2003). As a result of this potential inconsistency and volatility, there is no clear action to be taken. The values cannot be normalized. It is clear that normalizing a rating of 9 down to a 1 would rarely capture the user's intention. Even normalizing to a value between 1 and 9 could be incorrect if the user understands and intends to give the ratings they gave. Even if we know the user has made some poor assumptions to motivate the way ratings were assigned, there is little evidence that suggests how to deal with such a situation. As a result, there is no choice but to assume the user meant the ratings that were given, and to proceed with calculations.

This highlights the fact that trust is imprecise, particularly as it is expressed in web-based social networks. Variation between users, within a single user's rating depending on their state of mind at the time they gave the ratings, and in intended meaning all affect the possible accuracy. With this in mind, it is unreasonable to expect that a trust inference could be made to within, say ±0.1. Even the meaning of such small differences is unclear.

Additionally, it is important to note that trust is *not* simply another word for correlation. Although I have shown that there is a correlation between trust rating and user similarity by comparing ratings of common neighbors, trust captures other concepts as well. Chapter 7 will show that trust is particularly useful for users when their opinions differ from the average user's opinion. This suggests that it may also be the case that users are more likely to assign high trust ratings to friends who agree with them when the users' opinions are quite different from the average, even if there is some variation when their opinion is close to the average, than to assign high trust ratings to friends who have strong correlations in the average case but who disagree when the users' opinions are divergent.

### 6.4.3 Alterations to TidalTrust

As presented above, TidalTrust strictly adheres to the observed characteristics of trust: shorter paths and higher trust values lead to better accuracy. However, there are some things that should be kept in mind. The most important is that networks are different. Depending on the subject (or lack thereof) about which trust is being expressed, the user community, and the design of the network, the effect of these properties of trust can vary. While we should still expect the general principles to be the same – shorter paths will be better than longer ones, and higher trusted people will agree with us more than less trusted people – the proportions of those relationships may differ from what was observed in the sample networks used in this research.

For example, in some networks an increase in the depth of a search may only slightly decrease the accuracy of an inference. Similarly, the distribution of trust in some networks may state that there are only small differences among people rated above a certain threshold, but an increase in depth could lead to a dramatic drop in accuracy. Still another situation could be that in a given network, it is better to search one level deeper allowing people slightly below the *max* trust threshold to be included, because information from more people – even if they are potentially a bit less trusted – can lead to more accurate results overall.

These are all factors that will vary from one network to another. TidalTrust is not designed to be the optimal trust inference algorithm for every network in the state it is presented above. Rather, the algorithm presented here adheres to the observed rules of trust. When implementing this algorithm on a network, modifications *should be made* to the conditions of the algorithm that adjust the maximum depth of the search, or the trust threshold at which nodes are no longer considered. How and when to make those adjustments will depend on the specific features of a given network. These tweaks will not affect the computational complexity – or even the complexity of implementation.

TidalTrust is designed to be a base algorithm that can accept slight modifications to optimize results for any given network. Indeed, because trust is such a loose concept, that is implemented and thought of differently across communities and contexts, it is naive  to suggest that any single strict set of rules would apply equally across the board. When implementing this algorithm in a network, designers should take the time to study and understand the characteristics of their network and fine-tune the implementation accordingly.

### 6.4.3 Related Algorithms and Comparison

There are other methods for computing trust that have been presented in the literature (see chapter 4). Even considering the arguments in sections 6.4.1 and 6.4.2, a comparison of TidalTrust to other algorithms is useful. In this section, I describe which algorithms produce output that is suitable for direct comparison, and show that TidalTrust outperforms the algorithms most suitable for comparison.

Three of the major algorithms that first seemed to be candidates for comparison – Eigentrust (Kamvar, et al., 1998), Appleseed (Zeigler, Lausen, 2004a), and Richardson's algorithm (2003) – all had a common characteristic that prevented a comparison. Each relies on computing the principal eigenvector to make trust computations. The result from all three algorithms is a vector of trust values, corresponding to how much trust the source should have for each node in the network. The values in the vector are not recommended trust values on a given, scale, however. They are *ranks* of the trustworthiness of individuals. There is no way that these rank values can be converted to approximate trust values on a scale, like the 1 – 10 scale used in this work. Any conversion attempt ends with a small number of nodes with reasonable values, and most nodes with very small values.

The goal of those algorithms is to apply the results in a system where the ranks are appropriate. TidalTrust, on the contrary, is designed to recommend explicitly how much the user should trust any individually selected node.

Advogato (Levin, 1998) presents a similar problem for comparison. While the Advogato algorithm does not produce a rank of nodes, it is based on a network flow model that similarly alters the output. Each node is assigned a capacity and the Ford-Fulkerson max-flow algorithm is used to compute trust. With a given capacity, trust values will need to be normalized to within that capacity. That is, if two nodes have the same capacity where one node has rated a large number of people and another node has rated only a few people, the ratings of the first node will be diminished more than the ratings of the second node to fit within the capacity limit. As discussed in section 6.4.1, this adjustment can incorrectly change the intention of the user. For example, a user who only rates friends who are very trustworthy may have twenty friends with a rating of 10. With a capacity limit, those ratings would be normalized to much lower values than if the user had only two friends, each with a rating of 10. Because the normalized trust values are removed from the scale at which the user assigned trust, this algorithm will not output recommended trust values within the same scale as trust is assigned. Thus, since the output is different, it does not make for a valid comparison.

Other work  has suggested some trust mechanisms but these are far simpler than the goal set forth here. For example (Massa, Avesani, 2004) combines trust values with similarity measures in recommender systems. Their trust measure, however, is basically a measure of depth. That is, some maximum depth *d* is set, and the actual shortest distance from the source to the node in question is measured as *n*. The recommended trust value is then just calculated as ($d - n + 1) / d$. This is also designed to be used in systems with only binary trust ratings. Thus, every user at depth 1 will be trusted fully. Every user 2 steps away will have the same recommended trust value, and so on for *n* steps. Thus, there is no reasonable comparison between their algorithm and TidalTrust, which computes an actual value.

On the other hand, the Public Key Infrastructure trust algorithms are much better candidates for comparison. Beth-Borcherding-Klein (1994) is designed to make calculations in the same way as TidalTrust – given two nodes in a trust network, compute how much one node should trust the other, with output in the same scale that ratings are assigned. Although later work has extended and modified Beth-Borcherding-Klein (such as Reiter, Stubblebine, 1999), the original algorithm is still intuitive, and the extensions, designed to prevent attacks or malicious users, are not yet necessary in the small and observably honest networks available for testing.

In addition to Beth-Borcherding-Klein, the simple average of ratings assigned to the sink is also a valid comparison to make. While it is not a complex algorithm, the simple average is one of the most common ways of composing ratings (trust or otherwise).

To determine accuracy, a Δ value was calculated that measured the difference between a node's actual, direct rating of a neighbor and the inferred trust rating for that neighbor. These algorithms were run on the Trust Project network and the FilmTrust network.

Table 6.9: Average Δ for TidalTrust, Beth-Borcherding-Klein (BBK), and simple average.

| | *Algorithm* | | |
|---|---|---|---|
| **Network** | **TidalTrust** | **BBK** | **Simple Average** |
| Trust Project | 1.09 | 1.59 | 1.43 |
| FilmTrust | 1.35 | 2.75 | 1.93 |

As shown in Table 6.9, TidalTrust outperforms both Beth-Borcherding-Klein and the simple average. In both networks, this difference is statistically significant. It is somewhat surprising to see the simple average outperforming the Beth-Borcherding-Klein algorithm. This difference was not statistically significant in the Trust Project network, but was in the FilmTrust network. Because the FilmTrust network is small, and all of the algorithms tend to be less accurate (possibly due to the way members are assigning trust), it may be an artifact of the network that leads to this result, rather than a weakness in the BBK algorithm. Analyzing these types of situations, as well as experimenting on larger networks, is a problem discussed as future work (Chapter 10).

## *6.5 Conclusions*

This chapter introduced the TidalTrust algorithm for inferring trust relationships in trust networks with ratings on a continuous scale. Beginning with an analysis in two naturally-developed networks, I showed the correlation between accuracy, trust rating, and path length. The information gleaned from this analysis was used in evolving an algorithm from the binary network-based algorithm of chapter 5.

TidalTrust is designed to search to the minimum depth at which a path is found from source to sink, and to take only the most trusted paths available at that depth. This comes directly from the analysis in section 6.2. The TidalTrust algorithm was then compared against two other methods for finding and composing trust values. Experiments showed that selecting only the maximum trust paths significantly outperformed both a simple average and a weighted average with no minimum depth. A complexity analysis shows that the algorithm runs in time linear with the size of the network.

Finally, the accuracy of TidalTrust is judged. It is important to consider that accuracy as measured here may not capture all the intricacies of trust, and furthermore, the social nature of trust and the personal variations in the way trust is assigned prevent algorithms from being exceptionally accurate. With that in mind, TidalTrust was compared to the Beth-Borcherding-Klein algorithm and a simple average. On the two networks available for testing, TidalTrust significantly outperformed both.

# Chapter 7

# Trust Inferences in Application: FilmTrust

With evidence that these simple algorithms are effective at making accurate trust recommendations, the next step is to use the inferred trust values in applications. FilmTrust is a website that combines trust networks with movie information. Users can rate films and write reviews on the website. The website then combines the user's trust data and the movie ratings created by trusted friends (and friends-of-friends) to display a custom rating for each movie to the user. Similarly, trust values are used to order the reviews shown for a particular movie. Essentially, the social network becomes the foundation for a recommender system.

In this chapter, a description of the FilmTrust website is followed by an analysis of its features. The accuracy of the recommended ratings is shown to outperform both a simple average and a common recommender system algorithm. Theoretically and through a small user study, some evidence is also established that supports a user benefit from ordering reviews based on the users' trust preferences.

The results here suggest that trust can be used to enhance the user experience. However, because the network available here is small – only about 400 users, with only about 280 users actually connected to the social network – the results are preliminary. To verify them, experiments must be conducted in a larger network, with thousands of users. This is a primary space of future work.

### *7.1 Related Work*

Recommender systems help users identify items of interest. These recommendations are generally made in two ways: by calculating the similarity between items and recommending items related to those in which the user has expressed interest, or by calculating the similarity between users in the system and recommending items that are liked by similar users. This latter method is also known as collaborative filtering.

Collaborative filtering has been applied in many contexts, and FilmTrust is not the first to attempt to make predictive recommendations about movies. MovieLens (Herlocker, et al., 1999; Herlocker et al., 2000), Recommendz (Garden, Dudeck, 2005), and Film-Conseil (Perny, Zucker, 2001) are just a few of the websites that implement recommender systems in the context of films. In section 3, some of their algorithms are applied on the FilmTrust network and the results are compared.

Herlocker, et al. (2004) present an excellent overview of the goals, datasets, and algorithms of collaborative filtering systems. Their paper presents several categories of goals and tasks for which collaborative filtering is used. The category into which FilmTrust falls is also the category that comprises the majority of work in this space: *Find Good Items*. However, FilmTrust is unlike the approach taken in many collaborative filtering recommender systems in that its goal is not to present a list of good items to users; rather, the recommendations are generated to suggest how much a given user may be interested in an item that the user already found. For this to work, there must be a measure of how closely the item is related to the user's preferences.

In this work, the aim is to use trust ratings within the social network as the basis for making calculations about similarity. To determine whether or not a user will like an item, the opinions about the item expressed by other trusted users are composed into a recommendation. For this technique to be successful, there must be a correlation between trust and user similarity. Abdul-Rahman and Hailes(2000) showed that in a predefined context, such as movies, users develop social connections with people who have similar preferences. These results were extended in work by Ziegler and Lausen (2004b). Their work showed a correlation between trust and user similarity in an empirical study of a real online community.

Other work has touched on trust in recommender systems, including (Massa, Avesani, 2004) and (Massa, Bhattacharjee, 2004). These works address the use of trust within systems where the set of commonly rated items between users is sparse. That situation leads to a breakdown in correlation-based recommender system algorithms, and their work explores how incorporating even simple binary trust relationships can increase the coverage and thus the number of recommendations that can be made.

Ziegler and Lausen(2004a) also introduce a mechanism for using trust to calculate recommendations. They compute a trust ranking, based on spreading activation models. That ranking is then used in place of user similarity measures (such as the Pearson correlation) to calculate recommendations for the user.

Accuracy is one measure of the success of an algorithm in making a recommendation. User preference is another. Recent work provides two indications that users will prefer the sort of system that relies on trust in social networks. First, users tend to prefer recommendations from people they know and trust (Sinha, Swearingen, 2001). Related work also showed that users prefer recommendations from systems that they trust and understand (Swearingen, Sinha, 2001). Because social trust is a common part of our everyday lives, this makes the underpinning of the system accessible to users. Other systems have tried to take advantage of these facts by adding in explanations to recommendations. Herlocker, et al., (2000), presented users with information the different ratings assigned to a specific movie, and how many of their friends had made each rating. Their work showed that presenting these explanations to users helped them accept the recommendations from the system because they had more trust in how the recommendations were generated.

### *7.2 The FilmTrust Website*

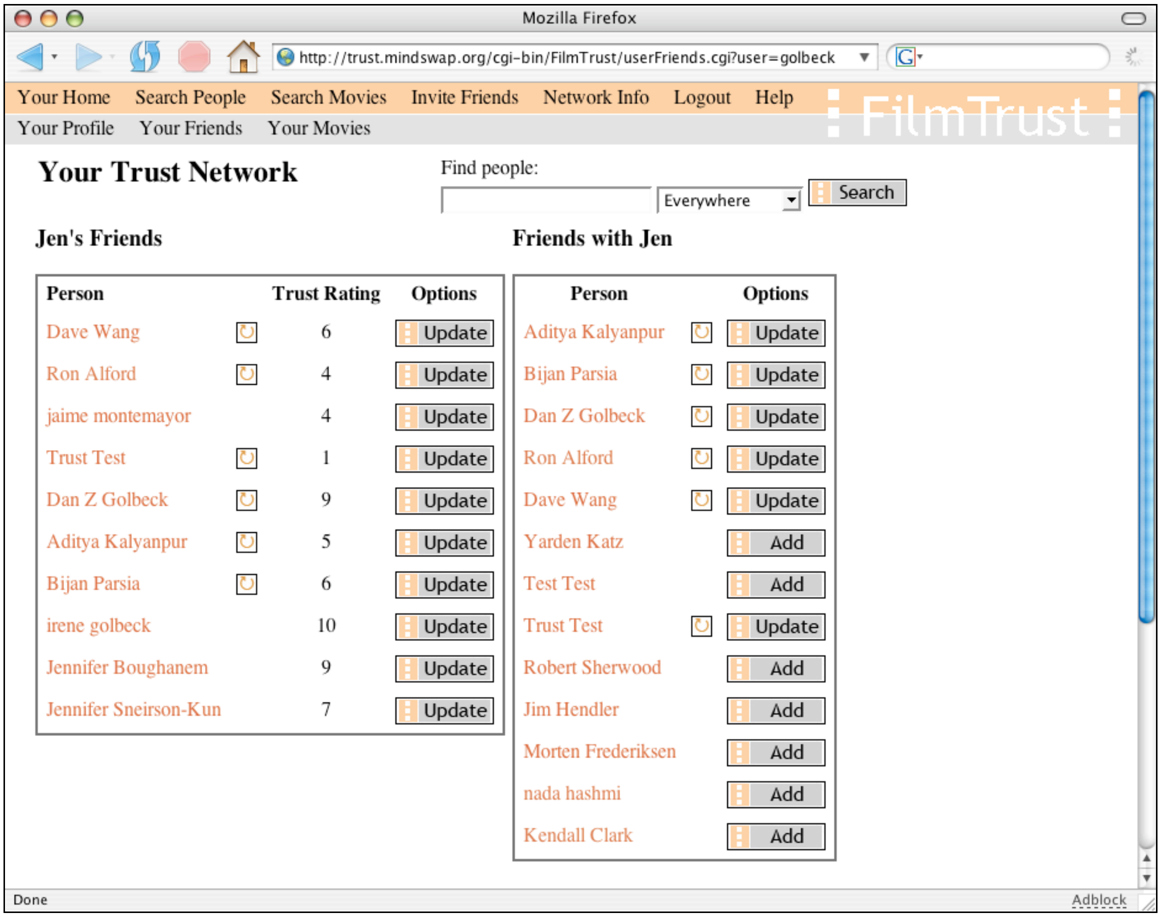
The website for FilmTrust is designed to hide the technical details from users and present a simple, easy-to-use interface.

The social networking component of the website requires users to provide a trust rating for each person they add as a friend. The definition of trust on the website uses the general definition presented earlier cast in the context of movies, that requires a commitment based on the belief about a person's actions. When creating a trust rating on the site, users are advised to rate how much they trust their friend about movies. In the help section, when they ask for more help, they are advised to, "Think of this as if the person were to have rented a movie to watch, how likely it is that you would want to see that film." Watching the film would be the commitment based on the belief that the person will rent a movie of interest.

Part of the user's profile is a "Friends" page, shown in Figure 7.1. In the FilmTrust network, relationships can be one-way, so the page displays a list of people the user has named as friends, and a second list of people who have named the user as a friend. An icon indicates reciprocal relationships and the trust ratings that the user assigned are shown next to each friend.

If trust ratings are visible to everyone, users can be discouraged from giving accurate ratings for fear of offending or upsetting people by giving them low ratings. Because honest trust ratings are important to the function of the system, these values are kept private and shown only to the user who assigned them. The ratings that people assigned *to* the user are not shown.

The other features of the website are movie ratings and reviews. Users can choose any film and rate it on a scale of a half star to four stars. They can also write free-text reviews about movies. The "movies" page of a user displays data for every movie that the user has rated or reviewed. This page is shown in Figure 7.2.



Your Trust Network

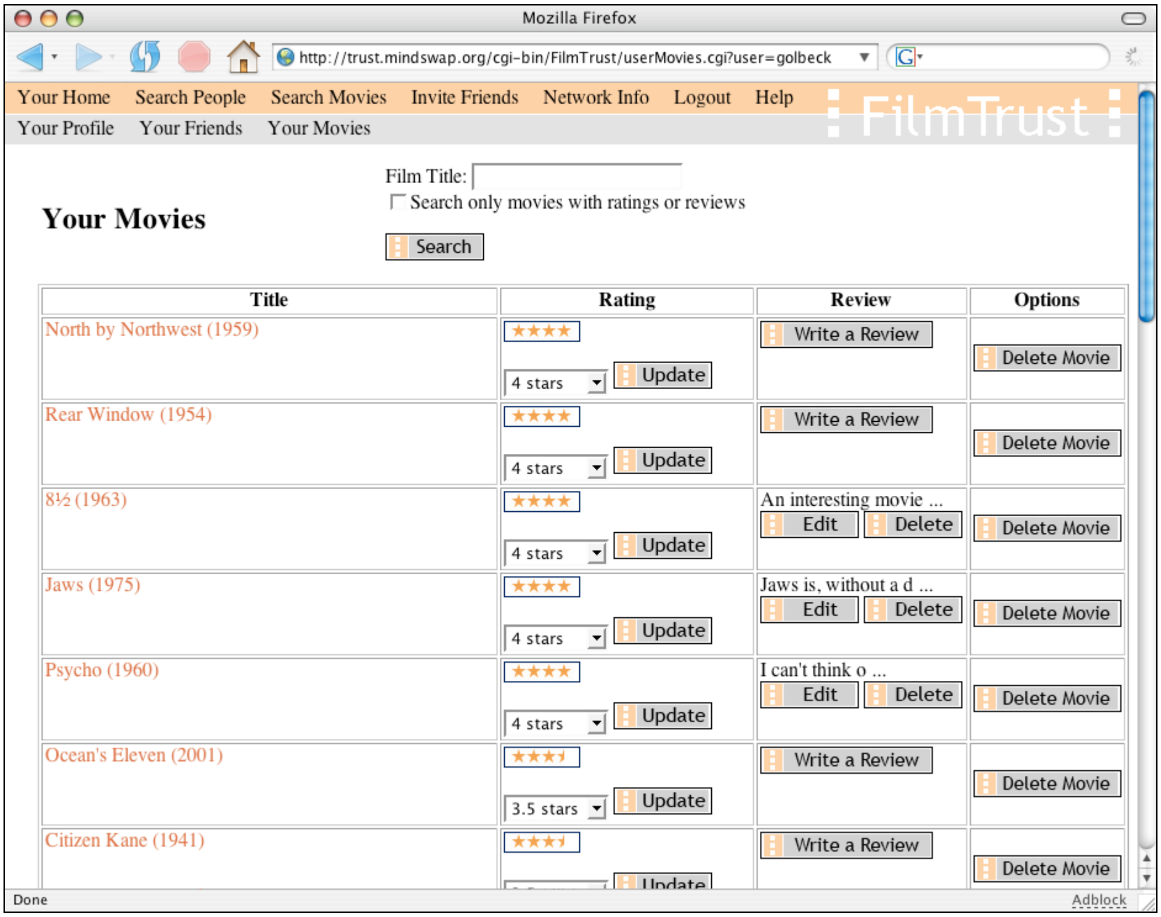
Jen's Friends

| Person | | Trust Rating | Options |
|---|---|---|---|
| Dave Wang | ↻ | 6 | Update |
| Ron Alford | ↻ | 4 | Update |
| jaime montemayor | | 4 | Update |
| Trust Test | ↻ | 1 | Update |
| Dan Z Golbeck | ↻ | 9 | Update |
| Aditya Kalyanpur | ↻ | 5 | Update |
| Bijan Parsia | ↻ | 6 | Update |
| irene golbeck | | 10 | Update |
| Jennifer Boughanem | | 9 | Update |
| Jennifer Sneirson-Kun | | 7 | Update |

Friends with Jen

| Person | | Options |
|---|---|---|
| Aditya Kalyanpur | ↻ | Update |
| Bijan Parsia | ↻ | Update |
| Dan Z Golbeck | ↻ | Update |
| Ron Alford | ↻ | Update |
| Dave Wang | ↻ | Update |
| Yarden Katz | | Add |
| Test Test | | Add |
| Trust Test | ↻ | Update |
| Robert Sherwood | | Add |
| Jim Hendler | | Add |
| Morten Frederiksen | | Add |
| nada hashmi | | Add |
| Kendall Clark | | Add |

Figure 7.1: A users' friend listing at the FilmTrust website.



Your Movies

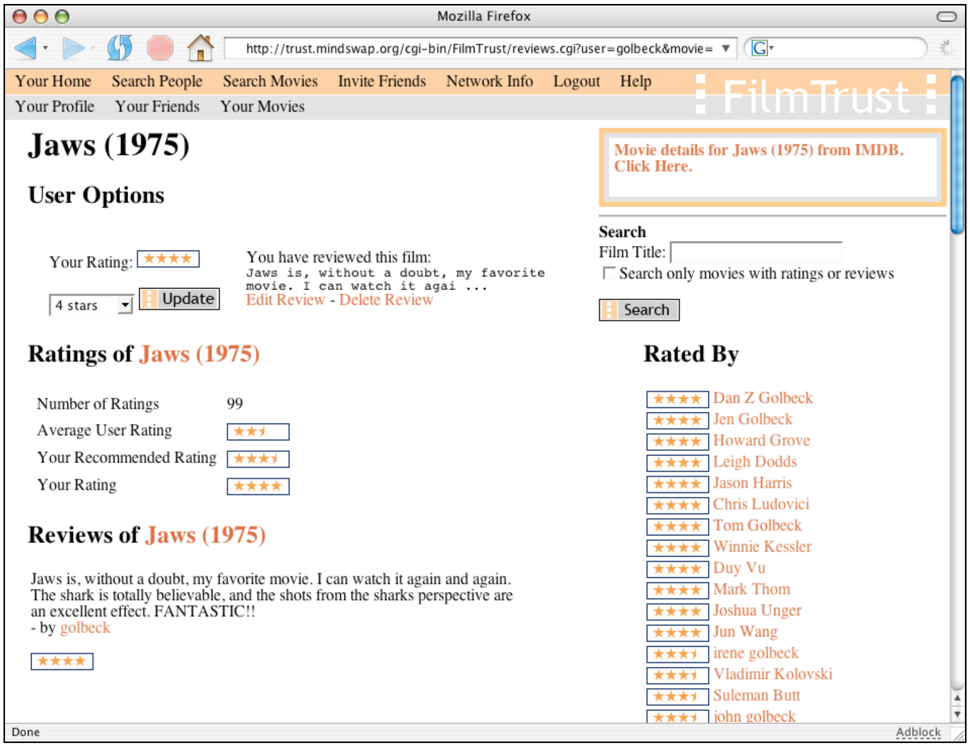
Film Title:

□ Search only movies with ratings or reviews

| Title | Rating | Review | Options |
|---|---|---|---|
| North by Northwest (1959) | ★★★★<br>4 stars Update | Write a Review | Delete Movie |
| Rear Window (1954) | ★★★★<br>4 stars Update | Write a Review | Delete Movie |
| 8½ (1963) | ★★★★<br>4 stars Update | An interesting movie ...<br>Edit Delete | Delete Movie |
| Jaws (1975) | ★★★★<br>4 stars Update | Jaws is, without a d ...<br>Edit Delete | Delete Movie |
| Psycho (1960) | ★★★★<br>4 stars Update | I can't think o ...<br>Edit Delete | Delete Movie |
| Ocean's Eleven (2001) | ★★★½<br>3.5 stars Update | Write a Review | Delete Movie |
| Citizen Kane (1941) | ★★★½<br>Update | Write a Review | Delete Movie |

Figure 7.2: A user's movies page with titles, ratings, reviews, and options.

Social networks meet movie information on the "Ratings and Reviews" page shown in Figure 7.3. Users are shown two ratings for each movie. The first is the simple average of all ratings given to the film. The "Recommended Rating" uses the inferred trust values for the users who rated the film as weights to calculate a weighted average rating. Because the inferred trust values reflect how much the user should trust the opinions of the person rating the movie, the weighted average of movie ratings should reflect the user's opinion. If the user has an opinion that is different from the average, the rating calculated from trusted friends – who should have similar opinions – should reflect that difference. Similarly, if a movie has multiple reviews, they are sorted according to the inferred trust rating of the author. This presents the reviews authored by the most trusted people first to assist the user in finding information that will be most relevant.



Jaws (1975)

User Options

Your Rating: ★★★★

4 stars

You have reviewed this film:

```
Jaws is, without a doubt, my favorite
movie. I can watch it agai ...
```

Edit Review - Delete Review

Movie details for Jaws (1975) from IMDB. Click Here.

Search

Film Title:

□ Search only movies with ratings or reviews

Ratings of Jaws (1975)

| | |
|---|---|
| Number of Ratings | 99 |
| Average User Rating | ★★½ |
| Your Recommended Rating | ★★★½ |
| Your Rating | ★★★★ |

Reviews of Jaws (1975)

Jaws is, without a doubt, my favorite movie. I can watch it again and again. The shark is totally believable, and the shots from the sharks perspective are an excellent effect. FANTASTIC!!
- by golbeck

★★★★

Rated By

| | |
|---|---|
| ★★★★ | Dan Z Golbeck |
| ★★★★ | Jen Golbeck |
| ★★★★ | Howard Grove |
| ★★★★ | Leigh Dodds |
| ★★★★ | Jason Harris |
| ★★★★ | Chris Ludovici |
| ★★★★ | Tom Golbeck |
| ★★★★ | Winnie Kessler |
| ★★★★ | Duy Vu |
| ★★★★ | Mark Thom |
| ★★★★ | Joshua Unger |
| ★★★★ | Jun Wang |
| ★★★½ | irene golbeck |
| ★★★½ | Vladimir Kolovski |
| ★★★½ | Suleman Butt |
| ★★★½ | john golbeck |

Figure 7.3: The move ratings and reviews page for Jaws. Notice that the average user rating and recommended rating differ. The recommended rating most strongly reflects the ratings of trusted friends.

## *7.3 Site Personalization*

### 7.3.1 Recommended Movie Ratings

One of the features of the FilmTrust site that uses the social network is the "Recommended Rating" feature. As figure 7.3 shows, users will see this in addition to the average rating given to a particular movie.

The "Recommended Rating" is personalized using the trust values for the people who have rated the film (the *raters*). The process for calculating this rating is very similar to the process for calculating trust ratings presented in chapter 6. First, the system searches for raters that the source knows directly. If there are no direct connections from the user to any raters, the system moves one step out to find connections from the user to raters of path length 2. This process repeats until a path is found. The opinion of all raters at that depth are considered. Then, using the algorithm from chapter 6, the trust value is calculated for each rater at the given depth. Once every rater has been given an inferred trust value, only the ones with the highest ratings will be selected; this is done by simply finding the maximum trust value calculated for each of the raters at the selected depth, and choosing all of the raters for which that maximum value was calculated. Finally, once the raters have been selected, their ratings for the movie (in number of stars) are averaged. For the set of selected nodes *S*, the recommended rating *r* from node *s* to movie *m* is the average of the movie ratings from nodes in *S* weighted by the trust value *t* from *s* to each node:

$$r_{sm} = \frac{\sum_{i \in S} t_{si} r_{im}}{\sum_{i \in S} t_{si}}$$

This average is rounded to the nearest half-star, and that value becomes the "Recommended Rating" that is personalized for each user.

As a simple example, consider the following:

- Alice trusts Bob 9
- Alice trusts Chuck 3

- Bob rates the movie "Jaws" with 4 stars
- Chuck rates the movie "Jaws" with 2 stars

Then Alice's recommended rating for "Jaws" is calculated as follows:

$$\frac{t_{Alice->Bob} * r_{Bob->Jaws} + t_{Alice->Chuck} r_{Chuck-Jaws}}{t_{Alice->Bob} + t_{Alice->Chuck}} = \frac{9*4+3*2}{9+3} = \frac{42}{12} = 3.5$$

Judging the accuracy of these ratings can also be done in a way similar to the analysis of the accuracy of the trust calculations. For each movie the user has rated, the recommended rating can be compared to the actual rating that the user assigned. In this analysis, we also compare the actual rating with the average rating. Most movie rating websites do not personalize ratings, and the average number of stars assigned is the most common scheme to find. For the trust-based method of calculating ratings, the difference between the personalized rating and the actual rating should be significantly smaller than the difference between the actual rating and the average rating.

On first analysis, it did not appear that that the personalized ratings offered any benefit over the average. The difference between the actual rating and the *recommended* rating (call this ∂r) was not statistically different than the difference between the actual rating and the *average* rating (call this ∂a). A close look at the data suggested why. Most of the time, the majority of users actual ratings are close to the average. Of course, it should be expected that there is a relatively normal distribution of ratings around the mean, and that a large percentage of ratings will fall close to that mean. A random sampling of movies showed that about 50% of all rating were within the range of the mean +/- a half star (the smallest possible increment). For these users, a personalized rating could not offer much benefit because, in this case, if the recommended rating is close to the actual rating, it is also close to the average. Since most people tend to assign ratings close to the mean, we expect that the average rating will be just as good as a personalized rating for most cases.

The point of the recommended rating is more to provide useful information to people who disagree with the average. For example, when users joined the network, they were asked to rate all of the movies they had seen that were in the top 50 on the American Film Institute's 100 Years 100 Films list (American Film Institute, 2000). Just by the fact that they were on this list, it is reasonable to expect that the average rating will be relatively high. Indeed, among these films the average rating is between 3 and 3.5 stars. At the same time, for nearly every film there is a small population of users who have given *very* low ratings, often of only 0.5 stars – the lowest rating in the system. Reasons for disliking these movies are personal matters of taste. People who have a distaste for westerns will probably give low ratings to a movie like "High Noon". In those cases, the personalized rating should give the user a better recommendation, because we expect the people they trust will have tastes similar to their own (Ziegler, Lausen, 2004b).

To examine the effectiveness of the personal rating to closely approximate the users' actual rating when the ∂a increases, both ∂a and ∂r were calculated with various minimum thresholds on the ∂a value. If the recommended ratings do not offer a benefit over the average rating, the ∂r values will increase at the same rate the ∂a values do. The experiment was conducted by limiting ∂a in increments of 0.5. The first set of comparisons was taken with no threshold, where the difference between ∂a and ∂r was not significant. As the minimum ∂a value was raised it selected a smaller group of user-film pairs where the users made ratings that differed increasingly with the average. Obviously, we expect the average ∂a value will increase by about 0.5 at each increment, and that it will be somewhat higher than the minimum threshold. The real question is how the ∂r will be impacted. If it increases at the same rate, then the recommended ratings do not offer much benefit over the simple average. If it increases at a slower rate, that means that, as the user strays from the average, the recommended rating more closely reflects their opinions. Figure 7.4 illustrates the results of these comparisons.

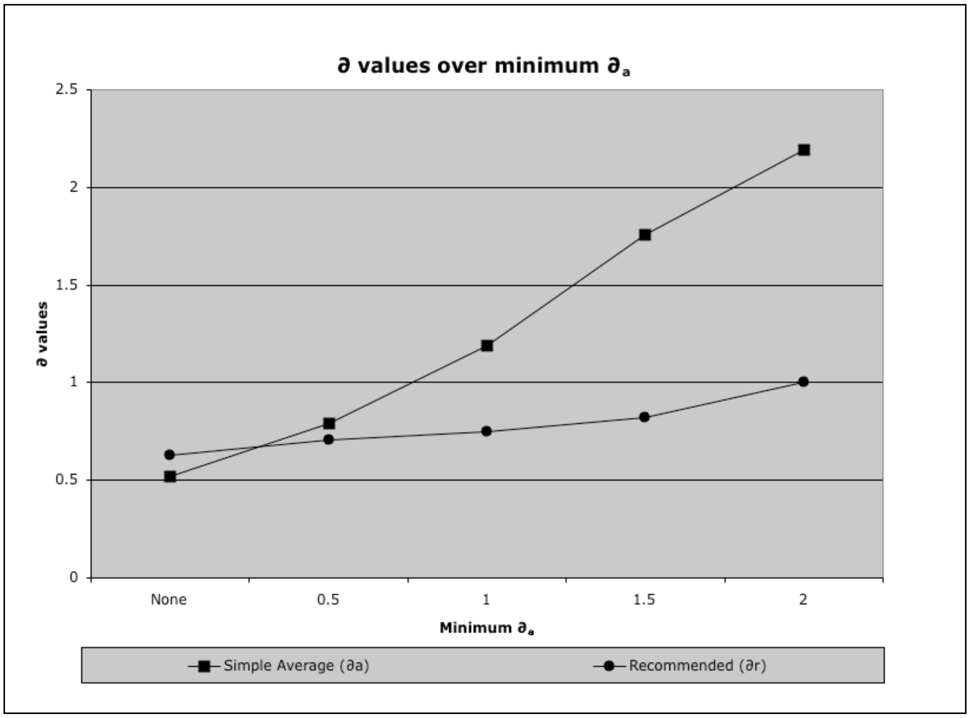
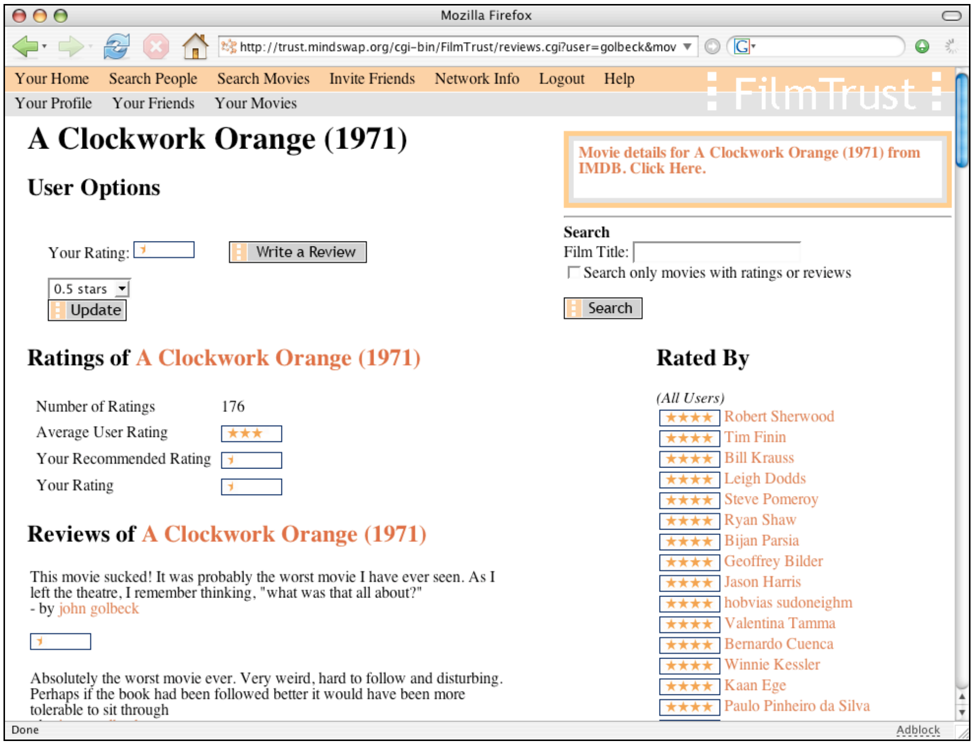


Figure 7.4: Average ∂a and ∂r values for an increasing minimum ∂a threshold. The results here show that as ∂a increases, and users' ratings become increasingly distant from the average, that the recommended rating stays closer to the users ratings. This is reflected by the slow increase in the ∂r value relative to the ∂a.

Notice that the ∂a value increases about as expected. The ∂r, however, is clearly increasing at a slower rate than ∂a. At each step, as the lower threshold for ∂a is increased by 0.5, ∂r increases by an average of less than 0.1. A two-tailed t-test shows that at each step where the minimum ∂a threshold is greater than or equal to 0.5, the recommended rating is significantly closer to the actual rating than the average rating is, with $p<0.01$. For about 25% of the ratings assigned, ∂a<0.5, and the user's ratings are about the same as the mean. For the other 75% of the ratings, ∂a>0.5, and the recommended rating significantly outperforms the average.

Figure 7.3 illustrates one of the examples where the recommended value reflects the user's tastes. The user's rating of "Jaws" is 4 stars, while the average is only 2.5 stars (∂a = 1.5). The recommended rating, on the other hand, is 3.5 stars (∂r=0.5), much closer to the user's actual opinion. Figure 7.5 shows an even clearer example. "A Clockwork Orange" is one of the films in the database that has a strong collective of users who hated the movie, even though the average rating was 3 stars and many users gave it a full 4-star rating. For the user shown, ∂a=2.5 – a very high value – while the recommended rating exactly matches the user's low rating of 0.5 stars. These are precisely the type of cases that the recommended rating is designed to address.



A Clockwork Orange (1971)

User Options

Your Rating: ½

Movie details for A Clockwork Orange (1971) from IMDB. Click Here.

Search

Film Title:

□ Search only movies with ratings or reviews

Ratings of A Clockwork Orange (1971)

| | |
|---|---|
| Number of Ratings | 176 |
| Average User Rating | ★★★ |
| Your Recommended Rating | ½ |
| Your Rating | ½ |

Reviews of A Clockwork Orange (1971)

This movie sucked! It was probably the worst movie I have ever seen. As I left the theatre, I remember thinking, "what was that all about?"
- by john golbeck

½

Absolutely the worst movie ever. Very weird, hard to follow and disturbing. Perhaps if the book had been followed better it would have been more tolerable to sit through

Rated By

(All Users)

| | |
|---|---|
| ★★★★ | Robert Sherwood |
| ★★★★ | Tim Finin |
| ★★★★ | Bill Krauss |
| ★★★★ | Leigh Dodds |
| ★★★★ | Steve Pomeroy |
| ★★★★ | Ryan Shaw |
| ★★★★ | Bijan Parsia |
| ★★★★ | Geoffrey Bilder |
| ★★★★ | Jason Harris |
| ★★★★ | hobvias sudoneighm |
| ★★★★ | Valentina Tamma |
| ★★★★ | Bernardo Cuenca |
| ★★★★ | Winnie Kessler |
| ★★★★ | Kaan Ege |
| ★★★★ | Paulo Pinheiro da Silva |

Figure 7.5: A user's view of the page for "A Clockwork Orange," where the recommended rating matches the user's rating, even though ∂a is very high (∂a = 2.5).

Thus, when the user's rating of a movie is different than the average rating, it is likely that the recommended rating will more closely reflect the user's tastes. When the user has different tastes than the population at large, the recommended rating reflects that. When the user has tastes that align with the mean, the recommended rating also aligns with the mean. Based on these findings, the recommended ratings should be useful when people have never seen a movie. Since they accurately reflect the users' opinions of movies they have already seen, it follows that they should also reflect how the users will like a movie that they have not seen. This is not the typical behavior of a recommender system, in that it does not generate a list of recommendations for the user, but it can help users decide whether or not a movie is within their tastes. Because the rating is personalized, originating from a social network, it is also in line with other results (Swearigen, Sinha, 2001; Sinha, Swearigen, 2001) that show users prefer recommendations from friends and trusted systems.

These results show that the trust-based recommendations are more accurate than the simple average in some cases, but there are many other mechanisms for recommending how much a user may like a given item. Among Automated Collaborative Filtering (ACF) algorithms, one algorithm that has been well tested is the classic user-to-user nearest neighbor prediction algorithm based on Pearson Correlation (Herlocker, et al., 2002). Correlation values between the user and another person in the system are computed by finding the Pearson correlation of the user's movie ratings and the other person's movie ratings. If the correlation is negative, meaning that the two people tend to give opposite ratings, the person is ignored. To compute a recommended rating for a specific item, the algorithm computes the weighted average of the ratings for that item. The average is weighted by the correlation values of everyone in the network who rated that item.

This algorithm was run on the FilmTrust network, repeating the same experiments as above. The difference between a user's actual rating of a film and the ACF calculated rating is given by $\partial cf$. As is shown in Figure 6, $\partial cf$ closely follows $\partial a$. As with the previous analysis, the comparison was made over a series of increasing minimum $\partial a$ values. For $\partial a<1$, there was no significant difference between the accuracy of the ACF ratings and the trust-based recommended rating. However, when the gap between the actual rating and the average increases, for $\partial a>=1$, the trust-based recommendation outperforms the ACF as well as the average, with $p<0.01$. These results are illustrated in Figure 7.6. Because the ACF algorithm is only capturing overall correlation, it is tracking the average because most users' ratings are close to the average.

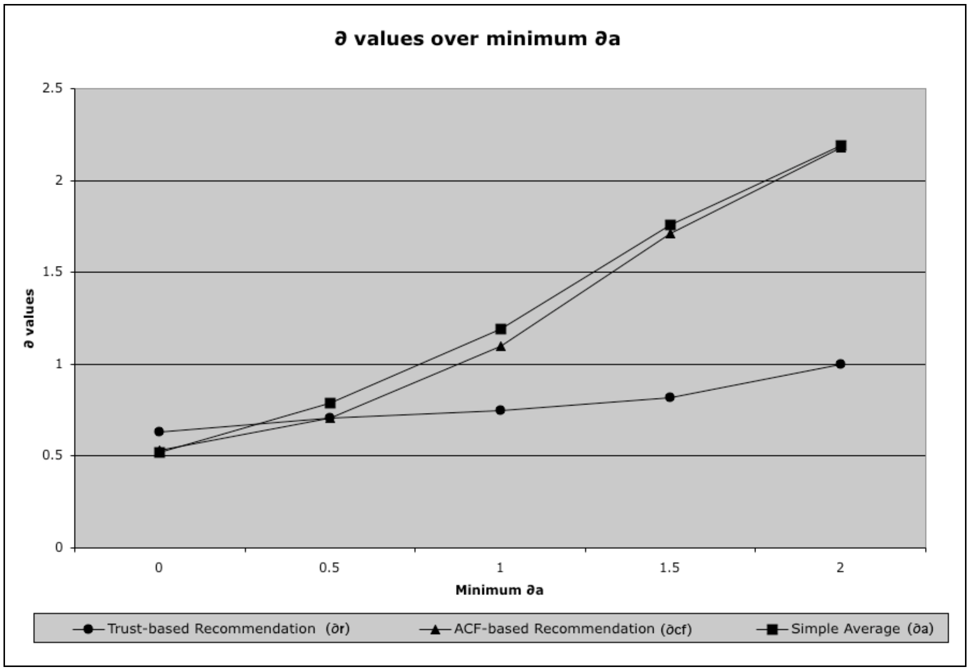


Figure 7.6: The increase in ∂ as the minimum ∂a is increased. Notice that the ACF-based recommendation (∂cf) closely follows the average (∂a). The more accurate Trust-based recommendation (∂r) significantly outperforms both other methods.

One potential drawback to creating recommendations based solely on relationships in the social network is that a recommendation cannot be calculated when there are no paths from the source to any people who have rated a movie. This case is rare, though, because as long as just *one* path can be found, a recommendation can be made. In the FilmTrust network, when the user has made at least one social connection, a recommendation can be made for 95% of the user-movie pairs. However, for users who are isolated from the network, having listed no one as friends, it is impossible to calculate a recommendation because there is no way to judge the trustworthiness of other people in the network. Forty-six percent of the users in the network had not created links to any friends; counting the movies rated by these people, an overall 60% of the user-movie pairs can be rated.

The purpose of this work is not necessarily to replace more traditional methods of collaborative filtering. It is very possible that a combined approach of trust with correlation weighting or another form of collaborative filtering may offer equal or better accuracy, and it will certainly allow for higher coverage. However, these results clearly show that, in the FilmTrust network, basing recommendations on the expressed trust for other people in the network offers significant benefits for accuracy.

### 7.3.2 Presenting Ordered Reviews

In addition to presenting personalized ratings, the experience of reading reviews is also personalized. The reviews are presented in order of the trust value of the author, with the reviews from the most trustworthy people appearing at the top, and those from the least trustworthy at the bottom. The expectation is that the most relevant reviews will come from more trusted users, and thus they will be shown first.

Unlike the personalized ratings, measuring the accuracy of the review sort is not possible without requiring users to list the order in which they suggest the reviews appear. Without performing that sort of analysis, much of the evidence presented so far supports this ordering. The definition of trust has been used to support many of the calculations made throughout this dissertation. That definition also supports the ordering of reviews. Trust with respect to movies means that the user believes that the trusted person will give good and useful information about the movies. The analysis from section 3 also suggests that more trusted individuals will give more accurate information. It was shown there that trust correlates with the accuracy of ratings. Reviews will be written in line with ratings (i.e. a user will not give a high rating to a movie and then write a poor review of it), and since ratings from highly trusted users are more accurate, it follows that reviews should also be more accurate.

### *7.4 User Study*

In order to understand what benefit, if any, the trust-based personalization of the website gave to users, I conducted a small usability study of the FilmTrust website.

The study comprised 9 subjects, 5 males and 4 females. All of the subjects had at least some college education and several had masters degrees. They ranged in age from 25-36. All of the subjects were frequent internet users, with all but one logging on daily. The subjects were also frequent movie watchers; all of them watched movies at home or in the theater about once a week. There was more of a range regarding the frequency at which subjects looked at movie websites and other types of media about films and the movie industry. Subjects' experience ranged from occasional use each month to daily use. All of them had established accounts with FilmTrust and logged on once or twice beyond the initial registration.

Subjects were asked to log in and then proceed to use the FilmTrust site as they normally would for approximately 15 minutes. After some of their normal usage patterns were observed, the users were directed to perform a few simple tasks: add or remove a friend, rate a movie, write a review, and to give some feedback about the personalized information shown with the movies.

None of the users experienced any problem completing the basic tasks. Everyone was able to find movies, rate them, read and write reviews without difficulty.

One area where feedback was of particular interest was in the effectiveness of the review sorting. The movie "E.T. the Extra Terrestrial" had been rated by 60% of the users in the system. It also had four reviews written by people who had rated it 1.5 stars with a negative rating, 3 stars with a mediocre rating, 3 stars with a positive rating, and 4 stars with a glowing rating. The subjects had a range of opinions about the movie, and this made it a good choice for testing how well users agreed with the ordering of the reviews.

Subjects were asked to look at the reviews and say which most accurately reflected their views and which were less accurate. Subjects read the reviews and generally commented one by one on their opinion of it. They made no comment about the ordering of the reviews in this process. However, after going through each item it was pointed out to the subjects that the ordering was made by inferring the trust value for the author and putting the reviews from more highly rated authors first, and lower-rated authors last. After they were told this, the subjects generally looked back at the reviews and realized that they were in an order that closely corresponded with their opinions. Most of the subjects were very excited when they realized this. One user responded, "They are in exactly the order I would rank them. That's amazing!" Another said, "That's so cool. I agree with them and the sequence they're in." Several users pointed out that they thought the website should explicitly state that the reviews were sorted according to each user's personal preferences so that everyone would be aware of the benefit offered by the ordering. This was a nice anecdotal validation of the analysis that lead to the incorporation of this feature.

Because each subject had different ratings relative to the average, little time was spent on determining whether or not the recommended ratings were more accurate. This was already supported by the actual results above. However, two subjects mentioned that they had occasionally used the recommended rating as a guide for making their own ratings. This was not surprising in light of previous work on this subject; Cosley et al. (2003) suggest that when users are shown a recommended rating in a system, they tend toward conforming their own rating to the recommended one.

### *7.5 Conclusions and Discussion*

Within the FilmTrust website, trust in social networks has been used to personalized the user experience. Trust took on the role of a recommender system forming the core of an algorithm to create predictive rating recommendations for movies. The accuracy of the trust-based predicted ratings in this system is significantly better than the accuracy of a simple average of the ratings assigned to a movie and also the recommended ratings from a Person-correlation based recommender system. The other trust-based feature in the site presents movie reviews in order of the calculated trustworthiness of the authors. Although measuring the quality of this ordering is difficult, anecdotal evidence from a usability study suggest that users generally agree with and appreciate this ordering.

This result provides us with some information about how users are encoding trust. If trust were merely a self-evaluated measure of correlation between two people, we would expect that the accuracy of the recommendations should be similar to the collaborative filtering based results (that use an actual correlation measure). The fact that, in this system, trust is much more effective for making recommendations when the user has an opinion *differing* from the average suggests that trust values also capture the specific notion of the correlation between users on *outlying* values. The correlation of these outlying values is not captured by an overall correlation measure.

One user of the FilmTrust system commented on this: "If we are half-a-star different on a lot of the movies I don't care about, that doesn't affect my trust value as much as the fact that you *really hated* Titanic, too." That correlation, when users are particularly passionate about a film, does seem to fit in with the notion of trust as leading the user to a "good outcome". If Bob recommends a movie with a rating of 3.5 stars, and Alice sees it and decides to rate it 3 stars, the difference of a half star has not done much to violate the terms of trust; Bob's information still lead to a good outcome for Alice (he predicted *approximately* how much she would like the movie). However, if Bob recommends the movie at 3.5 stars and Alice sees it and rates it at 0.5 stars, then Bob has given Alice information that lead her to waste her time seeing a movie she didn't like at all. When approached from the perspective of the definition of trust, it makes sense that trust should particularly capture correlation in those outlying points.

# Chapter 8

# TrustMail: Trust Networks for Email Filtering

## *8.1 Background and Introduction*

The fact that spam has become such a ubiquitous problem with email has lead to much research and development of algorithms that try to identify spam and prevent it from even reaching the user's mailbox. Many of those techniques have been highly successful, catching and filtering a majority of Spam messages that a person receives.

Though work still continues to refine these methods, some focus has shifted to new mechanisms for blocking unwanted messages and highlighting good, or valid, messages. "Whitelist" filters are one of these methods. In these systems, users create a list of approved addresses from which they will accept messages. Any whitelisted messages are delivered to the user's inbox, and all others are filtered into a low-priority folder. These systems do not claim that all of the filtered messages will be spam, but rather that a whitelist makes the inbox more usable by only showing messages that are definitely not spam. Though whitelists are nearly 100% effective at blocking unwanted email, there are two major problems cited with them. Firstly, there is an extra burden placed on the user to maintain a whitelist, and secondly, valid emails will almost certainly be filtered into the low-priority mailbox. If that box contains a lot of spam, the valid messages will be especially difficult to find.

Other approaches have used social networks for message filtering. In [1] Boykin and Roychowdhury create a social network from the messages that a user has received. Using the structural properties of social networks, particularly the propensity for local clustering, messages are identified as spam, valid, or unknown based on clustering thresholds. Their method is able to classify about 50% of a user's email into the spam or valid categories, leaving 50% to be filtered by other techniques.

Our approach takes some of the basic premises of whitelisting and social network based filtering and extends them. Unlike Boykin and Roychowdhury's technique that builds a social network from the user's own email folders, the trust-based technique uses a network that connects users. The algorithm from chapter 6 determines a trust rating for each sender that becomes the score for the email message.

The scoring system preserves the whitelist benefit of making the inbox more usable by making "good" messages prominent. The added benefit is that scores will also appear next to messages from people with whom the user has never has contact before. That is because, if they are connected through a mutual acquaintance in the reputation network, a rating can be inferred. This diminishes some of the problems with whitelists because, since scores are inferred instead of taken directly from a list, fewer valid messages will be filtered into a low-priority mail folder. Though some burden for creating an initial set of reputation ratings does fall on the user, it is possible to rate fewer people and rely on the inferred ratings.

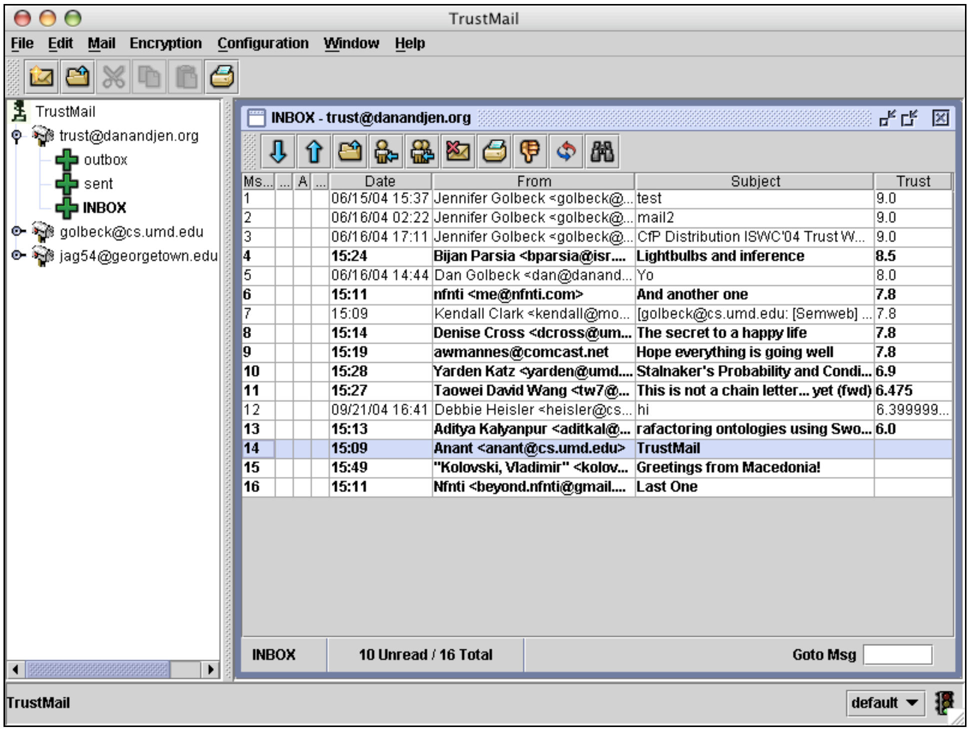
The goal of this scoring system is not to give low ratings to bad senders, thus showing low numbers next to spam messages in the inbox. The main premise is to provide *higher* ratings to *non-spam* senders, so users are able to identify messages of interest that they might not otherwise have recognized. This puts a lower burden on the user, since there is no need to rate all of the spam senders.

Because of this focus, this algorithm is not intended to be a solution to spam by itself; rather, it is a technique for use in conjunction with a variety of other anti-spam mechanisms. There are some spam issues that particularly effect this algorithm. Forged email headers, or *spoofing,* where the "From:" line of a message is altered to look like a valid address is one such issue. This work is not designed to address this problem, and some other technique must deal with forged headers. Because this technique is designed to identify good messages that make it past spam filters, it also do not address the case where a person has a virus sending messages from their account. Other spam-detection techniques will be required to flag these messages.

Essentially, trust is integrated into the email client to serve as a tool for ranking and filtering messages according to their presumed importance. This is not the first technique developed for this task. Maxims (Lasharki et al., 1994) is an agent integrated with an email client that learns how to filter, delete, and archive messages based on user actions. While my work takes a social network-based approach to the problem of message filtering instead of an agent-based approach, the two methods are not contradictory; they could, in fact, be integrated into a system as complementary in the task of easing email overload.

## *8.2 The TrustMail Application*

TrustMail is a prototype email client that adds trust ratings to the folder views of a message. This allows a user to see their trust rating for each individual, and sort messages accordingly. This is, essentially, a message scoring system. The benefit to users is that relevant and potentially important messages can be highlighted, even if the user does not know the sender. The determination of whether or not a message is significant is made using the user's own perspective on the trust network, and thus scores will be personalized to and under the control of each user.



| Ms... | ... | A | ... | Date | From | Subject | Trust |
|---|---|---|---|---|---|---|---|
| 1 | | | | 06/15/04 15:37 | Jennifer Golbeck <golbeck@... | test | 9.0 |
| 2 | | | | 06/16/04 02:22 | Jennifer Golbeck <golbeck@... | mail2 | 9.0 |
| 3 | | | | 06/16/04 17:11 | Jennifer Golbeck <golbeck@... | CfP Distribution ISWC'04 Trust W... | 9.0 |
| **4** | | | | **15:24** | **Bijan Parsia <bparsia@isr....** | **Lightbulbs and inference** | **8.5** |
| 5 | | | | 06/16/04 14:44 | Dan Golbeck <dan@danand... | Yo | 8.0 |
| **6** | | | | **15:11** | **nfnti <me@nfnti.com>** | **And another one** | **7.8** |
| 7 | | | | 15:09 | Kendall Clark <kendall@mo... | [golbeck@cs.umd.edu: [Semweb] ... | 7.8 |
| **8** | | | | **15:14** | **Denise Cross <dcross@um...** | **The secret to a happy life** | **7.8** |
| **9** | | | | **15:19** | **awmannes@comcast.net** | **Hope everything is going well** | **7.8** |
| **10** | | | | **15:28** | **Yarden Katz <yarden@umd....** | **Stalnaker's Probability and Condi...** | **6.9** |
| **11** | | | | **15:27** | **Taowei David Wang <tw7@...** | **This is not a chain letter... yet (fwd)** | **6.475** |
| 12 | | | | 09/21/04 16:41 | Debbie Heisler <heisler@cs... | hi | 6.399999... |
| **13** | | | | **15:13** | **Aditya Kalyanpur <aditkal@...** | **rafactoring ontologies using Swo...** | **6.0** |
| **14** | | | | **15:09** | **Anant <anant@cs.umd.edu>** | **TrustMail** | |
| **15** | | | | **15:49** | **"Kolovski, Vladimir" <kolov...** | **Greetings from Macedonia!** | |
| **16** | | | | **15:11** | **Nfnti <beyond.nfnti@gmail....** | **Last One** | |

INBOX 10 Unread / 16 Total

Figure 8.1: The TrustMail Interface. In this window, messages are sorted according to the trust rating of the sender, with the most trusted appearing highest in the list.

The values shown next to each message are trust ratings calculated with the recipient as the source, and the sender as the sink. In Figure 8.1, the Trust Project social network is used as the foundation. The continuous algorithm from Chapter 6 is applied to make the trust inferences since the Trust Project network uses a 1-10 scale for rating trust.

The ratings alongside messages are useful, not only for their value, but because they basically replicate the way trust relationships work in social settings. For example, today, it would sensible and polite for a student emailing a professor she has never met to start the email with some indication of the relationships between the student and the two professors, e.g., “My advisor has collaborated with you on this topic in the past and she suggested I contact you.” The professor may chose to verify the validity of this statement by contacting the student's advisor or finding information that verifies the claim. These ratings are developed by consulting the social network and ratings within it, and serve as evidence of mutual, trusted acquaintances.

TrustMail replaces the process of searching for information about a recipient by utilizing the data in web-based social networks. Because calculations are made from the perspective of the email recipient, high ratings will have necessarily have come through people the recipient trusts. This allows the trust network-based system to complement spam filters by identifying good messages that might otherwise be indistinguishable from unwanted messages, and carrying the validation of a rating drawn from the users own network of trusted acquaintances.

Techniques that build social networks from messages that the user has sent or received can identify whether or not a message has come from someone in the network.

However, because they are built only from the user's local mail folders, no information is available about people that the user has not previously seen. If the user's personal network is connected in to a larger social network with information from many other users, much more data is available. Previously unseen senders can be identified as part of the network.

Furthermore, since trust values are available in the system, the methods for inferring trust can be applied to present more information to the user about the sender of a message. In the FilmTrust system, it was shown that users benefited from having ratings sorted by the trustworthiness of the author. These results are the basis for sorting messages by the trustworthiness of the sender in TrustMail. However, unlike the FilmTrust where every review was authored by someone in the social network, people will undoubtedly receive many email messages from people who are not in their social network. To understand what benefit TrustMail might offer to users, it is important to understand what percentage of messages we can expect to have ratings for in TrustMail. The next section uses a real email corpus to gain some insight into this question.

### *8.3 Case Study: The Enron Email Corpus*

To gain some insight into how TrustMail may impact a user's mailbox, a large network with many users is required. Although the Trust Project had about two-thousand members, it is not ideal for this type of analysis because it only connects a small community of users, and thus it would only be possible to analyze the mailboxes of a few users. The ultimate application of TrustMail would involve a much larger network or a better connected community. Since this type of social network with trust ratings was not available to test TrustMail, it had to be generated from other existing data.

The Enron email dataset is a collection of the mail folders of 150 Enron employees, and it contains over 1.5 million messages, both sent and received. There are over 6,000 unique senders in the corpus, and over 28,000 unique recipients. These numbers are much greater than the number of users whose mailboxes have been collected because they represent everyone who has sent a message to the users, everyone who has been cc-ed on a message to the users, and everyone the users have emailed. The collection was made available by the Federal Energy Regulatory Commission in late 2003 in response to the legal investigation of the company. Because the messages represent a single community, they are ideal for analyzing the potential of TrustMail. Each message in the corpus was read, and an edge was added from the sender to each of the recipients. This produced an initial social network, although the connections are weak. To be more sure that the links between people represented a relationship, connections were removed for any interactions that occurred only once; edges were only added from source to sink when the source had emailed the sink at least twice.

This social network is obviously lacking trust values. While the strength of relationships could be derived from the corpus of messages, this measure would not correlate to trust as it was defined in chapter 3. Specifically, the justification for the transitivity and composability do not carry through to measures of the strength of relationships. Instead of creating data, the trust component will not be used in the analysis. Relying on the previous results from chapters 6 and 7 to justify the benefits of sorting messages by trust, it is reasonable to conclude that users will see a benefit from sorting messages according to the trustworthiness of the sender. What is unclear in TrustMail is how many messages will actually be rated.

The Enron data allows us to see exactly who has emailed each user. A list of all individuals who sent mail to a given user is compiled. The network is searched for a path from the recipient to each sender. These calculations allow us to determine, in this email corpus, what percentage of senders could be given trust ratings if there were actually a trust network supporting the Enron users.

An analysis of the Enron network showed the following statistics:

- 37% of recipients had direct connections to people who sent them email in the social network; in other words, 37% of the time the recipient had emailed the sender of a received message.
- 55% of senders who were not directly connected to the recipient could be reached through paths in the social network.
- Thus, a total of 92% of all senders can be rated if trust values were present in the social network.

These numbers indicate that users in a community like Enron, an application like TrustMail can provide information about a majority of the incoming messages. While the Enron corpus is a close community of users, it is reasonable to expect that, if users are properly supported in making trust ratings as part of their email client, a similarly high percentage of senders and messages would receive ratings.

## 8.4 Conclusions

This chapter introduced TrustMail, an email client that uses a trust network and algorithms for inferring trust to score each email message. Users are able to sort their mail folders according to the trustworthiness of the sender. An analysis of the structure of the Enron corpus showed that users can expect a majority of their messages to receive ratings if a well-connected trust network is available to support the application.

While the Enron email corpus represents a natural community of users, there are other natural communities where social networks could easily be combined with mail. For example, services like AOL or MSN could include a trust rating option in instant messaging buddy lists, and integrate this into their email applications. Google, for example, already has a social network with trust ratings, through its affiliation with Orkut, and an email application in Gmail. The two could be joined, connecting millions of social network users together.

When users feel they are receiving a benefit from using the trust ratings next to email messages, this encourages them to create more and more accurate trust ratings of people they know. This, in turn, improves the accuracy of the inferred ratings and the number of messages that can be rated. If this cycle is encouraged with a user interface that allows users to easily make trust ratings, trust rated email may become an effective method of email filtering.

# Chapter 9

# Conclusions

The goal of this dissertation was to show that trust relationships in web-based social networks can be used to create applications that offer some benefit to the user by integrating their social preferences. The results presented here show that the benefit is real.

The major contributions of this work are,

1. A formalization of trust as a computational concept within web-based social networks through a definition and description of the functional properties of trust that follow from the definition (Chapter 3)
2. A set of algorithms for inferring trust that have been demonstrated to be accurate (Chapters 5 and 6)
3. The FilmTrust website which shows that trust-based recommendations create significantly more accurate predictive movie recommendations than the simple average or those produced by a collaborative filtering technique when the user's opinion differs from the average. (Chapter 7)

This dissertation has been presented as a walk through the specific intellectual steps necessary to conclude that applications that integrate trust in web-based social networks can, in fact, offer a benefit to the user.

Before any algorithms are created or applications developed that use a social networking technique, several prerequisites that must be met:

1. There must be sufficient social network data on the web to motivate its use in applications. Without a large, active community of people participating in the networks, the incentive to create software around the networks does not exist.
2. Social network data must be publicly available. If each new piece of software requires the creation and growth of a new social network, it is unlikely that it will reach a large community. Optimally, socially inspired software will use networks that already exist on the web.

The social network survey presented in Chapter 3 shows that these criteria are definitely satisfied. With over 133 million user accounts in 127 networks, it is clear that the web is a place alive with social networking activities. Part of the definition I put forth for websites to qualify as web-based social networks is that the social connections must be browsable by others in the system. This means that the data is public in at least that minimal sense. Applications can spider these networks to build models of the social connections, if necessary. However, the Semantic Web offers a much better alternative. The Friend-of-a-Friend project (FOAF) is centered around a vocabulary, presented in the Web Ontology Language (OWL), for describing people and their social connections. Because of the nature of the Semantic Web, where distributed information can be linked together, all of the FOAF files can be merged into a single social network model.

According to current estimates, there are over 6 million people with FOAF files, most of which are automatically generated by social networking websites included in the survey. FOAF data is totally accessible to any software application, and because FOAF is one of the most popular efforts on the Semantic Web, it is likely that developers will have support from a variety of online sources if they choose to base their applications on the FOAF network.

The state of social networks on the web is clearly strong enough to justify developing applications around them. The next step in my work requires trust to be integrated into the social networks. In order for users to effectively express trust, and for developers to make computations with it, there must be a clear definition tailored to the nature of how relationship information is expressed in web-based social networks. In chapter 4, I develop such a definition. For two people, Alice and Bob, I define trust as follows: *Alice trusts Bob if she commits to an action based on a belief that Bob's future actions will lead to a good outcome*.

Computational properties of trust follow from this definition. Specifically, I have described in what way trust can be considered transitive over paths between two individuals, as well as its composability, asymmetry, and personalization. These properties lead to the capacity for making calculations with trust.

The core computational question using trust in web-based social networks is that of inferring relationships. Given two people (a *source* and a *sink*) who are not directly connected, can we use the information in the network to infer how much the source might trust the sink? I present two approaches to making trust inferences based on the values of trust expressed in the network.

The first algorithms, presented in chapter 5, are for networks where trust is expressed as a binary value: either trust or no trust. When those values are treated numerically, with 1 for trust and 0 for no trust, many nice statistical properties follow. If one node is chosen as a source, then every other node in the network will either return a correct or incorrect rating of the sink from the perspective of the source. With the simple algorithm introduced, the probability of a correct inference at a given search depth can be modeled with a binomial distribution. An application of the Central Limit Theorem leads to the conclusion that we can expect a highly accurate inference for the source when the initial accuracy in the network (given by the percentage of ratings made in the network that are accurate from the perspective of the source) is greater than 0.5. With the promising results from chapter 5, I move to the more complex problem of making trust inferences when there is a continuous range of values to represent trust.

In the binary-valued networks, computer-generated networks were ideal because they allowed for manipulation of certain parameters to understand the behavior of the algorithm. For continuous-valued networks, the number of parameters is far more complex, and real web-based social networks with trust values were more appropriate. To that end, I launched two projects designed to build these networks. The first, the Trust Project, is an entirely Semantic Web-based network. I created the FOAF Trust Module, a small extension to the FOAF vocabulary that allows users to express how much trust they have for one another. The Trust Project crawls the web for FOAF files that include trust information, and that information is built into a central model. The size of the network continues to grow, and currently has about 2,000 members. The second network grew with the FilmTrust project, a website that combines a trust network with a movie ratings and reviews. That more traditional web-based social network, where users have accounts and sign in to a central system, has about 400 members. Both networks allow users to rat the trustworthiness of others on a scale from 1 (low trust) to 10 (high trust). These two networks became the testbed for experiments involving continuous valued networks.

The algorithms used for computing trust inferences in binary networks are too simple in their treatment of trust to be directly translated for use in continuous networks; in binary networks, there is no differentiating between trusted people – each is trusted equally. In continuous networks, the values within the network are more complex, and the values to be returned – the inferred trust rating – also must be within the scale of trust, not a binary value. To make the transition, it is first important to have an understanding of how trust and the depth of a search affect accuracy. An analysis of the two trust networks described above showed that higher trusted neighbors tend to agree with a user more than lower trusted neighbors, and that nodes connected by shorter paths tend to agree more than nodes connected by longer paths.

This information is integrated into TidalTrust, the algorithm for inferring trust in continuous networks. Using the same Breadth First Search-based approach to searching for paths as was used in the binary-valued networks, TidalTrust looks for the shortest and most trusted paths. When TidalTrust was compared to the Beth-Borcherding-Klein algorithm in the Trust Project and FilmTrust networks, it produced statistically significantly mode accurate trust inferences. These measures of accuracy are not particularly interesting alone; to see the implications of the TidalTrust algorithm, it should be used in an application and judged by its performance there.

The FilmTrust website is the main place the usefulness of these trust inferences has been tested. On the site, users build a social network, rating how much they trust their friends' opinions about movies. They also can write reviews about films and rate them on a scale from half a star (very bad) to four stars (excellent).

Traditionally, websites that have offered users information about movies have presented an average of all ratings assigned to those films. Recommender systems and collaborative filtering algorithms have tried to personalize the information shown to the user. The rating shown in these systems is known as a predictive recommendation, since it estimates what rating the users might assign to the movie if they see it. In collaborative filtering systems, the user sees a weighted average of the ratings assigned to the film. Those weights usually originate from some measure of correlation between the user and the person who has rated the movie. In FilmTrust, users are shown a predictive rating based on trust rather than a direct measure of correlation. Ratings from more trusted individuals receive a higher weight than rating from lower trusted individuals.

In the general case, looking at every user and every movie, the simple average rating assigned to a movie is just as accurate as the predictive rating from a traditional collaborative filtering algorithm or from the trust-based algorithm. However, as the user's opinion about a movie diverges from the average, I have shown that the trust based recommendation significantly outperforms both the simple average and correlation-based collaborative filtering approach.

This suggests that users are capturing more than just correlation when they rate how much they trust people. While there is an established link between trust and rating correlations (Ziegler, Lausen, 2004b), trust also seems to specifically capture some information about correlation in those outlying points. One user of the FilmTrust system commented on this: "If we are half-a-star different on a lot of the movies I don't care about, that doesn't affect my trust value as much as the fact that you *really hated* Titanic, too." This feature of trust works nicely in the context of the definition of trust, since disagreement on the movies where the user has an extreme opinion will violate the "good outcome" expected from a trusted person much more than a small difference in ratings for movies where the user has an average opinion.

The results from these predictive ratings is a strong confirmation that trust can be used in applications, and that it does, in fact, capture the opinion of users. The second feature of FilmTrust to use trust ratings is the review sorting mechanism. For each movie, the user is shown all of the reviews that have been written. Those reviews are ordered according to the inferred trust value of the author from the perspective of the user. The goal is to show reviews from more trusted individuals more prominently. A small user study produced an enthusiastic response for this ordering, but a more in-depth study will be required to obtain more informative results.

Following on the insights gained from the FilmTrust website, I introduce a second application, TrustMail. This is an email client that uses the inferred trust value from the recipient to the sender as a score for the message. The user can order their mailbox according to this trust value, to show messages from more trusted users first. The results from the FilmTrust experiments suggest that this can be an effective mechanism for sorting messages. In addition to the actual trust value next to each message, the social network implications mean that messages from unknown people can come in with high ratings because of strong connections between the sender and the recipient.

Taken together, these results show that integrating trust in web-based social networks into applications can enrich the user's experience. There is ample data available on the web, and using the algorithms presented here, there is evidence that trust-based features can provide insights and enhancements to applications by respecting the user's social context.

# Chapter 10

# Future Work

There are many ways to continue and extend the work presented in this dissertation. Section 10.1 describes extensions of the FilmTrust system and the analyses of the trust networks. I have also begun other projects utilizing trust inferences, presented in section 10.2, and plan to carry that work into full implementation. Finally, I have chosen trust in social networks as a specific instance of the larger problem of inferring relationships in complex systems that forms the core of my research interests. I have done some work in spaces where a similar approach can be applied. Section 10.3 outlines the Meal of a Meal (MOAM) project, and how it follows from the work here.

## *10.1 Validation of Current Results*

The current results for both the accuracy of the TidalTrust algorithm and its application within FilmTrust are based on relatively small networks with hundreds of users. The FilmTrust and Trust Projects were also built up where the users understood that they were participating in a study.

To show that the accuracy of the algorithms and the predictive recommendation results will hold in real systems, they need to be used in much larger networks. That could be done with an implementation in an existing WBSN, or by growing the size of the current network.

## *10.2 Extensions to Current Work*

There are many aspects of my current work that were outside the scope of this dissertation. Some of the most immediate future work will be to investigate these extensions.

### 10.2.1 Network Structure and Trust Inferences

The two trust networks used in this network exhibit the small world properties expected of social networks. However, the structure alone is not responsible for the effectiveness of the trust inferences. There is meaning to the trust ratings on the edges in the network and, as section 6.2 showed, if the structure is maintained but the trust values randomized, the correlation between a user's trust ratings and the ratings of trusted friends disappears.

The algorithms for inferring trust depends on this correlation, and the relatively high accuracy of the algorithm is a result of it. In a living social network where users create relationships, delete them, and change their trust rating, the inferred trust ratings can change day to day. Even if these changes do not have a significant impact on the structural properties of the network, they have the potential to dramatically impact the quality of the trust ratings. Consider Figure 10.1.

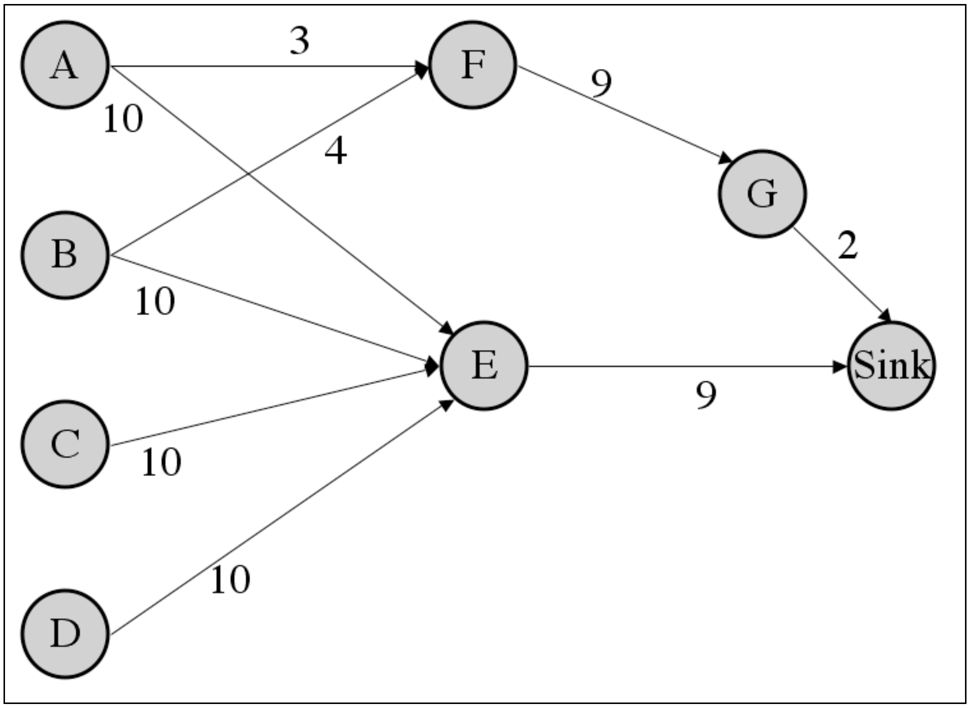


Figure 10.1: A sample social network. In this example, changes in trust values can significantly affect the quality of trust rating.

In Figure 10.1, nodes A, B, C, and D all have a one path to the sink through node E. Nodes A and B have a second path through nodes F and G. The path through E is clearly best: nodes A-D all have rated node E at a 10, and the path has the shortest length of 2. On the contrary, the alternate path has a longer length, and both nodes A and B assigned relatively low ratings to node F, the first node on that path.

If node E were to remove the connection to the sink entirely, nodes C and D would lose the ability to make a trust inference, and nodes A and B would be forced to use the weaker path. Because it is longer and less trusted, we can expect that there will be greater error. If E were to change its rating of the sink, nodes A-D would all be affected. Because nodes C and D receive all of their information about the sink through their connections to node E, any change made by E will pass directly to nodes C and D. While nodes A and B have an alternate path, we can expect the path through E will have more weight since it is shorter and more trusted. Thus, a change in E will also strongly impact nodes A and B. In fact, if TidalTrust is the algorithm being used, nodes A and B receiving all of their information from the path through node E, since it is the shortest path. In that case, the impact of a change in E's rating of the sink will be passed directly to nodes A and B.

Situations similar to that presented in Figure 10.1 can occur when there are hubs in the network, when a user has very few connections, or when a strongly connected group of nodes in the network has a small number of paths connecting it to the larger network. A topic of future research will be to find when changes in one or a small set of trust values will have a significant impact on the accuracy of the network overall, and how the properties of the nodes making changes are seated in the network topology.

An understanding of this connection could lead to more complex algorithms that cope with changes in key nodes more effectively. Because other complex systems will likely have a similar structure, this insight may have future implications for other systems where relationships are inferred.

### 10.2.2 Recommendations with FilmTrust

FilmTrust was designed to serve as a testbed for the hypothesis that basing predictive recommendations entirely on trust was an effective and accurate. The goal was not to create an optimal recommender system. Previous work (Massa, Bhattacharjee, 2004) has suggested that combining trust and traditional collaborative filtering algorithms could be effective. One of the most obvious extensions to the FilmTrust work is to experiment with combinations of different methods for making predictive movie ratings.

Attack resistance is also an important space to do further research in this context. While the goal of trust-based systems is not to make a totally secure system, in the sense that no malicious people can get in, there are natural properties of social networking systems that lead to an inherent ability to prevent bad information from entering the system. In theory, even if a large percentage of the network were made up of nodes spreading bad information (be it trust ratings, movie ratings, or other data), the good nodes in social network should not be affected. Because the algorithms used in this work make calculations from the user's perspective, the only way malicious nodes could affect the results shown to the user is if they were incorrectly included as trusted neighbors by a good node along the way. Since nodes want to preserve the proper function of the system for themselves, there is little incentive to falsely include bad nodes.

That is a theoretical resistance, though. Because of the social nature of trust and how it is assigned, some individuals may put other factors ahead of honest and accurate ratings. This could potentially lead to a deterioration in the accuracy of the results. A near-term project within the FilmTrust space will look at what conditions generally need to be met before bad information starts affecting good nodes, and also at how widely and easily that information can spread into the network. Results from the work described in section 10.2.1 also can be incorporated into this analysis. Ultimately, identifying ways that malicious nodes – or nodes that simply disagree with the perspective of the user – can be eliminated from consideration when social factors allow them in will be an important step in preserving the quality of the results in the system.

### 10.2.3 Privacy

Privacy is vitally important to the successful functioning of any trust and reputation system. The eBay reputation system is one of the most widely used rating systems on the web. While it does not capture trust in the same way I have presented it in this dissertation, it does help highlight some of the issues related to privacy.

On eBay there is a strong disincentive to leave negative feedback to users because they can retaliate with negative feedback, even if it is undeserved. There is little to no recourse for this, and so users are often inclined to say nothing, rather than to make a negative statement. Within web-based social networks, social pressures also make privacy important. For example, on many social networking websites, when a user adds someone as a friend, the friend needs to approve the connection. An informal survey of 300 Orkut users showed that approximately 25% of the connections that users had made were made out of a feeling of obligation. There is a social message that is sent when a friendship request is declined, and users often prefer to add an unwanted friend than to offend the person. When trust is involved, issues become even more complex. If a system relies on honest trust information to function properly, it is important that users do not feel pressured to manipulate their trust values away from what they truly believe. If those trust values are public, people can feel obligated to give higher ratings to friends to avoid offending them. In the Trust Project phase of this work, trust ratings were often publicly visible on the web or otherwise discernable through the network. Many users complained about this, and in FilmTrust the system was altered to make trust ratings totally private.

Finding the proper balance between privacy and public access to the data is a difficult question. One space that this work can be extended is to look at how this might be accomplished, and how it would vary across applications.

## *10.3 Filtering Semantic Web Statements with Trust*

Information – in particular, "content" – on the World Wide Web is presented with an expectation that the information consumer is a human being. People are expected to make use of a variety of cues to ascertain, for example, the proponent of a claim, the author of an article, or the photographer who took a photo and to distinguish these from the refuter of that claim, the publisher of the article, and the aggregator of the photos. Most of these cues are traditional: bylines, attributions, quotations, citations, authorial claims, copyright notices, and the like. Some cues derive from features of Web architecture, such as the use of the Domain Name System (DNS) in Universal Resource Identifiers (URIs), or HyperText Transfer Protocol (HTTP) redirects. Digital signatures can be used to verify the particular origin of a document, and that the document was unchanged in transit, but there is no provision for relating the authenticity of the source of the document and the trustworthiness of the content of that document. Human judgment is required to determine the nature of the document and its content (e.g., real purchase order, example order for debugging, or a parody for amusement). One way that the need for continual human intervention can be eliminated is for people and organizations to set up agreements that certain documents exchanged in certain contexts will be reliable in the appropriate ways. Given that the parties of such agreements all trust each other, accepting information reduces to verifying that it came from a trusted source. Such acceptance need not be only the acceptance of that information *as true* — the modality of the acceptance depends on the agreements. On a community website content may be acceptable for its entertainment value.

The Semantic Web is conceived as the "next generation" of the World Wide Web, wherein much of the content of the Web will not be solely, or even primarily, intended for human consumption. Instead, content sensitive programs will collect, process, exchange, generate, and make decisions based on Web accessible information. As Web agents make more significant decisions, it become more imperative that they are more sophisticated in how they accept information from the Web. While many Semantic Web programs will have significant domain knowledge and thus, presumably, some built-in methods for evaluating the plausibility of new information, perhaps the majority of them will be less specialized. Thus, there is a need for more general, not content specific, techniques.

Many websites are *open*, in that nigh anyone can submit information to be published on the site. This can range from very restricted submissions, such as comments on articles, to the entire content of the site, as with Wikis. This is relatively unproblematic when the information submitted is always presented as a cohesive chunk, say, a Wikipage, or a specific comment, or a specific blog entry. This permits the human reader to evaluate both the content of the chunk and the context of submission (i.e., the provenance).

In contrast, in an open Semantic Website, this is not sufficient. For example, http://www.mindswap.org/ is an open, RDF and OWL driven website. It accepts relatively arbitrary submission of bits of RDF and OWL. It incorporates the assertions in a submission in a variety of pages, presenting the assertions in contexts divorced from their submission and using those assertions to draw inferences, which are themselves presented on different pages. The page generation software has to decide how and where to present or otherwise use each assertion in a submission.

A future application of trust calculations and web-based social networks is as a method for rating and filtering a set of assertions. The provenance information indicates the creator of a statement, and the inferred trust value for that person is used to compose ratings of the reputation or trustworthiness of assertions. Those ratings on assertions can then be used to filter the set of statements used in an application, thereby creating a knowledge base with a known level of validity.

### 10.3.1 From Trust Network Inferences to Accepting Claims

A claim is simply any RDF triple submitted to our website, whether by a Web form, via some aggregation mechanism such as an RSS feed, or by a Web service API. Triples are typically submitted in batches, that I will call "*snippets*", with user-supplied metadata about the snippet inherited by each claim. For example, on http://www.mindswap.org/ users can submit snippets about papers they have authored, either through a free form textarea to craft their RDF directly by hand, or simple elicitation forms to help ensure data consistency, coherence, and completeness.

The key metadata for each snippet is the person submitting the claim, that is, the *claimant*. When a claimant is identifiable as a particular node in the trust network, we can attempt to determine a local trust rating for that claimant. If a user of the site has registered their trust network identifier with the site, then trust rating can be inferred to the claimant with the user's node as the source. Given a particular trust rating for a claim, we customize the display behavior of the site. The simplest customization is to suppress the display of any claim from a claimant whose trust rating is below a certain, user-configurable threshold.

The situation is slightly more complex if there are multiple claimants for a particular claim. There are a number of functions one could use to derive a trust rating for the claim based on the set of ratings for the claimants. However, the straightforward solution – take the maximum rating of the claimants – has a great deal of intuitive plausibility. Since the kind of web sites for which this filtering technique is intended are community oriented portals, the general goal for the site is to be interesting, relevant, and useful to that community. Thus, trust in this context is a measure of our belief that a person will create well-presented, relevant, interesting, and useful information, as determined by the portal's community standards. Since the trustworthiness of the claimant is not interpreted as evidence for or against the claim, there is no need to average or other balance divergent ratings.

### 10.3.2 Using Claim Ratings in Semantic Web Systems

The first application of the ratings for claims is to filter the content of the website based on the value of the rating. The FilmTrust approach to sorting reviews can be considered a simple application of this idea. Based on the calculated trustworthiness of the review's author, the user's experience is customized with respect to how the data is presented.

On a larger scale, consider applying this technology in the context of some of the many "rumor" sites on the web. As one example, MacRumors (http://macrumors.com) allows users to submit rumors about news and technology releases related to Apple Computer. The author of each rumor is tracked, and community members already have the ability to rate rumors as positive or negative. A website with that model could significantly benefit from a semantically-aware system of trust and provenance. A network of ratings that reflect one person's opinion about the quality of posts made by another user, and a system of generating ratings for statements based on their provenance creates the groundwork for allowing users to customize the site. For example, users could choose a minimum trust level of statements that appear on the site. Not only would the site be personalized, it would be *optimized* for the user according to their preferences and social network connections. Although "rumor" sites provide an intuitive example because of the obvious variation in the credibility of statements, this technique can be applied to any site where statements originate from a variety of sources.

### 10.3.3 Filtering Inferences in Knowledge-Bases with Trust Values

Filtering the base claims of the system is useful and interesting, but base claims on Semantic Web sites form only part of the picture. Semantic Web portals tend to be oriented around RDFS and OWL ontologies, that is, logical theories of varying degrees of expressiveness. A Semantic Web site, therefore, is based on a knowledge base and the character of the web site is significantly influenced by the sorts of reasoning it supports. The ordering of filtering and inferring is important to consider. If the set of base claims are filtered first, and then inferences are made over the filtered set, there are two results. First, using the filtered base claims as the fact base for any inferences already filters the inferences. This allows users to conclude that any inferred statements should have at least the same trust rating as the minimum value in the filter, because all of the claims that allow the inference meet or exceed that minimum. However, this does not provide a mechanism for actually calculating a value for an inferred statement  - it only sets a minimum bound. Using this filter-then-infer method also means that the set of inferred statements are limited – it is possible that many other statements could have been inferred from the unfiltered base.

If the order of inferring and filtering is reversed, so all of the inferences are made over the full set, and then the set of all statements – base and inferred – are filtered, the issue becomes more complex because it requires that some trust value be established for the inferred statements. The rating for an inferred statement should be made by some combination of the ratings for statements that lead to the inference. However, a number of different statements could lead to an inference.

Consider the following set of base claims in N3. Each statement is marked with a trust rating *calculated from its claimants*.

```
9    :Person a owl:Class .
8    :SpouseOfStudent a owl:Class;
8         rdfs:subClassOf :Person,
8          [a owl:Restriction;
              owl:allValuesFrom :Student;
              owl:onProperty :marriedTo ],
8           [a owl:Restriction;
               owl:cardinality "1";
               owl:onProperty :marriedTo ] .
7    :Student a owl: Class;
7             rdfs:subClassOf :Person .
9    :University a owl:Class .
6    :attendsUniversity  a owl:ObjectProperty;
6                        rdfs:domain :Student;
6                        rdfs:range :University .
10    :marriedTo a owl:ObjectProperty;
10               owl:inverseOf :marriedTo .
10    :Daniel  a :SpouseOfStudent;
9               :marriedTo :Jennifer .
8      :Jennifer a :Person;
6                :attendsUniversity :UMCP.
```

From this example, we can infer that `:Jennifer` is a `:Student`. What should be the rating for that inferred claim? There are several ways that it can be inferred. Because Jennifer `:attendsUniversity :UMCP` (known at level 6), and the domain of `:attendsUniversity` is `:Student` (rated at level 6) , we can infer that `:Jennifer` is a `:Student`. This inference comes from two simple statements, rated at the same level, so it seems intuitive to rate the inference from these sources at a level 6, like the composite statements. There are other ways to infer that `:Jennifer` is a `:Student`, though, and they may have a higher rating than the 6 achieved with the first method. We also know that `:Daniel`, a `:SpouseOfStudent` (known at level 10), is `:marriedTo :Jennifer` (known at level 9). Since, for instances of the `:SpouseOfStudent` class, the object of `:marriedTo` must be from the class `:Student` (known at level 8), and there must be exactly one spouse (because of the cardinality restriction known at level 8), we know `:Jennifer` must be the only person that `:Daniel` is `:marriedTo`, and thus we can infer that `:Jennifer` is a `:Student`. How to combine this series of statements into a rating for the inferred value is not as clear. .

This example illustrates several issues raised when considering how to rate inferred statements. For each set of statements that leads to an inference, we need a way to combine the ratings of the composite statements to come up with a rating for the inferred statement. Even if we took the simple route of just using the minimum rating from the set of composite statements as the rating for the inferred statement, there are still more problems. If a statement is inferred from several sets of statements, there are now several ratings for that inferred statement. How to choose a final value for the inferred statement is not clear. On top of that, the primary issue illustrated by the above example is that the number of ways a statement can be inferred can grow very quickly. To consider every possible combination of claims that lead to an inference could become computationally difficult. Because inferences are such a fundamental issue on the Semantic Web, the question of establishing trust values for inferred statements is a critical focus of future work in this space. In my ongoing work in this space, I am looking at heuristics for choosing the most trustworthy mechanisms of inference, when pre-filtering by trust can help in this process, and how multiple inference paths might be synthesized to evaluate the overall trustworthiness of an inferred statement.

### *10.4 Meal of a Meal: Inferring Trophic Relationships in Food Webs*

Food webs are models of trophic relationships in an ecosystem. They are built up from observations about what each species eats, and what eats it. This data is gathered from studies of individual species, direct observation, and even examination of stomach contents. However, it is difficult to directly observe every species consuming every species in its diet. As a result, many food webs are incomplete. Paleofoodwebs are food webs assembled from the fossil record, so they are generally much less complete. More complete food webs allow for conservation, management, and an overall better understanding of the dynamics of complex ecosystems.

Is there a way to infer trophic relationships in food webs? This requires additional information. We use taxonomic and phylogenic similarity measures between species to infer trophic connections. The logic supporting this is that species who are closely related are likely to eat similar diets. We have named this project Meal of a Meal (MOAM) in a nod to the Friend of a Friend (FOAF) project that is used in the Semantic Web representation of social network data.

Figure 10.2 illustrates a step in the MOAM process. It is observed that species A eats species B. Then, calculations are made that show species C is closely related to species A. That is evidence that suggests it may eat the same things, so an inferred trophic connection from species A is added to species B. Similarly, a second calculation shows that species D is similar to species B. Thus, we have some evidence that things that eat species B may also eat species D. This is results in the addition of the edge from species A to species D. Finally, a weaker link is added from species C to species D. That uses the inferred connection from species C to species B, and the similarity between species B and species D.

Each added edge is weighted to indicate the strength of the evidence for the inferred trophic relationship. It is then left up to an ecological expert to determine which of these may be viable edges and which are in congruent with ecological reality.

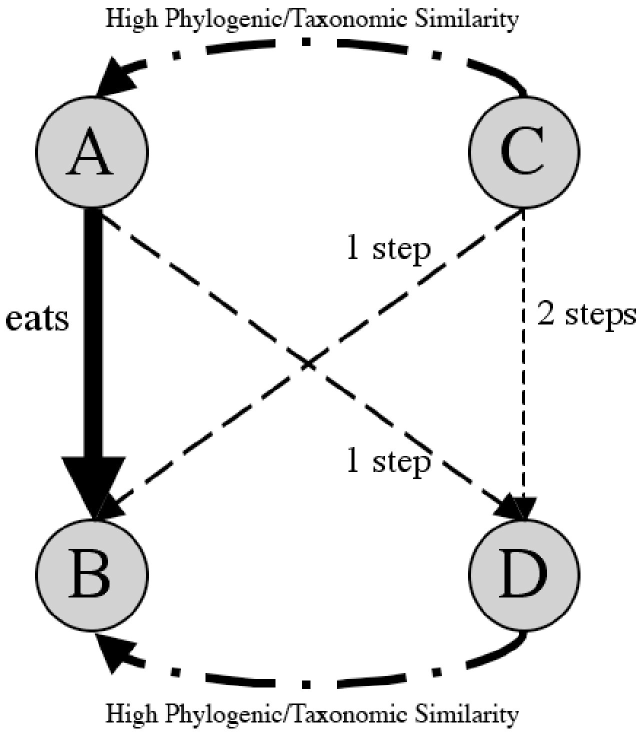

High Phylogenic/Taxonomic Similarity
A
C
1 step
eats
2 steps
1 step
B
D
High Phylogenic/Taxonomic Similarity

Figure 10.2: Inferring trophic connections in food webs. Beginning with the known trophic relationship between species A and species B, this figure illustrates the steps used to determine the similarity between A and C, and between B and D. Those similarities are used to then infer trophic relationships between A and D (using 1 step – the phylogenic/taxonomic similarity between B and D), between C and B (using one step for the same reason), and between C and D (using two steps: between A and C and between B and D).

This is a more complex analysis than is required for the social network analysis in this dissertation. Instead of only one type of relationship, there are two: the trophic and the phylogenic/taxonomic. With two dimensions of relationships, understanding how to compose the connections is less clear. We have begun analyzing lake-based food webs with this tool and are presenting the results to the ecological community for analysis. Eventually, steps will be taken to refine the similarity measures between species and understand how path lengths affect the number of accurate inferred trophic relationships.

Research in this space is underway with Neo Martinez, Jennifer Dunne, and Rich Williams of the Pacific Ecoinformatics and Computational Ecology Lab. We are in the process of a thorough analysis of the results produced from these inference methods. Forthcoming publications include *Webs on the Web: An Ontology for Representing Food Webs and Ecology on the Semantic Web* which has been submitted to the Journal of Web Semantics.

## *10.5 Conclusions and Vision*

In this chapter I have presented several concrete projects that extend my dissertation work , ranging from results that should quickly follow from the existing research, to projects like MOAM that are longer term investigations.

This dissertation project has served as a specific application of the techniques of studying the relationships in a system to understand their functional properties, developing algorithms based on those properties to infer information about indirect relationships, and integrating those inferences into applications. As a long-term project, I believe this type of analysis is interesting and important. Systems that can be subjected to this type of analysis underlie both common user experiences, like social networks, and scientific research problems, such as food webs. By understanding more information about the relationships in the system, and integrating that into applications in a way that supports users, I believe there is great promise to increase the productivity of users and the ability of applications to act intelligently with respect to the underlying systems.

# References

ABDUL-RAHMAN, A. AND HAILES, S. 1997. A distributed trust model. *New Security Paradigms Workshop*. Cumbria, United Kingdom: 48–60.

ABDUL-RAHMAN, A. AND HAILES, S. 2000. Supporting trust in virtual communities. *Proceedings of the 33rd Hawaii International Conference on System Sciences*. Maui, Hawaii, USA.

ABERER, KARL AND ZORAN DESPOTOVIC. Managing trust in a Peer2 -Peer information system. *Proceedings of the Tenth International Conference on Information and Knowledge Management(CIKM01)*, New York, November 5-10 2001: 310-317.

ALSPECTOR, J., KOLCZ, A., AND KARUNANITHI, N. 1998. Comparing feature-based and clique-based user models for movie selection. *Proceedings of The Third ACM Conference on Digital Libraries*. ACM Press, Pittsburgh, Pennsylvania: 11–18.

AMERICAN FILM INSTITUTE, "100 Years, 100 Movies," American Film Institute, http://www.afi.com/tvevents/100years/movies.aspx

ANSPER, ARNE, AHTO BULDAS, MEELIS ROOS, AND JAN WILLEMSON. 2001. Efficient long-term validation of digital signatures. *Advances in Cryptology - PKC 2001.*

AXELROD , ROBERT. 1984. *The Evolution of Cooperation.* New York: Basic Books.

BARBER, KS, J KIM. 2000. Belief Revision Process Based on Trust: Agents Evaluating Reputation of Information Sources. *Lecture Notes In Computer Science*; Vol. 2246: 73-82.

BARNES, J. A. 1972. *Social networks*. Reading, MA: Addison-Wesley.

BETH, T., M. BORCHERDING, AND B. KLEIN. 1994. Valuation of trust in open networks. *Proceedings of ESORICS 94*. Brighton, UK, November 1994.

BONHARD, P. 2004. Improving recommender systems with social networking. *Proceedings Addendum of the 2004 ACM Conference on Computer-Supported Cooperative Work*. Chicago, IL, USA.

BOUTILIER, C., BRAFMAN, R.I., DOMSHLAK, C., HOOS, H.H., AND POOLE, D. 2003. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research (JAIR), 2003.*

BOYKIN, P. O., V. ROYCHOWDHURY. 2004. Personal email networks: an effective anti-spam tool (Preprint), http://www.arxiv.org/abs/cond-mat/0402143

BRICKLEY, D., L. MILLER. 2004. FOAF Vocabulary Specification, Namespace Document, September 2, 2004, http://xmlns.com/foaf/0.1/.

BUSKENS, VINCENT. 2002. *Social Networks and Trust.* Dordrecht, The Netherlands: Kluwer Academic Publishers.

CATTELL, V. 2001. "Poor people, poor places, and poor health: the mediating role of social networks and social capital." *Social Science and Medicine* 52(10):1501-1516.

COOK, KAREN (e.d.). 2001. *Trust in Society*, New York: Russell Sage Foundation.

CORMEN, THOMAS H., CHARLES E. LEISERSON, RONALD L. RIVEST. 1999. *Introduction to Algorithms*. Cambridge, MA: MIT Press.

COSLEY, DAN , SHYONG K. LAM, ISTVAN ALBERT, JOSEPH A. KONSTAN, JOHN RIEDL 2003. "Is seeing believing?: how recommender system interfaces affect users' opinions", *Proceedings of the conference on Human factors in computing systems*. 585-592.

COSMIDES, L., J. TOOBY. 1992. "Cognitive Adaptations for Social Exchange," In *The Adapted Mind: Evolutionary Psychology and the Generation of Culture*, edited by J. H. Barkow, L. Cosmides, J. Tooby, 163-228. New York: Oxford University Press.

DASGUPTA, P. 2000. Trust as a Commodity. In *Trust: Making and Breaking Cooperative Relations*, edited by Diego Gambetta. Electronic edition, Department of Sociology, University of Oxford. http://www.sociology.ox.ac.uk/papers/trustbook.html

DAVIS, GERALD, MINA YOO, WAYNE BAKER. 2003. The Small World of the American Corporate Elite. *Strategic Organization*, 1(3): 301-326.

DAVIS, I., E. VITIELLO .2004. Relationship: A vocabulary for describing relationships between people, March 8, 2004. http://purl.org/vocab/relationship.

DEUTSCH, MORTON. 1962. “Cooperation and Trust. Some Theoretical Notes.” in Jones, M.R. (ed) *Nebraska Symposium on Motivation.* Nebraska University Press.

DEUTSCH, MORTON. 1973. *The Resolution of Conflict.* New Haven and London: Yale University Press.

DEZSO, ZOLTÁN, AND ALBERT-LÁSZLÓ BARABÁSI. .2002. Halting viruses in scale-free networks. *Physical Review E* 65 (055103).

DUMBILL, E. 2002. Finding friends with XML and RDF. IBM's XML Watch.

FOSTER, C. C., RAPOPORT, A., AND ORWANT, C. J. 1963. A study of a large cociogram: Elimination of free parameters. *Behavioural Science* 8:56-65.

FUKUYAMA, F. 1996. Trust: The Social Virtues and The Creation of Prosperity. New York: Free Press.

GAMBETTA, DIEGO. (1990). Can We Trust? In *Trust: Making and Breaking Cooperative Relations*, edited by Diego Gambetta. Electronic edition, Department of Sociology, University of Oxford. http://www.sociology.ox.ac.uk/papers/trustbook.html

GARDEN, MATTHEW, AND GREGORY DUDEK. 2005. Semantic feedback for hybrid recommendations in Recommendz. *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE05*), Hong Kong, China, March 2005.

GARTON, L, C HAYTHORNTHWAITE, B WELLMAN. 1997. Studying Online Social Networks. *Journal of Computer Mediated Communication* 3(1).

GIRVAN, M, AND M NEWMAN. 2002. Community Structure in Social and Biological Networks, *Proceedings of the National Academy of Sciences,* USA.

GOLBECK, JENNIFER. 2002. Evolving Strategies for the Prisoner's Dilemma, *Advances in Intelligent Systems, Fuzzy Systems, and Evolutionary Computation*. February 2002: 299-306.

GOLEMBIEWSKI, ROBERT T., AND MCCONKIE, MARK. 1975. The Centrality of Interpersonal Trust in Group Processes In *Theories of Group* Processes, edited by Cary Cooper. Hoboken, NJ: Wiley.

GRAY, ELIZABETH, JEAN-MARC SEIGNEUR, YONG CHEN, AND CHRISTIAN JENSEN. 2003. Trust Propagation in Small Worlds. *Proceedings of the First International Conference on Trust Management.* LNCS 2692, Springer-Verlag.

GRISHCHENKO, VICTOR S. 2004. Redefining Web-of-Trust: reputation, recommendations, responsibility and trust among peers. *Proceedings of the First Workshop on Friend of a Friend, Social Networking, and the Semantic Web*. Galway, Ireland.

GUHA, R., R. KUMAR, P. RAGHAVAN, AND A. TOMKINS. 2003. Propagation of Trust and Distrust. *Proceedings of the 13th Annual International World Wide Web Conference*, New York, NY.

HARDIN, RUSSELL. 2002. *Trust & Trustworthiness*. New York: Russell Sage Foundation.

HERLOCKER , JONATHAN L., JOSEPH A. KONSTAN , AL BORCHERS , JOHN RIEDL. 1999. An algorithmic framework for performing collaborative filtering. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. August 15-19, 1999, Berkeley, California: 230-237.

HERLOCKER , JONATHAN L., JOSEPH A. KONSTAN , JOHN RIEDL. 2000. Explaining collaborative filtering recommendations, *Proceedings of the 2000 ACM conference on Computer supported cooperative work,* December 2000, Philadelphia, Pennsylvania: 241-250.

HERLOCKER, J, KONSTAN, J, AND RIEDL, J. 2002. An Empirical Analysis of Design Choices in Neighborhood-based Collaborative Filtering Algorithms, *Information Retrieval*, 5 (2002): 287-310.

HERLOCKER , JONATHAN L., JOSEPH A. KONSTAN , LOREN G. TERVEEN , JOHN T. RIEDL. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1): 5-53.

JONES, JAMES HOLLAND, AND MARK S. HANDCOCK. 2003. Sexual contacts and epidemic thresholds. *Nature* 423:605-606.

JONKER, C. AND J. TREUR. Formal analysis of models for the dynamics of trust based on experiences. 1999. *Multi-Agent System Engineering, Proceedings of the 9th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, MAAMAW'99*. Lecture Notes in Artificial Intelligence 1647.

JØSANG, A. The Right Type of Trust for Distributed Systems. 1996. P*roceedings of the 1996 New Security Paradigms Workshop*.

JØSANG, AUDUN, ELISABETH GRAY, MICHAEL KINATEDER. 2003. Analysing Topologies of Transitive Trust, *Proceedings of the First International Workshop on Formal Aspects in Security & Trust (FAST2003)*.

JUNG, Y., A. LEE. 2000. Design of a Social Interaction Environment for Electronic Marketplaces. *Proceedings of Designing Interactive Systems: Processes, Practices, Methods, & Techniques 2000*. 129-136.

KAMVAR, SEPANDAR D. MARIO T. SCHLOSSER, HECTOR GARCIA-MOLINA. 2003. The EigenTrust Algorithm for Reputation Management in P2P Networks. *Proceedings of the 12$^{th}$ International World Wide Web Conference*, May 20-24, 2003, Budapest, Hungary.

KAUTZ, H., B SELMAN, M. SHAH. 1997. Combining Social Networks and Collaborative Filtering. *Communications of the ACM* 40(3): 63-65.

KEENEY, RALPH AND HOWARD RAIFFA. 1976. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Cambridge, UK: Cambridge University Press.

KENT, S., AND R. ATKINSON. 1998. Security Architecture for the Internet Protocol. RFC 2401.

KHARE,R., A. RIFKIN. 1997. Weaving a Web of Trust. World Wide Web Journal, 2(3), pp. 77-112.

KREPS, D.M., R. WILSON. 1982. Reputation and Imperfect Information. *Journal of Economic Theory*. 27: 253-279.

LASHARKI, Y., METRAL, M., AND MAES, P. 1994. Collaborative interface agents, *Proceedings of the National Conference on Artificial Intelligence*. Cambridge, MA: MIT Press.

LEE, SEUNGJOON , ROB SHERWOOD, BOBBY BHATTACHARJEE. 2003. Cooperative Peer Groups in NICE, *IEEE Infocom*, April 2003.

LEVIN, RAPH AND ALEXANDER AIKEN. 1998. Attack resistant trust metrics for public key certification. *7th USENIX Security Symposium*. January 1998, San Antonio, Texas.

MAES, P, R KOZIEROK. 1994. Agents that reduce work and information overload. *Communications of the ACM*. 37(7): 30-40.

MARSH, STEPHEN. 1992. "Trust and Reliance in Multi-Agent Systems: A Preliminary Report" *4th European Workshop on Modeling Autonomous Agents in a Multi-Agent World*. Lecture Notes in Computer Science 830.

MARSH, STEPHEN. 1994. Formalising Trust as a Computational Concept. PhD thesis, Department of Mathematics and Computer Science, University of Stirling.

MASSA, P., P. AVESANI. 2004. Trust-aware Collaborative Filtering for Recommender Systems. *Proceedings of the International Conference on Cooperative Information Systems (CoopIS) 2004*.

MASSA, P., B. BHATTACHARJEE. 2004. Using Trust in Recommender Systems: an Experimental Analysis. *Proceedings of iTrust2004 International Conference*.

MENDES ,S. AND HUITEMA, C. 1995. A new approach to the X.509 framework: Allowing a global authentication infrastructure without a global trust model. *Proceedings of the 1995 Internet Society Symposium on Network and Distributed System Security*.

MAURER, UELI. 1996. Modelling a public-key infrastructure. *Proceedings of Computer Security - ESORICS'96*. Springer-Verlag, 1996.

MCCABE, KA, ML RIGDON, V SMITH. 2003. Positive Reciprocity and Intentions in Trust Games. *Journal of Economic Behavior and Organization*.

MILGRAM, S. 1967. The small world problem. *Psychology Today* 2, 60–67.

MONTGOMERY, J. 1991. "Social Networks and Labor-Market Outcomes: Toward an Economic Analysis." *American Economic Review* 81(5): 1407-1418.

NEJDL, WOLFGANG, DANIEL OLMEDILLA, MARIANNE WINSLETT. 2004. PeerTrust: Automated Trust Negotiation for Peers on the Semantic Web, *Proeedings. of the Workshop on Secure Data Management in a Connected World (SDM'04) in conjunction with 30th International Conference on Very Large Data Bases,* August.-September 2004, Toronto, Canada

NEWMAN, M. E. J. 2001. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*. 98: 404 - 409.

NEWMAN, M. E. J. 2002. The spread of epidemic disease on networks. *Physical Review E* 66 (016128).

NOWAK, M.A., AND K. SIGMUND. 2000. Cooperation versus Competition. *Financial Analyst Journal*, July/August:13-22.

PAGE, L., BRIN, S., MOTWANI, R., & WINOGRAD, T. 1998. The PageRank citation ranking: Bringing order to the web. *Technical Report 1998*, Stanford University, Stanford, CA.

PAGEL, M., W. ERDLY, J. BECKER. 1987. Social networks: we get by with (and in spite of) a little help from our friends. *Journal of Personality and Social Psychology* 53(4):793-804.

PERNY, P. AND J. D. ZUCKER. 2001. Preference-based Search and Machine Learning for Collaborative Filtering: the ``Film-Conseil'' recommender system. *Information, Interaction , Intelligence,* 1(1):9-48.

POLLOCK, G. B., L. A. DUGATKIN. 1992. Reciprocity and the Evolution of Reputation. *Journal of Theoretical Biolog*y. 159: 25-37.

PREECE, J. 2000. *Online Communities: Designing Usability, Supporting Sociability*. Chichester, UK: John Wiley & Sons.

REITER ,M.K. AND STUBBLEBINE, S. G. 1998. Resilient authentication using path independence. *IEEE Transactions on Computers*. 47, 12 (Dec.): 1351–1362.

REITER ,M.K. AND STUBBLEBINE, S. G. 1999. Authentication Metric Analysis and Design. *ACM Transactions on Information and System Security*. 2(2): 138-158.

RICHARDSON, MATTHEW, RAKESH AGRAWAL, PEDRO DOMINGOS. 2003. Trust Management for the Semantic Web. *Proceedings of the Second International Semantic Web Conference*. Sanibel Island, Florida.

SHANKAR N., ARBAUGH W. 2002. On Trust for Ubiquitous Computing. *Workshop on Security in Ubiquitous Computing, UBICOMP 2002*, Gteborg Sweden.

SHAPIRO, SUSAN. 1987. Social Control of Impersonal Trust. *The American Journal of Sociology*, 93(3): 623-658.

SHNEIDERMAN, B. 2000. Designing websites to enhance online trust. *Communications of the ACM*. 43(12): 81-83.

SINHA, R., AND SWEARINGEN, K. 2001. Comparing recommendations made by online systems and friends. *Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries*. June, 2001, Dublin, Ireland.

SWEARINGEN, K. AND R. SINHA. 2001. Beyond algorithms: An HCI perspective on recommender systems. *Proceedings of the ACM SIGIR 2001 Workshop on Recommender Systems*. New Orleans, Louisiana.

STUBBLEBINE, S. 1995. Recent-Secure Authentication: Enforcing Revocation in Distributed Systems. *Proceedings of the1995 IEEE Symposium on Research in Security and Privacy,* May, 1995, Oakland, California: 224-234.

SZTOMPKA, PIOTR. 1999, *Trust: A Sociological Theory*, Cambridge: Cambridge University Press.

TARAH,A. AND HUITEMA, C. 1992. Associating metrics to certification paths. *Computer Security*. 175–189.

USLANER, E. 2002. The Moral Foundations of Trust. Cambridge, UK: Cambridge University Press.

WALLACH, DAN S., DIRK BALFANZ, DREW DEAN, EDWARD W. FELTEN. 1997. Extensible Security Architectures for Java. *Sixteenth Symposium on Operating Systems Principles.*

WATTS, D. 1999. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton, NJ: Princeton University Press.

WATTS, D. AND S. H. STROGATZ. 1998. Collective Dynamics of Small-World Networks. *Nature* 393(1998):440-442.

WASSERMAN, S., & FAUST, K. 1994. *Social network analysis: Methods and applications*. Cambridge: Cambridge University Press.

WELLMAN, B. 1982. Studying personal communities. *Social structure and network analysis,* edited by P. Marsden & N. Lin, 61-80. Beverly Hills, CA*:* Sage.

YANIV, I., E. KLEINBERGER. 2000. Advice taking in decision making: Egocentric discounting and reputation formation. *Organizational Behavior and Human Decision Processes*. Nov; 83(2):260-281

ZIEGLER, C.N., LAUSEN, G. 2004(a). Spreading activation models for trust propagation. *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service*, Taipei, Taiwan.

ZIEGLER, CAI-NICOLAS, GEORG LAUSEN. 2004(b). Analyzing Correlation Between Trust and User Similarity in Online Communities. *Proceedings of Second International Conference on Trust Management.*