DISSERTATION
submitted
to the
Combined Faculty for the Natural Sciences and Mathematics
of
Heidelberg University, Germany
for the degree of
Doctor of Natural Sciences

Put forward by

Master.Sw.E: DUAN, HUIYING

Born in: CHONGQING, CHINA

Oral examination: _____

Trust Building and Managing in Service-oriented Environment

Advisor: Prof. Dr. Andreas Reuter

Abstract

Services are ubiquitous. In daily life, we can find service provisioning everywhere such as online shopping, online storage, doctor, hotel, lawyer, restaurant, etc. With the development of Web 2.0 technology, a huge amount of information about services has become available on the Internet. For instance, on a review website people can discuss which restaurant serves the best Chinese food; in a blog, an author posts an article about the experience of visiting a doctor. The abundance of services and the overload of service information online, result in two main problems. The first problem is service selection; the second one is the overload of consumer-driven information which refers to information such as reviews, articles, assessments, and discussions generated by service consumers.

The concept of trust is proposed to solve the two problems. The computational concept of trust is defined as a subjective probability, which makes a prediction of the occurrence of an event such as a good service provisioning. Software used for building and managing trust data related to service offerings, is called Trust Management System (TMS). The first topic is trust model. A trust model is the computing kernel of a TMS that calculates the trust value of a service. Another significant topic regarding trust management for services is the robustness of a TMS. Robustness of a TMS refers to the ability of a TMS to cope with inaccuracy (deliberate or accidental) in the consumer-provided information used for computing trust. There are many trust models that have been proposed. I do not know of any survey analyzing and comparing different trust models with respect to trust in services. In this thesis, 40 trust models are compared from both a theoretical and a practical perspective, using criteria such as application context, information representation, properties of trust evaluation, and robustness of system. In addition, a trust model framework for service provisioning is proposed. This framework is considered a meta-model covering all existing trust models. A concrete trust model can be derived by instantiating the meta-model.

In the thesis, four concrete services which cover both quantitative and qualitative services are studied. A quantitative service refers to a service the quality of which can be measured objectively. For a qualitative service there is no general agreed-upon objective measure for service quality. The first case study is about Online File Storage Service (OFSS) which is categorized as a quantitative service. The trust model, R-Rep, for a OFSS is proposed. In order to mitigate manipulation, a statistics based detection mechanism,

named Baseline Sampling (BS), is introduced. In addition, when social network information among users is available, Clique Identification (CI) is used to detect manipulative groups. One e-commerce website, Taobao.com, and two review websites, TripAdvisor.com and Dianping.com, are chosen as case studies for trust building and managing in the context of qualitative service. For each case, specific trust models which consider intrinsic robustness enhancement by designing special weight functions are proposed. Meanwhile, machine learning-based extrinsic robustness enhancement is applied. Three types of machine learning approaches, clustering, classification and Annotation-Auxiliary Clustering (AAClust), are applied to identify manipulative behavior.

Abstract

Dienstleistungen sind allgegenwärtig. Im alltäglichen Leben finden wir Dienstleistungen überall, wie beim Internetshopping, Online-Datenspeicherung, Arzt, Hotel, Anwalt, Restaurant, etc. Mit der Entwicklung des Web 2.0 wurde eine riesige Menge an Daten über Dienstleistungen verfügbar. Zum Beispiel können Nutzer auf einer Bewertungsseite diskutieren welches Restaurant die beste deutsche Küche serviert; in einem Blog postet ein Autor seine Erfahrungen eines Arztbesuchs. The Vielzahl von Dienstleistungen und das Überangebot an Serviceinformationen im Internet, resultiert in zwei Hauptproblemen. Das erste Problem ist die Serviceauswahl; das zweite Problem ist das Überangebot an Onlineinformationen die von Konsumenten bereitgestellt wurden, und Reviews, Artikel, Bewertungen, und Diskussionen umfassen.

Das Konzept des Vertrauens (Englisch: "Concept of Trust") wird vorgeschlagen diese beiden Probleme zu lösen. Das rechentechnische Konzept des Vertrauens wird definiert als eine subjektive Wahrscheinlichkeit, die eine Vorhersage über das Eintreten eines Ereignisses, wie die einer guten Dienstleistung, trifft. Software die eingesetzt wird um Daten zu Vertrauen aufzubauen und zu verwalten wird Vertrauensverwaltungssystem (Englisch: "Trust Management System" (TMS)) genannt. Der erste Aspekt, den wir untersuchen ist das Vertrauensmodel. Das Vertrauensmodel ist der rechnerische Kern eines TMS, das den Vertrauenswert einer Dienstleistung berechnet. Ein anderer wichtiger Aspekt die Vertrauenswürdigkeit von Dienstleistungen betreffend ist die Robustheit eines TMS. Robustheit eines TMS bezieht sich auf die Fähigkeit eines TMS mit Ungenauigkeiten (beabsichtigte oder unbeabsichtigte) in von Konsumenten bereitgestellten Informationen, die für die Berechnung von Vertrauen benutzt werden, umzugehen. Es existiert bereits eine Vielzahl von Vertrauensmodellen. Uns ist keine Studie bekannt, die die verschiedenen Vertrauensmodelle hinsichtlich ihrer Eignung fr die Untersuchung von Vertrauen in Dienstleistungen analysiert und vergleicht. In der vorliegenden Arbeit werden 40 Vertrauensmodelle aus theoretischer und praktischer Sicht vergleichen, unter Anwendung verschiedener Kriterien, wie Anwendungsbereich, Informationsdarstellung, Eigenschaften der Vertrauensbewertung, und Robustheit des Systems. Desweiteren schlagen wir einen Vertrauensmodel Rahmenkonzept für Dienstleistungen vor. Dieses Rahmenkonzept kann als Metamodel aller existierenden Vertrauensmodelle betrachtet werden. Ein konkretes Vertrauensmodel kann durch Instanziierung des Metamodels hergeleitet werden.

In der vorliegenden Arbeit wurden vier konkrete Dienstleistungen, die sich von quantitativen bis qualitativen Dienstleistungen erstrecken, untersucht. Eine quantitative Dienstleistung beschreibt Dienstleistungen deren Qualität objektiv bewertet werden kann. Für eine qualitative Dienstleistung existiert keine allgemein akzeptierte Messung der Qualität. Die erste Fallstudie befasst sich mit Onlinedatenspeicherungsdienstleistungen ODSD, welche als quantitative Dienstleistung eingeordnet werden. Ein Vertrauensmodel R-Rep für ODSD wird vorgeschlagen. Um Manipulation zu vermeiden führen wir einen statistisch basierten Erkennungsmechanismus ein, das Baseline Sampling. Wenn Informationen aus sozialen Netzwerken über Nutzer verfgbar sind, schlagen wir zusätzlich das Cliquen Identifizieren vor, um manipulierende Gruppen zu identifizieren. Eine Elektronische-Handels-Webseite, taobao.com, und zwei Bewertungswebseiten, TripAdvisor.com und Dianping.com, wurden als Fallstudien zu Vertrauensbildung und Verwaltung im Kontext der qualitativen Dienstleistungen ausgewählt. Für jeden Einzelfall spezifische Vertrauensmodelle werden vorgeschlagen, die intrinsische Robustheitsverbesserung durch den Einsatz spezieller Gewichtungsfunktionen berücksichtigen. Weiterhin wird extrinsische Robustheitsverbesserung, das auf Maschinenlernen basiert, angewandt. Drei Arten von Maschinenlernen-Ansätzen werden angewandt um manipulatives Verhalten zu identifizieren, und zwar Clusterbildung, Klassifizierung und Annotations-Auxiliär-Clusterbildung.

To my family

Acknowledgements

I would like to express my special appreciation and thanks to my parents who support me all the time. We share the same dream that one day I can study as a Ph.D candidate and finally finish the study. Sometimes I can feel that the determination they have is even larger than what I have. During my whole study, my parents have done what they can do best in order to help me to achieve our dream. They can not help me to solve an equation or to analyze experimental results. What they behave is like a coach who teaches me how to deal with difficulties and complicated situations mentally. The most significant inspiration they deliver to me is persistence and determination. Without their encourage, I cannot reach this phase.

I owe my deepest gratitude to my advisor Professor Dr. -Ing. Dr. h.c. Andreas Reuter. I would like to thank you for choosing me as your Ph.D student and for allowing me to grow as a research scientist. I can still remember the great moment when I received the admission letter from your previous secretary Mrs. Neu and the first time we met in your office at TU Kaiserslautern. Your way of supervising is concise and to the point which gives me free space for developing original ideas. In addition, I would like to thank Mr. Klaus Tschira and the Klaus Tschira Foundation for providing me scholarship and I would like also to thank Heidelberg University and the faculty of Mathematics and Computer Science for accepting me as a Ph.D student.

I would like to offer my special thanks to Prof. Dr. Michael Strube and Prof. Hendrik Speck who give me a lot of hints for my research work. I am glad to work with Cäcilia Zirn on the research about tripadvisor.com. I also want to thank Dr. Jonathan Fuller for proofreading my previous journal article and this thesis draft. I would like to give credit to my research assistants Feifei Liu, Peng Yang, Mao Ye, Bai-Cheng Jim and Andrea Maier for analyzing and annotating data. My fiancee Yakun Zhou has also annotated some data and I give the very special thank to her. I owe an important debt to Shan Lu, Wenchan Jiang, Lejing Wang, Yu Bai, Yu Huang, Hongyao Zhao for helping me to execute an experiment about Online File Storage Service (OFSS). I would also like to thank my previous and current colleagues Xiaofeng Xia, Nikolas Nehmer, Frank Böhr and Martin Größl, and my best friends in Germany Yu Bai, Xiaofeng Xia, Xiang Yang, Xuefeng Zhang, Bo Zhang, Hao Li, Yi Ou and Yong Hu who give me a lot of help during my Ph.D study.

I would like to show my greatest appreciation to Mrs. Heike Neu who is the secretary of our research group in TU Kaiserslautern. From my point of view, she is the best secretary I have ever met. I would also like to thank the International School for Graduate Studies (ISGS), especially the previous/current working staffs in there Wolfgang Reisel, Arthur Harutyunyan and Heike Döring. Without the help of Mr. Reisel, maybe I can not even find where the university is on the first day when I came to Germany.

Thanks all of you who ever helped, encouraged and even frustrated me and my Ph.D work. Whatever you have done contribute to my work and this dissertation.

Karlsruhe 06.09.2014

Huiying Duan

Contents

List of Figures

LIST OF FIGURES

LIST OF FIGURES

List of Tables

Glossary

AAClust annotation-auxiliary clustering; a semi-unsupervised machine learning approach for detecting manipulative behavior

AMT Amazon mechanical turk;

ATT attributes set; a set of refined attributes of a service

BFS breadth-first search;

BO bandwidth offset;

BRS beta reputation system;

BS baseline sampling;a technique used for detecting and filtering manipulative behavior

BVR benefit variation ratio;

CAPTCHA Completely Automated Public Turing test to tell Computers and Humans Apart; a type of challenge-response test used in computing to determine whether or not the user is human

CI clique identification;a social-network-based technique used for detecting manipulative behavior

CRQ confidence for rating quantity at the query level;

CRS commonly rated services;

CRS confidence for rating quantity at the system level;

CSI clustering-based suspect identification; a clustering approach for detecting suspicious vendors and suspicious customers in a Chinese e-commerce website Taobao.com

CSQ confidence for referral similarity at the query level;

CSS Confidence for referral similarity at the system level;

CTQ confidence for transitivity in the query level;

CTS confidence for transitivity at the system level;

DMB demoting manipulative behavior;

DUP dishonest user proportion;

FTF file transfer frequency;

HNO Hanoi;capital of Vietnam

IAB Internet access bandwidth;

ISP Internet service provider; an organization that provides services for accessing, using, or participating in the Internet

MaxDev maximum deviation;

NFO number of failure offset;

NYC new york city;

OFSS online file storage service; an Internet hosting service, which is particularly designed to host user files

P2P peer-to-peer;

PDF probability density function;

PMB prmoting manipulative behavior;

PMF probability mass function;

PS preference structure; a relation on an n-dimensional service evaluation space, establishes an order among the points in this space

RCI ranking comparison index;

RPS ratio of promoter to slanderer;

SaaS software as a service;

SDM suspicion degree meter; an unsupervised learning approach for detecting suspicious reviewers and hotels in tripadvisor.com.

SID suspicion index for demoting;

SIP suspicion index for promoting;

SVMs support vector machines; supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis

TA tripadvisor.com; a hotel reviewing website.

TM trust model;

TMS trust management system; any combination of information technology and activities that support trust building, managing and decision making

TV trust value;

tw time window;

VF value function; a mapping from the domain of trust evaluation to a satisfaction space

1

Introduction

1.1 Problem Statement

Services are ubiquitous. In daily life, we can find service provisioning everywhere. Consider a day in the life of Tim. In the morning, he takes bus to work, which belongs to a transportation service. He works as a receptionist in a hotel. After work, he goes to a restaurant with his girlfriend. Over dinner they decide to order prints of their photos online. After he comes back home, he checks his e-mail and finds information about a shoe sale at Amazon.com. Afterwards he finds that a tap in the bathroom is broken. He wants to call a company to repair it. Within those few sentences sketching a fairly typical day in modern life, as many as seven types of service have been mentioned. Services, digital and otherwise, are ubiquitous; they are indispensable part of the fabric of our modern societies.

In addition, with the development of Web 2.0 technology, a huge amount of information about services has become available on the Internet. On many websites people can discuss which restaurant serves the best Chinese food; in a Business-to-Consumer e-commerce website like Amazon.com[1], people look for an online shopping (retail) service which provides them with products; in a blog, an author posts an article about the experience of visiting a doctor. On first approximation, the information about services can be categorized into two types: provider-driven information and consumer-driven information. Provider-driven information aims at propagating a service; the provider wants customers. A typical example of provider-driven information is advertisement. For instance, a service provider creates the official website for a service and sends an advertisement via e-mail. In an advertisement, usually the promised features and quality of a service are introduced. Consumer-driven information, on the other hand, refers to information such as reviews, articles, assessments, and discussions generated by service consumers. One typical type of consumer-driven information is a review. For example, on e-commerce websites like eBay.com and Amazon.com, one can submit a review

[1] Amazon.com, www.amazon.com, is an e-commerce platform, where users buy products or services from a vendor over the Internet.

including a rating (positive or negative) after buying a product provided by an online retail service.

The abundance of services and the overload of service information online, result in two main problems. The first problem is service selection. There were more than two million third-party sellers and 188 million active customers of Amazon in 2012; more than 100 million active users on eBay in 2011; more than one million hotels and motels worldwide in TripAdvisor.com[1]. The service space is so large that people find it difficult to choose the best service. If you plan to travel New York City in the following week and there are over 400 hotels in the city, which hotel should you choose? The second problem is the overload of consumer-driven information. One cannot read all the reviews in order to compare the services. For instance, usually there are over one hundred hotels in a big city. Each hotel has between tens and thousands of reviews in one reviewing website. Additionally, there is more than one source providing information for the same service. People find the information about the same service in different websites. All this adds up to (yet another case of) information overload. But since we vitally depend on the use of services, and since we do not want to select them randomly, the question is clear. How can we structure the vast amount of service-related information effectively and to our benefit?

The key approach we want to introduce and analyze in this thesis is the concept of trust. It has been studied and investigated in many branches of science such as, but not limited to, sociology, philosophy, psychology, economics, political science, management, and computer science (1, 2, 4). The computational concept of trust is defined as a subjective probability (4), which makes a prediction of the occurrence of an event such as a good service provisioning. Moreover, trust is not just a probability estimate, but it assigns different weights to different pieces of consumer-driven information. In classic statistics, all information pertaining to the phenomenon under consideration (e.g. observations, evidences and experiences) are treated with the same likelihood. In our context, however, we need to consider the notion of relevance, too. For instance, for computing a trust value of a hotel service, a review posted five years ago is considered less important than a review posted recently. The quality of a hotel may change over time. This phenomenon is called dynamics of service quality. The main advantage of using computational trust is that, the huge collection of consumer-driven information is mapped into a numerical value or vector. A single value is understandable since a numerical value can indicate the quality of a service. In addition, when the service quality is refined into attributes, for each attribute of service quality, one numerical value is calculated as a trust measurement. If there are n attributes of a service, the trust of the service corresponds to an n dimensional vector. Hence, by computing the trust value or vector of a service from a large amount of consumer-driven information, the comparison of services is transformed into the comparison of numbers or vectors. The large amount of information for a service is compressed into a single piece of information.

Software used for building and managing trust data related to service offerings, is called Trust Management System (TMS). TMS refers to any combination of information

[1]TripAdvisor.com, www.tripadvisor.com, is a review website, where users can give ratings and reviews of travel-related services such as hotels, flights, restaurants, etc.

technology and activities that support trust building, managing and decision making. The most significant component of a TMS is a trust model. A trust model is the computing kernel of a TMS that calculates the trust value of a service. There are many different trust models that have been proposed since 1997 (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50). They differ in aspects such as application context, information representation, properties of trust evaluation and robustness of the system. For instance, EigenTrust (10) applies transitive relation of trust to a Peer-to-Peer file sharing application; PeerTrust (22) gives more weight to transactions with a higher price. Yet there is no survey that analyzes and compares different trust models regarding the trust in services. More importantly, a meta-model of trust models is necessary to generalize and summarize all the proposed trust models.

Another significant issue regarding trust management for services is the robustness of a TMS. Robustness of a TMS refers to the ability of a TMS to cope with inaccuracy (deliberate or accidental) in the consumer-provided information used for computing trust. The robustness issue is even more urgent than the design of a trust model, since as a special type of Information System, a TMS is useless when it delivers inaccurate results. There are two main reasons causing evaluation inaccuracy: data sparsity and attack. Data sparsity refers to the state of a TMS where there is not enough data for evaluating trust. There are only a few works that consider data sparsity (4, 18). The second reason for evaluation inaccuracy is the possibility of compromising a TMS by feeding it fraudulent information (51, 52). Such attacks can be classified as purposeless attack (system destruction) and purposeful attack (manipulation). The difference between the two types of attack is whether the intention of the attack is explicit or not. System destruction is extremely difficult to identify since so far, there has been no explicit intention to develop an identification mechanism. Regarding manipulation, there are two types of counter-approaches: intrinsic robustness enhancement and extrinsic robustness enhancement. Intrinsic robustness enhancement refers to a trust model having a robustness enhancement feature to mitigate negative influences from attempted manipulation. Extrinsic robustness enhancement refers to mechanisms which are not part of a trust model but an important component of a TMS aiming at eliminating or identifying manipulation.

1.2 Contributions

In this section we list the main contributions of the thesis to trust evaluation and management as follows.

- We give a definition of trust in service-oriented systems. In our work, trust is a computational concept. Trust is a special type of expected value which predicts service quality. Meanwhile the definition of trust is not as same as that of an expected value in classic probability theory, since a trust model assigns different weights to different observations[1]. Note that in classic probability theory,

[1]In this work, observation, experience, evidence and rating are equivalent items, referring to a piece of information which is used as an input for trust evaluation, and all are used interchangeably.

an observation is called an outcome and unlike in trust evaluation all possible outcomes in a sample space are treated equally.

- The basic idea of evaluating trust is to calculate the trust value via aggregating observations by giving different weights to different observations. In order to create weight functions, properties of trust evaluation, such as transitivity, personal experience, recommendation, preference structure, etc., are considered as key criteria.

- Although there is a wide variety of trust models that have been proposed since 1997 (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50), we do not know of any survey analyzing and comparing different trust models with respect to trust in services. In the present thesis, 40 trust models are compared from both a theoretical and a practical perspective, using criteria such as application context, information representation, properties of trust evaluation, and robustness of system.

- We propose a trust model framework for service provisioning. This framework is considered a meta-model covering all existing trust models. A concrete trust model can be derived by instantiating the meta-model. The key step of instantiating the meta-model is to design weight functions. The design of a weight function is subject to context such as the application domain and properties of trust evaluation.

- We propose a number of confidence metrics at both system and query level to solve the problem of data sparsity. The system level concerns the confidence from a TMS perspective, while the query level focuses on how much confidence an evaluator has with respect to trust evaluation of a service. By considering both trust value and the corresponding confidence degree, one can make a better service selection.

- We choose an Online File Storage Service (OFSS) as a case for studying trust building and managing in the context of a quantitative service. A quantitative service refers to a service the quality of which can be measured objectively. An OFSS is an Internet hosting service, which is particularly designed to host user files. One quality measure in an OFSS could be the speed of uploading a certain amount of data. We propose a trust model, R-Rep, for an OFSS. In order to mitigate manipulation, we introduce a statistics based detection mechanism, Baseline Sampling (BS). In addition, when social network information among users is available, we propose Clique Identification (CI) to detect manipulative groups.

- One e-commerce website, Taobao.com[1], and two review websites, TripAdvisor.com and Dianping.com[2], are chosen as case studies for trust building and managing in the context of qualitative services. For a qualitative service there is no general agreed-upon objective measure for service quality. Consider, for example, the rating for a hotel, which is a natural number in the range one to five. Different people give different ratings for the same hotel based on their preferences. For each case, we propose trust models which consider intrinsic robustness enhancement by designing special weight functions. Meanwhile, machine learning-based extrinsic robustness enhancement is applied. Three types of machine learning approaches, clustering, classification and Annotation-Auxiliary Clustering (AAClust), are applied to identify manipulative behavior.

1.3 Organization

The remainder of the thesis is organized as follows.

Chapter 2 introduces the background of trust building and managing for service provisioning. In this chapter different definitions of trust are introduced. Regarding trust evaluation, properties of trust evaluation are introduced such as transitivity and evaluation confidence. 40 trust models are compared from both theoretical and practical perspectives using the criteria such as application context, information representation, properties of trust evaluation and robustness of system. The state-of-the-art regarding extrinsic robust enhancement is described.

Chapter 3 provides the definition of trust evaluation on service provisioning. Both a trust model framework and confidence evaluation metrics are proposed. All the trust models can be generated by instantiating the framework and implementing the corresponding evaluation metrics.

Chapter 4 introduces a case study on trust building and managing for a quantitative service, an Online File Storage Service (OFSS). The corresponding trust models are proposed. Considering statistical differences between honest and dishonest users, a manipulation detection method, Baseline Sampling (BS) is proposed. In addition, when social network information among users is available, we provide Clique Identification (CI) to detect the manipulative group. The design of a simulation testbed and the simulation results are given.

Chapter 5 introduces three case studies for trust building and managing for qualitative services. The first case study is a Chinese e-commerce website, Taobao.com. In order to mitigate manipulation, both intrinsic and extrinsic robustness enhancement techniques are applied. Intrinsic robustness enhancement is applied, where advanced trust models are instantiated from the trust model framework by considering basic assumptions about the suspects behavior. Extrinsic robustness enhancement, namely a manipulation detection system, Clustering-based Suspect Identification (CSI), is proposed to detect customers who provide manipulative ratings, and vendors who intend

[1]Taobao.com, www.taobao.com, is the biggest e-commerce platform in China.

[2]Dianping.com, www.dianping.com, is a Chinese reviewing website, where users give ratings and reviews of lifestyle-related services such as restaurants, shops, lawyers, doctors, home services, etc.

to manipulate their reputation value. After identifying the suspects (suspicious vendors and suspicious customers), we explore characteristics of suspects intensively by comparing statistics of the whole population to the suspicious sub-population.

The second case study is of a travel review website, TripAdvisor.com, where the travel-related services such as hotels and flights are reviewed. In order to mitigate manipulation, both intrinsic and extrinsic robustness enhancement techniques are applied. We consider the manipulation at three different levels, the review level, the reviewer level and the service level. Regarding extrinsic robustness enhancement, both a supervised learning approach, Support Vector Machines (SVMs), and an unsupervised learning approach, Suspicion Degree Meter (SDM), are applied. SDM assigns a real number to every object at each level. Regarding intrinsic robustness enhancement, time-window-based, time-decay-based and suspicion-index-based trust models are proposed to enhance the robustness of TMSs. After identifying suspects, the statistical character of the suspicious sub-population and innocent sub-population are compared.

The third case study is a review website, Dianping.com, where a variety of lifestyle-related services such as restaurants and home services are reviewed. We propose an advanced clustering approach, Annotation-Auxiliary Clustering (AAClust), to identify reviewers suspected of manipulation. In order to base the identification of manipulative behavior on broader knowledge, we explore social network information for innocent and suspicious reviewers.

Chapter 6 provides a conclusion and perspective for the problems that we have addressed. We propose a general solution to remaining problems.

2

Background

2.1 Diversity of Trust Definition

The concept of trust has been studied and investigated in many branches of science such as, but not limited to, sociology, philosophy, psychology, economics, political science, management, computer science. (1, 2, 4)

In order to answer the basic but unsettled question of what the word "trust" means, Mcknight and Chervany investigate sixty research articles or books mainly about management, sociology, economics, political science and psychology (1). The phenomenon that, "researchers are still far from a consensus on what trust means", is called conceptual confusion (1). The authors argue that,"narrow definitions of trust do not accurately depict the concept's rich set of meaning". In order to make a systematic comparison, the authors define categories of trust construct types and perceived attributes of the trusted party, and observe to which specified labels a certain article or book corresponds. Mcknight and Chervany find that trust is most often defined in terms of expectations or beliefs. A large number of definitions refer to trust as behavior. The most frequently mentioned attributes are benevolence, competence, good intentions and honesty. In addition, the paper proposes two kinds of trust typologies:

a) a classification system for types of trust, and

b) definitions of six related trust types that form a model which is shown in Fig. 2.1.

The authors argue that"Trusting Behavior is the extent to which one person voluntarily depends on another person in a specific situation with a feeling of relative security, even though negative consequences are possible. Depends is a behavioral term, which distinguishes Trusting Behavior from Trusting Intention (willingness to depend)". And "Trust Intention is based on the person's confidence in beliefs about the other person".

Stephen Marsh's work (2) is the first PhD thesis which formalizes trust as a computational concept. Instead of discovering cross-disciplinary trust building blocks, Stephen's main contribution is to create a computational trust model which can be

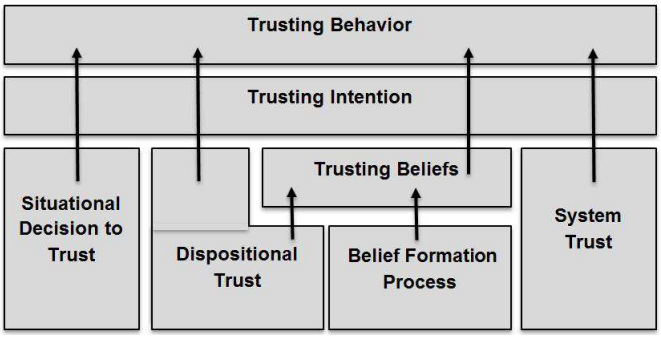

Figure 2.1: Relationships among trust constructs (1)

implemented in the domain of Multi-Agent Systems (MAS). In every part of the thesis, the trace of trust-definition diversity can be sensed. Stephen states that, "trust is a judgement of unquestionable utility". However, trust can be categorized as hope, despair, confidence, innocence and impulsiveness as well (53, 54). Deutsch (55) states that,

a) an individual is confronted with an ambiguous path, a path that can lead to an event perceived to be beneficial or to an event perceived to be harmful;

b) he perceives that the occurrence of these events is contingent on the behavior of another person; and

c) he perceives the strength of a harmful event to be greater than the strength of a beneficial event. If he chooses to take an ambiguous path with such properties, he makes a trusting choice; else he makes a distrustful choice.

Luhmann's main thesis (56) is that trust is a means for reducing the complexity of society. He considers it as a "basic fact of human life". Interestingly, trust is not only a basic fact of human life, but also is that of animal life. Harcourt shows that, when vampire bats have had a good night and a surplus of blood, they feed those who have not. They cooperate in this way such that they can be fed when they don't get enough blood on another night (57). In Stephen's model, a number of concepts are formalized, such as agent, situation, knowledge, basic trust, general trust in agents, situational trust in agents, utility, importance of a situation and time. There is a positive threshold and a negative one specifying the bottom line for "trusted" and "distrusted" respectively, which is illustrated in Fig. 2.2.

Audun Jøsang's survey (4) contains a large volume of information regarding Trust and Reputation Systems for online service provision. Jøsang provides two versions of trust definition: reliability trust and decision trust. Reliability trust is defined as "the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends". The definition of decision trust is the same as that of Trusting Behavior (1). In fact, the two definitions are nothing but a re-iteration of Gambetta's definition (58) and Trusting Behavior (1). The main

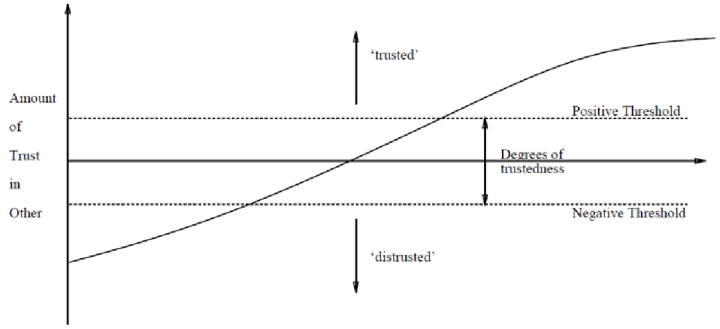

Figure 2.2: Positive and negative thresholds for trust (2)

difference between the two is that, decision trust considers risk but reliability trust does not. Gambetta's definition is the most powerful definition, which transforms concepts like belief, expectancy or competence into a probability which can be computed. We believe that, Gambetta's trust definition is a key building block for trust and reputation models used in service provisioning.

2.2 Properties of Trust Evaluation

Leaving the ultimate definition of trust aside, we consider trust as a computational concept. By applying some computational mechanisms, trust can be represented by a mathematical quantity, which is either a value or a vector. The basic idea is to evaluate trust via aggregating experiences (ratings, observations or evidences) by giving different weights to different experiences. In the present thesis, the terms experience, rating, observation and evidence are used exchangeably. In order to create weight functions, properties of trust evaluation are the key criteria to consider. We list all the important properties as follows:

a) Transitivity. Regarding trust as a binary relation between a truster and a trustee, transitivity is the most significant property to consider. A classic description of this property is that "...if A has trust u in B and B has trust v in C, then A should have some trust t in C that is a function of u and v" (5). The function of u and v can be implemented as $u * v$ in the usual case. Transitivity is so powerful that many trust models are based on this property (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16). However, it is "context-sensitive". Under certain semantic constraints (17), trust can be transitive. Some models such as Eigentrust (10) succeed in bypassing the constraint via considering trust to be universal, which means that "A trusts B" expresses A's unrestricted trust in B for doing anything in the target system.

b) Reflexivity. This basically says that you should trust yourself in the first place. However, it is not a very useful property in terms of computation since we are interested in trust among different entities such as service consumers and providers.

c) Asymmetry. This property states that "A trusts B" does not imply that B trusts A. If trust relations were symmetry, the trust of B in A could be derived from the trust of A in B and vice versa. On the contrary, asymmetry is not very useful property in terms of computation.

d) Personal experience. If a consumer used a service before, personal experience is taken into account for evaluating trust.

e) Recommendation. Second-hand information such as recommendation is indispensable when there is no or not enough personal experience. The usage of recommendation appears very often when online service provisioning takes place (4). Most trust models can handle this problem. In particular, Recommender Systems (18, 27) provide typical solutions for a user to find recommendations from previous users. Note that in comparison to recommendation (second-hand information), personal experience should be given more weight (18, 19, 20, 21, 22, 23, 24, 25, 26).

f) Experience context. Whether an experience is second-hand or not, the context of an experience is significant as well (8, 19, 22, 28). A typical example is the size of a transaction which can be represented by the monetary cost (28).

g) Preference structure. Different trusters evaluate trust of the same trustee differently since they might have had different personal experience (22, 25, 26, 27). Even though there is no personal experience with the trustee, different trusters might end up with different values of trust due to different personal tastes. The preference structure is rendered to characterize the trusters' differences. Furthermore considering trust is a multi-dimension notion, one can come up with a weighted sum to derive a single trust value from the corresponding trust vector, in which each entry represents the trust in a certain dimension. The setting of weights given to different dimensions is different from truster to truster and this fact is also captured by preference structure.

h) Dynamics of service provisioning quality. Most trust models consider older experiences less relevant than recent ones (15, 19, 21, 22, 23), because the quality of service provisioning may vary over time. Trust measures should capture the dynamics by creating an appropriate weight function.

i) Evaluation confidence. This is a measure for the "quality" of evaluation used as a basis for computing a trust value or a trust vector (19, 24, 25, 27, 31). Such a measure is important because in practice many trust and reputation systems suffer from the problem of data sparsity (4, 18). This reflects that fact that in many situations one has only a few evaluations on a small number of services from a (potentially) large inventory.

j) Feedback mechanism. Feedback is a process is which information about the past or the present influences the phenomenon under consideration in the present or future. Considering a trust model as a key component of a Trust Management System (TMS), the feedback mechanism (21, 30, 32) is a candidate to evaluate the accuracy of a trust model and resist manipulation (4, 51). This will be discussed in more detail later in the thesis.

2.3 Related Trust Models

A trust model, i.e. a computational model for evaluating trust, plays a key role in Trust Management Systems (TMSs). Variety large number of trust models has been proposed since 1997. We select 40 trust models (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38) that from our perspective covers the whole spectrum and categorize them according to their scope and the techniques applied. In addition, a number of web sites, such as Ebay.com, Amazon.com, TripAdvisor.com, etc., integrate trust management into their systems in order to provide an indicator for service selection. In this section the 40 trust models are compared from both theoretical and practical perspectives using the criteria such as application context, information representation, properties of trust evaluation and robustness of system.

2.3.1 Comparison of Trust Models with respect to Application Context

Different trust models have been proposed for different application types. This is an important criterion which has usually been overlooked in previous work. Most trust models are derived from a specific application domain such as file sharing in the P2P communities, or e-commerce, or web services, etc. Trust is introduced into the domain to solve a specific problem such as agents' cooperation (2, 26), software components selection (37), movie or product recommendation (18, 27), etc. But as a matter of fact, trust is application-context sensitive, because an application domain endows trust with semantics and properties. In other words, trust is instantiated within an application domain. Afterwards, trust can be defined and the corresponding trust model can be created.

Table 2.1 shows the comparison results. We distinguish 10 different application domains. The most important result we can see from Table 2.1 is that, almost all the trust models correspond to the domain of either Multi-Agent System & P2P community or e-commerce. The definition of trust models for those two domains is not surprising because they cover a large portion of the service provisioning space. Note that one trust model can cover more than one domain if it is properly designed. Some trust models are designed to be application independent, but some are designed for a specific domain such as file sharing (10, 14) and SaaS (37). The recommender model (18, 27, 32) is considered a specific paradigm for modeling trust. Its typical application domains are e-commerce and movie/music recommendations. There are two special domains in this table: travel and living. The former refers to the services related to travel arrangements such as accommodation, transportation, catering, etc. The latter refers to the services related to daily life such as food, drink, shopping, home services, health and medical services, etc. The two domains are special since unlike normal e-commerce application such as online shopping, currently payment of travel or living related service is carried on in reality. After payment and consumption of the service, users provide a rating or a review to the TMS. In addition, the ratings for travel-related and living-related

services are highly subjective. Features such as offline payment and rating subjectivity increase difficulty of trust model design.

2.3.2 Comparison of Trust Models with respect to Information Representation

Before designing a trust model one has to specify what types of information the system is dealing with. One type of input information for the TMS consists of observations (also called experiences, ratings or reviews). A basic approach towards trust evaluation is to aggregate observations by giving different observations different weights depending on the character of observations. For instance, new observations should be given a higher weight than old observations. The choice of computational model is largely dependent on the character of observations. There are at least two questions one has to ask:

a) Is the observation discrete or continuous?

b) Is the measure subjective or objective? In some cases, an observation is objective, for instance if agent A downloads a file which contains virus from agent B, the system will collect a negative observation of B's service provisioning behavior (10). However, in other cases, an observation will be subjective (18, 22, 27, 30, 32), as is typically the case for rating systems.

In addition, the outcome of the system is either a single value or a vector which measures the quantity(ies) of trust. There are two types of trust structure. In a simple trust structure, trust is measured along just one dimension. A classic example is EigenTrust (10). Most trust models follow the simple trust structure. A complex trust structure models trust as a multi-dimensional concept. In a complex trust structure, service quality is decomposed into different dimensions (attributes) in order to refine the definition of trust. A complex trust model doesn't necessarily lead to representing trust as a vector (19, 30), because a single value of trust can be finally calculated by using weighted sum (31, 37) or partial order (29). Table 2.2 shows the comparison results.

2.3.3 Comparison of Trust Models with respect to Properties of Trust Evaluation

In section 2.2 we introduced the important properties of trust evaluation. Here we compare different trust models w.r.t. the criterion of trust evaluation properties; this is shown in Table 2.3. There are several abbreviations and special symbols that we need to define. The symbol "#" after a property name expresses that this property is essential to evaluate trust for service provisioning. For instance, unlike recommender systems (18, 27, 32) which work in the domain of online shopping and movie recommendation, service quality could vary over time. Therefore, the property of dynamics of quality is followed by a "#". Most of the properties are introduced in section 2.2. The properties of attack resistance and machine learning are not mentioned in section

	Multi-Agent Systems & P2P Community	E-Commerce	Information Quality	File Sharing	Web Service	Movie	Travel	Living	SaaS
BizRate		*							
Ebay		*							
Amazon		*							
Epinions		*							
Yelp								*	
TripAdvisor							*		
Advogato	*								
CBFiltering 1997 (33)		*				*			
CFiltering 1999 (27)		*				*			
Yu 2000 (6)	*								
Abdul-Rahman 2000 (38)		*							
Manchala 2000 (7)		*							
Jøsang 2001 (8)	*	*							
Chen 2001 (31)		*	*						
Aberer 2001 (34)	*								
Sarwar 2001 (18)						*			
REGRET 2001 (19)	*	*							
Yu 2002 (20)		*							
Yu 2003 (9)	*								
Richardson 2003 (5)			*						
Wang 2003 (30)	*								
EigenTrust 2003 (10)	*			*					
Buchegger 2004 (21)	*	*							
PeerTrust 2004 (22)	*	*							
Guha 2004 (11)		*							
FuzzyTrust 2005 (28)	*	*							
TrustGuard 2005 (23)		*							
Maximilien 2005 (29)	*				*				
TRAVOS 2006 (24)	*								
Wang 2006 & 2007 (12, 13)	*								
PowerTrust 2007 (14)	*			*					
Quercia 2007 (15)			*						
YZhang 2007 (36)	*								
Resnick 2007 (32)		*							
JZhang 2008 (25)	*	*							
Liu 2008 (16)		*							
Limam 2010 (37)									*
Noorian 2011 (26)	*	*							
Duan 2012a (59)		*							
Duan 2012c (68)							*		

Table 2.1: Trust models comparison with respect to application context

	Binary/**D**iscrete/**C**ontinuous Observation	**O**bjective/**S**ubjective Observation	**S**imple/**C**omplex Trust Structure
BizRate	D	O	S
Ebay	D	O	S
Amazon	D	O	S
Epinions	B	S	S
Yelp	D	S	S
TripAdvisor	D	S	S
Advogato	D	S	S
CBFiltering 1997 (33)	D	S	S
CFiltering 1999 (27)	D	S	S
Yu 2000 (6)	C	O/S	S
Abdul-Rahman 2000 (38)	D	S	S
Manchala 2000 (7)	B	O	S
Jøsang 2001 (8)	B	O/S	S
Chen 2001 (31)	D	S	C
Aberer 2001 (34)	B	O	S
Sarwar 2001 (18)	D	S	S
REGRET 2001 (19)	C	S	C
Yu 2002 (20)	B	O/S	S
Yu 2003 (9)	D	O/S	S
Richardson 2003 (5)	C	S	S
Wang 2003 (30)	B	S	C
EigenTrust 2003 (10)	B	O	S
Buchegger 2004 (21)	B	O	S
PeerTrust 2004 (22)	C	S	S
Guha 2004 (11)	C	O/S	S
FuzzyTrust 2005 (28)	C	O	S
TrustGuard 2005 (23)	C	O/S	S
Maximilien 2005 (29)	B/D/C	O/S	C
TRAVOS 2006 (24)	B	O	S
Wang 2006 & 2007 (12, 13)	B	O/S	S
PowerTrust 2007 (14)	B	O	S
Quercia 2007 (15)	D	S	S
YZhang 2007 (36)	D	O	S
Resnick 2007 (32)	D	S	S
JZhang 2008 (25)	B	S	S
Liu 2008 (16)	B	S	S
Limam 2010 (37)	C	O	C
Noorian 2011 (26)	B	S	S
Duan 2012a (59)	B	S	S
Duan 2012c (68)	D	S	S

Table 2.2: Trust models comparison with respect to formulation

2.2, since they are not strongly related to trust evaluation per se. A trust model with the innate feature of attack resistance is superior to the one without it, because TMSs are vulnerable to attack. Machine learning, focusing on prediction based on known properties learned from the training data, has been developed for purposes other than trust modeling. But there are two interesting models (15, 16) using machine learning technology to predict a trust value. Furthermore, we develop some properties such as personal experience first. Personal experience first means considering both personal (first) experience and recommendation, the former should always be given more weights than the latter. The symbol "*" represents that the trust model in that row has the property corresponding to that column. The symbol "D" represents that whether the model has the corresponding property depends on the parameters specified in the trust model. For instance, whether REGRET (19) has the property of personal experience first depends on the weights given to personal experience and recommendation (second-hand experience).

Table 2.3 shows us the results of trust models comparison w.r.t. properties. Almost all of the models have the property of personal experience first or the property of recommendation. Some trust models have both of them. It is obvious to have either of the two properties or both because trust value should be derived from either first-hand or second-hand information or both. Exceptionally, there are two models (29, 37) which have neither of them. The two models assume existence of a strong system trust (1) which monitors service quality. The system trust refers to "the extent to which one believes that proper impersonal structures are in place to enable one to anticipate a successful future endeavor" (1). Service consumers do not need to provide observations in this case. Instead, system collects the corresponding evidence. Different trust models treat the property of personal experience first differently. On the one hand, from the service selection perspective, the property of personal experience first is practically not very useful since in most cases a user queries the system when he has no information about a certain service or the information is out of date. On the other hand, the property of personal experience first is indispensable for multi-agent cooperation (2). In particular, REGRET (19) has an uncertain state on the property of personal experience first. Since it uses a weighted sum to combine first-hand and second-hand information. When the weight given to first-hand information is larger than 0.5, this model has the property; otherwise, it does not have it.

Furthermore, the property of evaluation confidence is very important for trust evaluation on service provisioning due to data sparsity (4, 18), yet only a few models (19, 24, 26, 27, 31, 32) contain the property. Whether transitivity of trust is applicable depends on features of application domain and semantic constraints (17). Whether the property of preference structure is applicable depends on features of application domain as well. The properties of transaction context and dynamics of quality are the least often considered, furthermore only a few models (7, 19, 22, 23, 28, 37) have both of them.

In particular, many models have drawn attention to the robustness of TMSs since it is such a serious issue (7, 9, 10, 14, 21, 22, 23, 24, 25, 26, 32, 36, 59). All of the models that include robustness intend to mitigate negative influence of attacks by enhancing the robustness of trust models per se. We call it intrinsic robustness enhancement. A

feedback mechanism refers to the mechanism that TMSs compare the difference between two numbers. The first number is a value of trust evaluation for a service. The second one is the numerical rating given by an evaluator after service consumption. It is widely used for recommender systems to evaluate accuracy of prediction or recommendation (18, 27, 32).

2.3.4 Comparison of Trust Models with respect to Robustness of System

Robustness of a TMS is the ability of a TMS to cope with inaccuracy during trust evaluation. There are two main reasons causing evaluation inaccuracy: data sparsity and attack. Data sparsity refers to the state of a TMS where there is not enough data for evaluating trust. If there is no data available on a service, one cannot compute a trust value. Item-based collaborative filtering (18), which regards the trust value of an analogous service as the result, is an exception to this case. However the validity of the item-based collaborative filtering is evaluated only in the movie recommendation domain. The problem of data sparsity usually only occurs at the very beginning of a TMS' life cycle. It is also known as the cold-start problem. The second reason for evaluation inaccuracy is attack on TMSs (51, 52). This refers to any attempt at influencing or controlling the evaluation of trust. Attack can be classified as purposeless attack (system destruction) and purposeful attack (manipulation). The difference between the two types of attack is whether the intention of attack is explicit or not. Almost all the attacks are purposeful (51, 52). There are five main types of manipulation: promoting, slandering, dynamic character, white-washing and orchestration.

a) Promoting: attackers provide fraudulent positive observations or ratings to promote the trust value of a service.

b) Slandering: attackers provide fraudulent negative observations or ratings to demote the trust value of a service.

c) Dynamic personality (22): attackers can build trust and then start cheating (promoting/slandering) or oscillating between building and losing the trust.

d) White-washing (51): attackers abuse the system for short-term gains by letting their trust degrade and then reentering the system with a new identity with fresh trust.

e) Orchestration (51): colluders follow a multi-faceted, coordinated attack. These attacks utilize multiple strategies.

From Table 2.4 we can see that most of the trust models suffer from the cold-start problem, with the exception of Epinions.com and (36, 37). The cold-start problem is solved by importing a revenue based incentive mechanism (4), allowing credit transfer between users in the same social group (36) or importing system trust (1) to guarantee data sufficiency (37).

In addition, regarding the white-washing problem, the trust models in yelp.com and tripadvisor.com are immune to white-washing since they verify identities of service

	Transitivity	Preference Structure #	Personal Experience First	Recomm-endation #	Experience Context #	Dynamics of Quality #	Evaluation Confidence #	Attack Resistance #	Feedback Mechanism #	Machine Learning
BizRate				*						
Ebay				*						
Amazon				*						
Epinions				*				*		
Yelp				*						
TripAdvisor				*						
Advogato	*			*				*		
CBFiltering 1997 (33)				*						
CFiltering 1999 (27)		*		*			*			
Yu 2000 (6)	*									
Abdul-Rahman 2000 (38)		*								
Manchala 2000 (7)		*								
Jøsang 2001 (8)	*	*								
Chen 2001 (31)				*			*			
Aberer 2001 (34)				*						
Sarwar 2001 (18)		*	*							
REGRET 2001 (19)		*	D	*	*	*	*			
Yu 2002 (20)			*							
Yu 2003 (9)	*			*				*		
Richardson 2003 (5)	*			*						
Wang 2003 (30)		*		*					*	
EigenTrust 2003 (10)	*			*				*		
Buchegger 2004 (21)			*	*		*		*	*	
PeerTrust 2004 (22)		*	*	*	*	*		*		
Guha 2004 (11)	*			*						
FuzzyTrust 2005 (28)				*	*	*				
TrustGuard 2005 (23)		*	*	*	*	*		*		
Maximilien 2005 (29)		*								
TRAVOS 2006 (24)			*				*	*		
Wang 2006 & 2007 (12, 13)	*			*						
PowerTrust 2007 (14)	*			*				*		
Quercia 2007 (15)	*			*		*				*
YZhang 2007 (36)	*									
Resnick 2007 (32)		*		*				*	*	
JZhang 2008 (25)		*	*	*		*	*	*		
Liu 2008 (16)	*			*						*
Limam 2010 (37)		*			*	*				
Noorian 2011 (26)		*	*	*			*	*		
Duan 2012a (59)				*	*			*		
Duan 2012c (68)				*	*	*		*		

Table 2.3: Trust models comparison with respect to properties

providers. The recommendation models (18, 27, 33) are immune as well because service providers cannot easily change their identities. Noura Limam's trust model (37) solves the white-washing problem by using system trust (1). Zhang's trust model (36) proposes methods such as CAPTCHA and assigns the lowest trust to new comers to cope with the white-washing problem.

There are two methods that can be employed to restrict the effect of dynamic personality. One is to use a time window or time decay factor (7, 10, 15, 19, 21, 23, 25, 28, 36), because this reduces the time interval in which attackers can play with a TMS. The other is to consider the transaction context (22, 28) such as size of the transaction when evaluating trust. This method prevents one from cheating in a large transaction right after building the trust by succeeding in having made a number of small transactions.

The problem of orchestration is so complicated that only a few models, including the website Advogato, manage to cope with it by assuming the strategies of orchestration (9, 14, 22, 23, 36), using a feedback mechanism (32) or pre-trust (7, 10, 37). By assuming the strategies of orchestration, one can capture the character of orchestration and propose efficient detection algorithms. Feedback mechanism is a good solution against orchestration since we can always compare the real experiences to the contaminative trust value and decide whether a system is under attack or not. Pre-trust assumes the whole set or some subset of observations, which is used for trust evaluation, is free of orchestration. Then we can derive the genuine trust value based on the trustworthy data.

In order to detect and counter purposeful attacks, there are seven main strategies being used by the trust models which are shown in Table 2.5.

a) Statistics. Some trust models filter out fraudulent observations or ratings by assuming that the majority of observations are genuine (6, 9, 20, 36). The basic idea is to override the effect of fraudulent ratings by considering a large number of users who offer honest ratings.

b) Pre-trust. It is defined that some peers (7, 10) or the trust evaluation infrastructure (37) are trustworthy. The robustness of a TMS is built on this base. The website Advogato uses a method similar to EigenTrust (10).

c) Security mechanism. The common security mechanisms used for enhancing robustness are: verification (7), CAPTCHA (36) and encryption/decryption (36). In the domain of e-commerce, verification is used to check a customer's authentication credentials. Verification and CAPTCHA can increase the cost of changing a new identity in a TMS. Moreover, classical security technology like encryption/decryption can be used to prevent attackers from intercepting the data of a TMS in a distributed computing environment.

d) Feedback mechanism. At certain point of time a user A asks a TMS to what degree a service can be trusted. The TMS computes a score T for the request. After using the service, the user A will submit his/her own rating T' for the service. Then the system can compare the difference between T and T' in order to evaluate the

accuracy of the trust evaluation. There are two models (21, 32) using this mechanism to resist manipulation.

e) Manipulative behavior assumption. PeerTrust (22) and TrustGuard (23) follow the assumption that peers in a collusive group give good ratings within the group and bad ratings outside the group. A collusive group refers to a set of users who agree to behave collaboratively in order to make a profit from a TMS. The trust model (34) gives a typical malicious behavior pattern and defines the corresponding metrics based on this pattern. TRAVOS (24) assumes that the trust value of a trustee won't change over time.

f) Personal experience. Trust models (21, 22, 23, 24, 26, 32) evaluate the trustworthiness of the second-hand information and the corresponding referral by comparing personal experience to the second-hand information. It is obvious that you trust the referrals more, if their observations are more similar to yours. However it may not always work since you may have no experience on the service or your experience has already been out of date.

g) Abuse report. Websites such as Amazon.com, Yelp.com and TripAdvisor.com allow users to submit a message to report the suspiciousness of a rating.

2.4 Robustness Enhancement Mechanism

In subsection 2.3.4 we introduced the main techniques of intrinsic robustness enhancement, such as pre-trust, feedback mechanism and personal experience. However, the way of enhancing trust model per se is not enough to fight against manipulation (60, 61). Some external approaches, so called extrinsic robustness enhancement, must be adopted to detect and to filter out manipulative behavior.

During the study, we find 13 relevant works including ours which manage to detect attacks in TMSs. At first we compare the works w.r.t. application domains. None of them are proposed in the P2P or MAS domain (see Table 2.6). Instead, all of them focus on e-commerce, travel-related services or living-related services. It is not a coincidence that most of them focus on e-commerce, since a large number of manipulative behaviors have already been uncovered in this domain (59, 62, 63, 64, 65). In addition to e-commerce websites like Amazon.com and Taobao.com which suffer from manipulation, many review websites in which different types of service are discussed and reviewed, such as TripAdvisor.com, Yelp.com, Dianping.com, etc., are struggling against manipulation as well (66, 67, 68, 69). We can see from Table 2.6 that manipulation on TMSs exists universally online.

Regarding the methodology used for manipulation detection, all of the enhancement approaches are compared in Table 2.7. There are four types of approaches, which are statistical filtering, clustering, classification and semi-unsupervised learning. The first approach is called statistical filtering. This type of approach assumes that unfair ratings can be recognized by their statistical properties. The previous work (70, 71, 72) intends to build such a statistical model to filter out unfair or fraudulent ratings. For

	Promoting	Slandering	White-washing	Dynamic Personality	Orchestration	Cold-start
BizRate	*	*	*	*	*	*
Ebay	*	*	*	*	*	*
Amazon	*	*	*	*	*	*
Epinions	*	*	*	*	*	
Yelp	*	*		*	*	*
TripAdvisor	*	*		*	*	*
Advogato				*		*
CBFiltering 1997 (33)	*	*			*	*
CFiltering 1999 (27)	*	*			*	*
Yu 2000 (6)	*	*	*	*	*	*
Abdul-Rahman 2000 (38)	*	*	*	*	*	*
Manchala 2000 (7)			*			*
Jøsang 2001 (8)	*	*	*	*	*	*
Chen 2001 (31)	*	*	*	*	*	*
Aberer 2001 (34)		*	*	*	*	*
Sarwar 2001 (18)	*	*			*	*
REGRET 2001 (19)	*	*	*		*	*
Yu 2002 (20)	*	*	*	*	*	*
Yu 2003 (9)			*	*		*
Richardson 2003 (5)	*	*	*	*	*	*
Wang 2003 (30)	*	*	*	*	*	*
EigenTrust 2003 (10)			*			*
Buchegger 2004 (21)			*		*	*
PeerTrust 2004 (22)			*			*
Guha 2004 (11)	*	*	*	*	*	*
FuzzyTrust 2005 (28)	*	*	*		*	*
TrustGuard 2005 (23)			*			*
Maximilien 2005 (29)	*	*	*	*	*	*
TRAVOS 2006 (24)	*	*	*	*	*	*
Wang 2006 & 2007 (12, 13)	*	*	*	*	*	*
PowerTrust 2007 (14)			*	*		*
Quercia 2007 (15)	*	*	*		*	*
YZhang 2007 (36)					*	
Resnick 2007 (32)			*			*
JZhang 2008 (25)			*		*	*
Liu 2008 (16)	*	*	*		*	*
Limam 2010 (37)						
Noorian 2011 (26)			*	*	*	*

Table 2.4: Trust models comparison with respect to robustness against attacks

	Statistics	Pre-trust	Security Mechanism	Feedback Mechanism	Manipulative Behavior Assumption	Personal Experience	Abuse Report
BizRate							
Ebay							
Amazon							*
Epinions							
Yelp							*
TripAdvisor							*
Advogato		*					
CBFiltering 1997 (33)							
CFiltering 1999 (27)							
Yu 2000 (6)	*						
Abdul-Rahman 2000 (38)							
Manchala 2000 (7)		*	*				
Jøsang 2001 (8)							
Chen 2001 (31)							
Aberer 2001 (34)					*		
Sarwar 2001 (18)							
REGRET 2001 (19)							
Yu 2002 (20)	*						
Yu 2003 (9)	*						
Richardson 2003 (5)							
Wang 2003 (30)							
EigenTrust 2003 (10)		*					
Buchegger 2004 (21)				*	*		
PeerTrust 2004 (22)					*	*	
Guha 2004 (11)							
FuzzyTrust 2005 (28)							
TrustGuard 2005 (23)					*	*	
Maximilien 2005 (29)							
TRAVOS 2006 (24)					*	*	
Wang 2006 & 2007 (12, 13)							
PowerTrust 2007 (14)					*		
Quercia 2007 (15)							
YZhang 2007 (36)	*		*				
Resnick 2007 (32)				*	*		
JZhang 2008 (25)						*	
Liu 2008 (16)							
Limam 2010 (37)		*					
Noorian 2011 (26)						*	

Table 2.5: Trust models comparison with respect to strategies against manipulation

	Travel	E-Commerce	Living
Dellarocas 2000 (70)		*	
Whitby 2005 (71)		*	
Jindal 2008 (62)		*	
O'Mahony 2010 (75)	*		
Wu 2010 (66)	*		
Lim 2010 (63)		*	
Duan 2011 (72)		*	
Ott 2011 (67)	*		
Mukherjee 2012 (64)		*	
Duan 2012a (59)		*	
Duan 2012b (65)	*		
Duan 2012c (68)	*		
Duan 2013			*

Table 2.6: Robustness enhancement mechanisms comparison with respect to application domain

instance, unfair positive ratings are detected by using clustering the target ratings and they assume that unfair positive ratings exist then the cluster with higher ratings is the unfair group; the unfair negative ratings have been avoided by using controlled anonymity mechanism (70). Alternatively the observed ratings which stay outside of the confidence interval are considered to be unfair (71).

The other three approaches can be generally called machine-learning-based approaches. Theoretically there are four types of machine-learning-based approaches which can be applied for this task: clustering, classification, semi-supervised learning (73) and semi-unsupervised learning (74). From the-state-of-the-art we know that semi-supervised learning has not been applied to the topic of manipulation detection, therefore we list only three types in Table 2.7 except semi-supervised learning. Most of the work (62, 63, 64, 66, 67, 69, 75) use classification, however the quality of human annotations can be a weak point for a classification-based approach (67). We argue that, unless one has a set of high-quality training data, clustering or semi-unsupervised learning is a better choice.

	Statistical Filtering	Clustering	Classification	Semi-unsupervised
Dellarocas 2000	*			
Whitby 2004	*			
Jindal 2008			*	
O'Mahony 2010			*	
Wu 2010			*	
Lim 2010			*	
Duan 2011	*			
Ott 2011			*	
Mukherjee 2012			*	
Duan 2012a (59)		*		
Duan 2012c (68)		*		
Duan 2012c			*	
Duan 2013				*

Table 2.7: Robustness enhancement mechanisms comparison with respect to methodology

3

Trust in Service Provisioning

In this chapter we propose a framework for calculating trust in service provisioning and introduce how to evaluate trust in service.

3.1 A Trust Model Framework for Service Provisioning

In this section a trust framework and its building components are introduced. Next we define metrics for evaluating confidence of trust calculation. Finally accuracy of a TMS and feedback mechanism are discussed.

3.1.1 A Framework for Trust Evaluation

Section 2.3 presented a classification of 40 trust models (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38). But this comparison does not put the models into a consistent framework. Such a framework (or metamodel) would, however, be very useful for systematically analyzing the properties of existing models and chart out the characteristics of new ones. Therefore, we propose a trust model framework for service provisioning to fill in the blank. The essence of the framework is captured by formula 3.1.

$$TM(e,s) = \sum_{i=1}^{R(s)} r_i * \prod_k w_k(eva_i, e, r_i, s) \tag{3.1}$$

TM denotes a trust model, e represents a trust evaluator, the service is denoted by s, the rating is r_i and eva_i denotes a referral who provides the rating r_i. TM, which is defined as a function of a trust evaluator e and the service s, is called a trust model. The outcome of a trust model is the quantity of trust that e has in s. The key idea of trust evaluation is to aggregate ratings (observations or experiences) after normalizing them by weight functions w_k. The design of weight functions can determine whether a trust model is fit for purpose. For instance, if the quality of s changes over time and the quantity of trust is an indicator of the quality, we could design a function which assigns

more weights to the latest ratings than the old ones. In formula 3.1, e represents a trust evaluator who could be either a service consumer in a classic e-commerce setting, or a peer who both provides and consumes service. $R(s)$ represents the set of ratings assigned to service s. A trust evaluator who provides the rating r_i is represented as eva_i. The trust evaluator e could ever provide a rating to the service s. Therefore, eva_i could be the trust evaluator e itself or other evaluators.

When people rate services, facilities etc., they do not do this completely objectively even though they often think they do. Different people have different backgrounds and different standards for judging their environment. In addition, some are generally more critical (or more generous) than others. So given a scale 1 to 10, some will use all the 10 values, others will only use 1 to, say, 7, while yet others will rate everything between 5 and 10. In order to make those different ratings comparable, we suggest normalizing them using the weight functions such as Pearson correlation coefficient and cosine similarity.

The most common similarity measurements are Pearson correlation coefficient and cosine similarity (27). The Pearson correlation coefficient is a measure of the linear dependency between two variables, with a value ranging from -1 to +1. +1 indicates strict linear dependency; 0 indicates complete independence. The instantiation of a weight function based on a correlation coefficient is shown in formula 3.2. Letter a and b represent two trust evaluators. $CRS(a, b)$ represents a set of services which are commonly rated by both a and b.

$$W_{pcc} = \frac{\sum_{i=1}^{|CRS(a,b)|}(r_{ai} - \bar{r_a})(r_{bi} - \bar{r_b})}{\sqrt{\sum_{i=1}^{|CRS(a,b)|}(r_{ai} - \bar{r_a})^2}\sqrt{\sum_{i=1}^{|CRS(a,b)|}(r_{bi} - \bar{r_b})^2}} \tag{3.2}$$

Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between the two vectors. In this case, the set of ratings on $CRS(a, b)$ given by either a or b is considered as a vector. The instantiation of a weight function with respect to cosine similarity is given by formula 3.3.

$$W_{cos} = \frac{\sum_{i=1}^{|CRS(a,b)|} r_{ai} * r_{bi}}{\sqrt{\sum_{i=1}^{|CRS(a,b)|}(r_{ai})^2}\sqrt{\sum_{i=1}^{|CRS(a,b)|}(r_{bi})^2}} \tag{3.3}$$

Regarding a service, dynamics of service quality is the most significant factor to consider for trust evaluation. Dynamics of service quality means that the quality of service could change over time. Different weight functions can be designed to implement the idea that the older a rating, the less important the rating is. The first type of weight function is called time window; it simply ignores all the ratings outside of a specified time window. The width of a time window is usually determined empirically or by assumptions (70). For instance, we can set such a time window that there are

quantitative enough ratings within it. The weight function for a time window is given by formula 3.4, where t_i is a time stamp of a rating i.

$$w_{tw} = \begin{cases} 1 & \text{if } t_i \text{ is within the time window} \\ 0 & \text{Otherwise} \end{cases} \tag{3.4}$$

The second type is called forgetting or discount factor (21). The corresponding weight function for the discount factor is given by formula 3.5, where tw_i represents the time window which a rating i belongs to. The whole time axis is divided into small time windows, which are referenced by an index. The length of a time window is application-dependent. Each rating falls into exactly one time windows. Function $idx()$, whose argument is a time window tw_i, returns an index of a time window. The index of a time window changes dynamically over time: The time window which the current time belongs to, always has index 0. The next time window has index 1, and so on. What the weight function w_{ff} does is to give 1 to the ratings in the most recent time window, and give α to the ratings shown in the next time window, and so on. The weight function w_{ff} gives exponentially less weight to older observations.

$$w_{ff} = \alpha^{idx(tw_i)}, \text{ where } \alpha \in (0, 1] \tag{3.5}$$

Regarding a rating, transaction context is a key concern for creating the weight function. Transaction context refers to the context with respect to a financial transaction in E-commerce. For instance, price is a typical transaction context. It is obvious that a rating for a transaction of 1000 dollars should not be weighed the same as that for a transaction of 10 cents. However, it is difficult to propose an abstract weight function since it is application-dependent.

Manipulation influences trustworthiness of ratings as well. Manipulation suspicion can be considered as an extra factor for creating a weight function. Based on assumptions, the statistical character of the manipulation could be learned by statistical inference or machine learning (59, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 75, 76). After applying a manipulation detection approach, we could label each rating with a real number which indicates to what degree this rating is suspicious in terms of manipulation. For instance, for each rating, a value between 0 and 1 is obtained directly using fuzzy c-means (68). In fuzzy clustering, every point has a degree of belonging to clusters, as in fuzzy logic, rather than belonging completely to just one cluster. In this case, the outcome of a manipulation detection approach can be treated as that of a weight function considering manipulation suspicion.

Fig. 3.1 shows a typical scenario for service provisioning and consumption. All the objects are categorized into two classes: evaluator and services. An edge between two points represents a rating which an evaluator gives to a service. Formally, this can be described as a bipartite graph, where an edge connects a vertex in the evaluator set to one in the service set. Recommender Systems and e-commerce websites can be modelled in this way. Let us take evaluator e_3 as a subject and consider how to build weighting functions in different circumstances.

If e_3 wants to calculate his trust in s_2, first of all he needs to consider first-hand information. The first-hand information refers to the ratings provided by e_3. A dashed

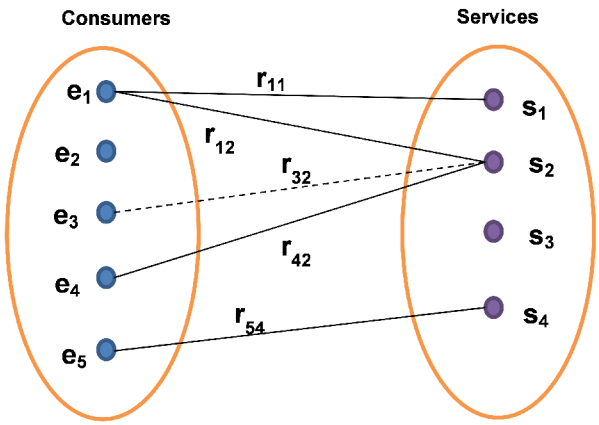

Figure 3.1: Service provisioning and consumption scenario

edge presents an out-of-date rating (first-hand information). The reason that the edge is out-of-date could be that, it is out of a time window. Since e_3 doesn't have any qualified first-hand information, he has to consider only recommendations (second-hand information), which are r_{12} and r_{42}. Different weight functions in terms of forgetting factor, transaction context, manipulation suspicion, can be used to normalize the ratings.

If e_3 wants to evaluate the trust of s_1, he can only consider recommendations since he never used the service before. Note that the $CRS(e_1, e_3)$ is equal to s_2. Formulae 3.2 and 3.3 are qualified to implement similarity-based weight function. Moreover, using formula 3.1, e_3 can't evaluate the trust on s_3, since s_3 has no rating at all. However, item-based collaborative filtering (18) aims to solve the problem by measuring the similarity between services. If s_2 and s_3 are similar, then the ratings of s_2 can be used for evaluating the trust of s_3. Item-based collaborative filtering (18) is regarded as an exceptional case which our trust model framework does not cover.

Regarding a referral, transitivity, which is defined in section 2.2, can be used to build up a weight function with respect to trustworthiness of the referral. Note that transitivity doesn't fit into the representation shown in Fig. 3.1, since there is no trust between evaluators. In this part, we are not considering the trust between evaluators and services, but the trust among evaluators. MAS and P2P community are the typical scenarios for considering transitivity (2, 4, 5, 10, 14), because in these communities a trust evaluator provide service to other trust evaluators. Therefore the trust between evaluators can be calculated. Under certain semantic constraints (17), trust among evaluators is transitive. If trust is transitive, a weighted directed graph, so called trust network, can be generated. In this graph, a vertex represents a trust evaluator. If an evaluator a trusts an evaluator b, then there is an edge from vertex a to vertex b labeled with a value indicating to what degree a trusts b. There are two ways representing the labeled degree: subjective logic (8) and normal transitive closure (5, 10). Regarding trust inference in a trust network, operations of concatenation (5) and aggregation (5) must be instantiated. In Fig. 3.2, a subset of a trust network which is given in

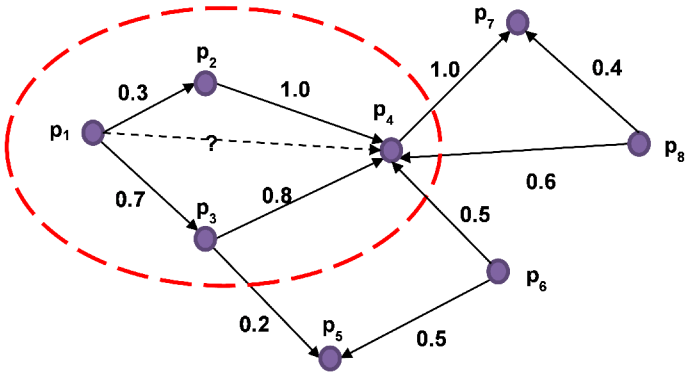

Figure 3.2: Example of a trust network

the dashed circle shows an application of the two operations. In order to infer to what degree p_1 trusts p_4, we need to combine the trust values among evaluators by performing two steps. First, we need to follow each path from p_1 to p_4, concatenating the weights associated with edges along the way. In Fig. 3.2 there are two paths: $p_1p_2p_4$ and $p_1p_3p_4$. The second step is to aggregate the concatenation results from the two different paths. Both concatenation and aggregation can be implemented using subjective logic (8) or normal transitive closure (5). In subjective logic, trust is represented as a triple $< b, d, u >$ where the elements of the triple measure belief, disbelief and uncertainty respectively. The corresponding concatenation operator is implemented by formula 3.6, where p_1 and p_2 represent two trust evaluators; p_4 represents now the service. Formula 3.6 calculates the triple $< b_{p_4}^{p_1p_2}, d_{p_4}^{p_1p_2}, u_{p_4}^{p_1p_2} >$ with respect to the path $p_1p_2p_4$. Following the same idea, the triple $< b_{p_4}^{p_1p_3}, d_{p_4}^{p_1p_3}, u_{p_4}^{p_1p_3} >$ with respect to another path can be calculated. The aggregation operator is implemented by formula 3.7, where P_1 and P_2 represent the two paths and $P_1 \bigoplus P_2$ represents the aggregation of the two paths.

$$\begin{aligned} b_{p_4}^{p_1p_2} &= b_{p_2}^{p_1} b_{p_4}^{p_2} \\ d_{p_4}^{p_1p_2} &= b_{p_2}^{p_1} d_{p_4}^{p_2} \\ u_{p_4}^{p_1p_2} &= d_{p_2}^{p_1} + u_{p_2}^{p_1} + b_{p_2}^{p_1} u_{p_4}^{p_2} \end{aligned} \tag{3.6}$$

$$
\begin{aligned}
\kappa &= u_{p_4}^{p_1p_2} + u_{p_4}^{p_1p_3} - u_{p_4}^{p_1p_2} u_{p_4}^{p_1p_3} \\
P_1 &= p_1 \rightarrow p_2 \rightarrow p_4 \\
P_2 &= p_1 \rightarrow p_3 \rightarrow p_4 \\
b_{p_4}^{P_1 \bigoplus P_2} &= (b_{p_4}^{p_1p_2} u_{p_4}^{p_1p_3} + b_{p_4}^{p_1p_3} u_{p_4}^{p_1p_2}) \,/\, \kappa \\
d_{p_4}^{P_1 \bigoplus P_2} &= (d_{p_4}^{p_1p_2} u_{p_4}^{p_1p_3} + d_{p_4}^{p_1p_3} u_{p_4}^{p_1p_2}) \,/\, \kappa \\
u_{p_4}^{P_1 \bigoplus P_2} &= (u_{p_4}^{p_1p_2} u_{p_4}^{p_1p_3}) \,/\, \kappa
\end{aligned}
\tag{3.7}
$$

When normal transitive closure is applied, multiplication and minimum value are the valid candidates of the concatenation operator (5). Addition and maximum value are the corresponding candidates of the aggregation operator (5). In particular, Eigentrust (10) uses multiplication as the concatenation operator, and addition as the aggregation operator. In Fig. 3.2, the trust p_1 places in p_4 is equal to $0.3*1.0+0.7*0.8 = 0.86$. The choice of implementation of the weight function results from the problem we are dealing with. In order to decrease the number of downloads of inauthentic files in P2P file-sharing network, it is a good design for EigenTrust to use multiplication and addition to implement the weighting function. However, it could be a bad idea to apply the same design in other problem domains. The rationality and correctness can be evaluated only in a specific problem domain.

The last point to consider regarding building a weight function is the type of ratings. There are two types of them in a TMS: first-hand ratings and second-hand ratings. A first-hand rating refers to the ratings provided by the evaluator himself. A second-hand rating refers to the ratings provided by the other service consumers rather than the evaluator. One popular idea is that, first-hand ratings are superior to second-hand ones (21, 22, 23, 24, 25, 26). Some trust models only consider first-hand ratings (36). A more flexible method is to give different weights to different types of ratings depending on the particular circumstance (19).

3.1.2 Confidence Evaluation

Researchers usually focus on the design of trust evaluation however they ignore the issue of confidence evaluation. Confidence evaluation refers to the metrics to measure the quality of trust evaluation. Using confidence evaluation, we can ask the questions like how confident a trust evaluation is or whether the amount of ratings is large enough to calculate the trust value. Although there are some models (19, 24, 25, 26, 27, 31) considering the issue of confidence evaluation, they don't treat the issue in a systematic way. We consider this issue at two different levels: the system level and the query level. The system level concerns the confidence evaluation from a TMS perspective, while the query level focuses on how much confidence an evaluator has regarding trust evaluation of a certain service. The basic idea of integrating confidence metrics into the framework is to provide more information for end users who want to select a service. At each level, we propose confidence metrics for different evaluation components such as trust inference by transitivity, rating quantity, etc.

a) Confidence for transitivity in the query level (CTQ). At the query level, the metric intends to calculate the ratio of connected referrals to all the referrals provided to a rating of a service. A connection from the evaluator e to a referral means there is an arrow from e to the referral in a trust network. For instance, in Fig. 3.2, if we take p_1 as a trust evaluator and p_4 as a service provider, the trust evaluator has two connected referrals $\{p_2, p_3\}$. $CTQ(e, s)$ is defined by formula 3.8, where R(s) represents the set of referrals who ever provided a rating or ratings to a service s. $ConRef(e, R(s))$ picks up the set of referrals to whom the evaluator e connects, the set of all the referrals is equal to $R(s)$. In Fig. 3.2, p_4 has 4 referrals which are $\{p_2, p_3, p_6, p_8\}$ and $|ConRef(p_1, R(p_4))| = 2$, so $CTQ(p_1, p_4) = 0.5$.

$$CTQ(e, s) = \frac{|ConRef(e, R(s))|}{|R(s)|} \tag{3.8}$$

b) Confidence for transitivity at the system level (CTS). We can convert a directed graph such as the one illustrated in Fig 3.2, into a matrix representation. We create a connection matrix C_{nn}, where n is the number of peers (users) involved. An entry e_{ij} of the matrix is equal to 1 if there is at least one path from peer i to peer j, otherwise it is equal to 0. CTS is defined by the ratio of number of connected pairs to the number of all the possible connections. CTS indicates how many percentage of pairs of users are connected.

$$CTS = \frac{\sum_{i \neq j} e_{ij}}{n(n-1)} \tag{3.9}$$

c) Confidence for rating quantity at the query level (CRQ). When a rating is a binary value, there are two methods for evaluating confidence: certainty measurement (13) and Chernoff bound (77). The certainty measurement considers trust as a certainty in terms of evidence based on a statistical measure defined over a probability distribution of the probability of positive outcomes. Trust is represented as a triple of belief, disbelief and uncertainty. Certainty, which is $1 - uncertainty$, is defined by formula 3.10, where x represents the probability of a positive outcome. p and n represent the number of positive and negative ratings respectively regarding the evaluator e. The Chernoff bound, shown in formulae 3.11 and 3.12, calculates a lower bound for a sequence of N_{min} independent Bernoulli trials. ϵ is the maximum level of error (e.g., 0.05) and γ is a confidence measure on the portion of success . When the number of ratings N_e^s is larger than the lower bound N_{min}, the confidence is 1, otherwise it is 0. When a rating is a discrete or continuous, some heuristics can be applied such as a piecewise function (31) or a sine function (19). The basic idea of the two methods is to create a monotonically increasing function on number of ratings N_e^s that returns a value in the range 0 to 1 inclusive.

$$CRQ(e, s) = c(p, n) = \frac{1}{2} \int_0^1 \left| \frac{x^p (1-x)^n}{\int_0^1 x^p (1-x)^n dx} - 1 \right| dx \tag{3.10}$$

$$N_{min} = -\frac{1}{2\epsilon^2} ln \frac{1-\gamma}{2} \tag{3.11}$$

$$CRQ(e,s) = \begin{cases} 1 & \text{if } N_e^s \geq N_{min} \\ 0 & \text{Otherwise} \end{cases} \tag{3.12}$$

d) Confidence for rating quantity at the system level (CRS). The confidence metric at the system level is derived from the metric CRQ from the query level. One possible implementation is given by formula 3.13, where E is the evaluator set and S is the service set in a TMS.

$$CRS = \frac{\sum_{i=1}^{E} \sum_{j=1}^{S} CRQ(e_i, s_j)}{|E||S|} \tag{3.13}$$

e) Confidence for referral similarity at the query level (CSQ). The basic idea is to create a monotonically increasing function based on the quantity of similar referrals that returns a value in the range [0, 1]. We provide a simple implementation by formula 3.14, where N_e^s represents the number of similar referrals to the evaluator e. All of the referrals have at least one rating of a service s. $N_{threshold}$ is learned empirically and it will vary depending on the concrete situation. For instance, based on the experience of a TMS designer, maybe 3 similar referrals are enough to evaluate the trust of a hotel. More complicated heuristic approaches have been proposed such as a piecewise function (31) and a sine function (19).

$$CSQ(e,s) = \begin{cases} \frac{N_e^s}{N_{threshold}} & \text{if } N_e^s \leq N_{threshold} \\ 1 & \text{otherwise} \end{cases} \tag{3.14}$$

f) Confidence for referral similarity at the system level (CSS). We define matrix CR_{nn}, where n is the cardinality of the consumer population. Entry e_{ij} is equal to 1 if there is at least one service which is rated the same by evaluators i and j, otherwise it is equal to 0. Obviously, the entries on the main diagonal are all equal to 1. CSS is defined by formula 3.15.

$$CSS = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} e_{ij}}{n^2} \tag{3.15}$$

A confidence metric on trust evaluation can be built up from the metrics defined above. The definition of confidence metrics for trust evaluation at both the system and the query level are given in formulae 3.16 and 3.17, respectively.

$$ConfTrust\_sys = CTS * CRS * CSS \tag{3.16}$$

$$ConfTrust\_query(e,s) = CTQ(e,s) * CRQ(e,s) * CSQ(e,s) \tag{3.17}$$

3.1.3 Accuracy of a TMS and Feedback Mechanism

Regarding trust in service provisioning, accuracy of a TMS is the degree of closeness of trust evaluation of service quality to the evaluator's personal perception of the service quality. The accuracy of a TMS is a critical issue since as an Information System (IS), a TMS becomes rather useless when the system delivers inaccurate results. There are two reasons influencing the accuracy of a TMS. The first reason is the design of a trust model. Inadequate design of a trust model decreases the accuracy of a TMS. For instance, giving the same weight to all the ratings for a hotel is a bad idea since that is based on the incorrect assumption that every evaluator (traveler) shares the same standard for rating a hotel service. The second reason is the robustness of a TMS, which refers to the ability of a TMS to cope with manipulation when a TMS is in use. Usually the trust model of a TMS is sensitive to the ratings. If the ratings are given wrongly, the trust model will deliver quite different result. Manipulation refers to the actions of injecting fraudulent ratings to influence trust evaluation.

The key concept for dealing with inaccurate trust evaluation and for fighting manipulation attempts is feedback. Feedback is a mechanism for adjusting previous ratings and evaluations by including information regarding their validity. In this subsection we will argue that a TMS is accurate in the long run given a well-designed feedback mechanism.

Definition 3.1.1. *A rating of a service s_j provided by an evaluator e_i is represented by r_{ij}. $r_{ij}^{(t)}$ stands for a rating given in a time interval $(t-1, t]$. The scale of a rating could be binary, nominal-with-order or continuous. A binary rating represents trust or distrust. A nominal-with-order scale is the most frequently used type. For instance, in eBay and Taobao a user can provide a negative, neutral or positive rating; whereas Amazon and TripAdivsor use a 5-star scheme. Continuous scale is not used very often, but it is possible to design the scale in this manner. For instance, we can regard the observed bandwidth of an online file storage service (OFSS) as a continuous-scaled rating.*

Definition 3.1.2. *$\hat{r}_{ij}$ represents a reference rating which is used to calculate the accuracy of a TMS. A reference point is chosen from all the ratings in a TMS. $\hat{r}_{ij}^{(t)}$ stands for a reference rating given in a time interval $(t-1, t]$.*

Definition 3.1.3. *Evaluator group, E, represents the set of evaluators in TMS. $E(t)$ represents the set of evaluators in a time interval $(t-1, t]$.*

Definition 3.1.4. *Service set, S_i, represents the set of all the services rated by evaluator e_i. REF_j represents the set of referrals who provide ratings to s_j. When considering a rating as an ingredient of trust evaluation, the consumer who provided the rating is called a referral ref.*

Definition 3.1.5. *A weight given to a referral ref_k with respect to the evaluator e_i, w_{ik}, represents to what degree that e_i should trust a recommendation of the referral ref_k. Note that different referrals correspond to different weights even considering the same service. $w_{ik}^{(t)}$ stands for a weight updated in a time interval $(t-1, t]$.*

Definition 3.1.6. *The mismatch of a TMS at time t, $MMATCH(t)$, is defined in formula 3.18. The metrics of mismatch represents the difference between reference ratings and trust evaluation with respect to the evaluator group E at time t. For a TMS without feedback mechanism, $w_{ik}^{(t)}$ is always equal to 1.*

$$MMATCH(t) = \sum_{i=1}^{|E(t)|} \sum_{j=1}^{|S_i|} \sum_{k=1}^{|REF_j|} |\hat{r}_{ij}^{(t)} - w_{ik}^{(t-1)} * r_{kj}^{(t-1)}| \tag{3.18}$$

Lemma 3.1.1. *If all the ratings never change over time and the reference ratings are genuine, then the mismatch of any TMS never increases over time given the feedback mechanism described by Algorithm 1, where α represents a small real number which control when a weight update procedure should be ended.*

Algorithm 1: Weights update procedure

initialization: $\forall i \in E(0), j \in E(0), i \neq j$ set $w_{ij}^{(0)} = 1$;
while $|\hat{r}_{ij}^{(t)} - w_{ik}^{(t-1)} * r_{kj}^{(t-1)}| > \alpha$ **do**
  if $\hat{r}_{ij}^{(t)} > w_{ik}^{(t-1)} * r_{kj}^{(t-1)}$ **then**
    update $w_{ik}^{(t)} = w_{ik}^{(t-1)} + \frac{|\hat{r}_{ij}^{(t)} - r_{kj}^{(t-1)}|}{\sum_{a=1}^{REF(j)} |\hat{r}_{ij}^{(t)} - r_{aj}^{(t-1)}|}$;
    if $w_{ik}^{(t)} > 1$ **then**
      $w_{ik}^{(t)} = 1$;
    end
  end
  if $\hat{r}_{ij}^{(t)} < w_{ik}^{(t-1)} * r_{kj}^{(t-1)}$ **then**
    update $w_{ik}^{(t)} = w_{ik}^{(t-1)} - \frac{|\hat{r}_{ij}^{(t)} - r_{kj}^{(t-1)}|}{\sum_{a=1}^{REF(j)} |\hat{r}_{ij}^{(t)} - r_{aj}^{(t-1)}|}$;
    if $w_{ik}^{(t)} < 0$ **then**
      $w_{ik}^{(t)} = 0$;
    end
  end
end

Proof. Given $\forall t, t > 1, \hat{r}_{ij}^{(t-1)} = \hat{r}_{ij}^{(t)}$ and $\forall t, t > 1, r_{kj}^{(t-1)} = r_{kj}^{(t)}$, when $\hat{r}_{ij}^{(t)} > w_{ik}^{(t-1)} * r_{kj}^{(t-1)}$, and fixing i, j and k, according to the update rule in algorithm 1, $|\hat{r}_{ij}^{(t)} - w_{ik}^{(t-1)} *$

$r_{kj}^{(t-1)}| \geq |\hat{r}_{ij}^{(t+1)} - w_{ik}^{(t)} * r_{kj}^{(t)}|$. Similarly, when $\hat{r}_{ij}^{(t)} < w_{ik}^{(t-1)} * r_{kj}^{(t-1)}$, $|\hat{r}_{ij}^{(t)} - w_{ik}^{(t-1)} * r_{kj}^{(t-1)}| \geq |\hat{r}_{ij}^{(t+1)} - w_{ik}^{(t)} * r_{kj}^{(t)}|$. Therefore $\forall t, t > 0, MMATCH(t+1) \leq MMATCH(t)$ □

Theoretically the mismatch never increases, however, in the practice, we can argue that the mismatch not only non-increase but also decrease over time. Because it is nearly impossible to keep the equation 3.19 being true. If we want to keep the equation 3.19 being true, we need to make sure that the equation 3.20 is true for all the combination of i, j and k and it is very difficult to achieve in practice. Because for a certain k, $\hat{r}_{ij}^{(t)}$ does not equal to $r_{kj}^{(t-1)}$.

$$MMATCH(t) = MMATCH(t+1) \tag{3.19}$$

$$\hat{r}_{ij}^{(t)} - w_{ik}^{(t-1)} * r_{kj}^{(t-1)} = \hat{r}_{ij}^{(t+1)} - w_{ik}^{(t)} * r_{kj}^{(t)} \tag{3.20}$$

Lemma 3.1.2. *If all the ratings never change over time, there are a number of fraudulent evaluators, who provide fraudulent reference ratings. The influence on a TMS is bounded by the feedback mechanism described by algorithm 1.*

Proof. According to the update rules of $w_{ij}^{(t)}$ described in algorithm 1, $\hat{r}_{ij}^{(t)}$ influences only the trust evaluation regarding the evaluator e_i. Hence, the $w_{ij}^{(t)}$ which is effected by $\hat{r}_{ij}^{(t)}$ will influence the trust evaluation regarding the fraudulent evaluator e_i. Over time, these fraudulent ratings will influence the trust evaluation of other genuine evaluators, however according to Lemma 3.1.1, the mismatch never increase. Furthermore, in practice, the mismatch will decrease. □

3.2 Trust Building in Service Provisioning Applications

A service, which is the object of trust, refers to the non-material equivalent of a good, and it is materialized at the moment of service delivery. A TMS is a kind of Information System where all the information can be collected, processed and propagated in an electronic manner. We mainly focus on services whose ratings or reviews are available in a computing environment such as a P2P environment or the Internet. For instance, there are review websites about services such as hotels, restaurants, doctors, office cleaning, lawyers, etc. When considering IT-level services, such as Online File Storage Services (OFSSs), ISPs, e-mail and Web service based applications, ratings and reviews could be substituted with direct observation of service quality. For instance, an end user can observe a successful file transfer or Web service delivery. It is effortless to record these observations programmatically. In the perspective of a TMS, observation, review, rating and experience are interchangeable. Note that in the IT-level service provisioning, it is much easier to collect data for a TMS than in the other domains such as hotel or restaurant reviews.

Moreover, it is complicated to evaluate trust in a service. This difficulty is because service quality is unpredictable. For example, the quality of a hotel may improve or degrade over time. This variation in quality could be due to a variety of reasons such as substitution of the board, lack of staff training, salary increase, and so on.

Regarding service quality, there are many aspects which a user can perceive. In further a service can be specified in a hierarchical way (78) in order to evaluate service quality. Therefore a service can be refined as a set of attributes in terms of trust evaluation.

Definition 3.2.1. *A trust model for an attribute a_i of a service, tm_i, is an instantiation of the trust model framework specified by formula 3.1 restricted to an attribute a_i. $tm_i(e, s)$ returns the trust value of a service s with respect to an evaluator e and attribute a_i.*

Definition 3.2.2. *A trust evaluation regarding the service quality, $< ATT, TM >$, is a pair of sets. ATT represents a set of refined attributes of a service. TM represents a set of trust models (tm_i) with respect to those attributes. The design of a trust model tm_i regarding attribute a_i depends on a variety of criteria which were introduced in section 3.1.*

An evaluator, or a service consumer, which is the subject of trust, refers to a previous, current or potential user of service. There are two ways of representing the trust of an evaluator (subject) in a service. Either the trust is represented as a vector of trust in the attributes of a service, or as a single value which is calculated by aggregating the trust values restricted to different attributes using a preference structure and value functions.

A preference structure (PS), defined as a relation on an n-dimensional service evaluation space, establishes an order among the points in this space. The definition of the order mainly depends on the given comparison rule, e.g., lexicographic ordering, Pareto efficiency ordering or utility function. Utility function (79) refers to a measure of the relative satisfaction from, or desirability of, consumption of goods or services. Here we only discuss the utility function, since it is a complete, transitive, reflexive relation. The universal form of a utility function is represented by formula 3.21. w_i stands for a weight given to a value function $VF_i()$ with respect to attribute a_i. A value function (80) refers to a mapping from the domain of trust evaluation to a satisfaction space. Whether a consumer satisfies with the trust with respect to attribute a_i is determined by an imperative statement, which refers to the preference that cannot be violated. Considering, for example, online file storage service (OFSS) with a service consumer claims that price must less than or equal to 10 $/Month. For attributes with a data type that is discrete and countable but has no order, e.g., a European customer satisfies only with that the server is located in EU, the value function is defined as formula 3.22, where 1 and 0 stand for total satisfaction and total dissatisfaction respectively. It is more complicated to deal with the attributes with data types having a total order, e.g., rating levels, time, price and probability. Because in these cases, the value function

returns not only 0 and 1, but maybe a real number between 0 and 1. A satisfaction function (81) whose inputs are perceived quality and expectation of quality with respect to the end user, is applied to deal in this case.

$$TV_{es} = U(e,s) = \sum_{i=1}^{|A_s|} w_i * VF_i(tm_i(e,s)) \tag{3.21}$$

$$VF(x) = \begin{cases} 1 & \text{if } x \in \text{ valid part} \\ 0 & \text{otherwise} \end{cases} \tag{3.22}$$

It could be argued that the utility function and value function are sometimes not known, since the preference information of a service consumer is difficult to capture. Instead of aggregating trust values regarding attributes into one value, another possibility is to consider a trust vector. For instance, the trust in service provisioning could be presented by a spider web diagram. In the spider diagram in Fig. 3.3, the blue circle represents the trust vector regarding service $s1$; the red one represents the one for service $s2$. Both services are defined in the same structure, which means that both carry the same 5 attributes. The evaluator can then make a decision about which service to choose by comparing those diagrams.

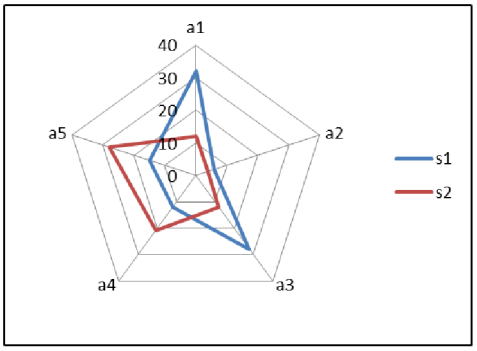

Figure 3.3: Trust representation in a spider web diagram

Considering different attributes, an ideal TMS should contain a collection of trust models which are instances of the trust model framework introduced in section 3.1. Each trust model corresponds to an attribute of a service. A TMS provides multiple trust evaluation results in terms of different criteria. In addition, with the evolution of a TMS, the significance of an evaluation component might change. Consider an extreme example: If an evaluator has some new personal experience regarding a service, then all the weighting functions defined in subsection 3.1.1 seems to be of no importance except for the one regarding transaction context. An ideal trust model provides different trust evaluation results with corresponding parameters and confidence scores. The choice of which result is more valuable is made by the system user.

We give an example to illustrate the idea. Assume that there are two services: s_1 and s_2. ATT is defined as $\{a_1, a_2, a_3, a_4, a_5\}$. The corresponding trust model set TM is defined as $\{tm_1, tm_2, tm_3, tm_4, tm_5\}$. The criterion set is defined as $\{c_1, c_2, c_3, c_4\}$. A

	tm_1	tm_2	tm_3	tm_4	tm_5
a_1	*				
a_2		*			
a_3			*		
a_4				*	
a_5					*

Table 3.1: Relation between attributes and trust models

	c_1	c_2	c_3	c_4
tm_1	*	*		
tm_2		*	*	
tm_3			*	
tm_4	*	*		*
tm_5		*		

Table 3.2: Relation between criteria and trust models

criterion could be any one introduced in section 3.1 combined with a specific parameter set. For instance, a time window w_{tw} combined with a window width (e.g., one month) is a valid criterion here. Note that two criteria having the same core component but a different parameter set are considered to be non-identical. For instance, a time window with a width of one month and a time window of one year are different criteria. Tables 3.1 and 3.2 describe the relations among attributes, trust models and criteria. Fig 3.3 shows the vectors of trust values regarding s_1 and s_2. Fig 3.4 shows the corresponding confidence scores. Again, our trust framework intends to present different types of information to end users, and a final decision is made by the end users themselves.

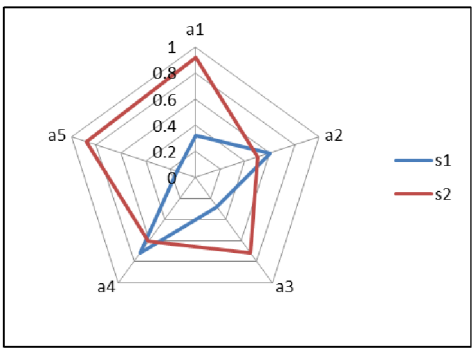

Figure 3.4: Confidence metric on trust evaluation at the query level

4

Online File Storage Service (OFSS)

An Online File Storage Service (OFSS) is an Internet service for hosting user files. Users can backup personal files via the service, they can share photos, audio and video files. An OFSS allows users to upload files so they can be accessed from any computer connected to the Internet. Typically, an OFSS allows access via the HTTP-protocol.

An OFSS is mainly used in two ways, the first case being personal file storage. One can upload files via the Internet and access them again from the same or a different location. When a local system suffers from data loss, users can recover the system from a backup stored on the OFSS. In this case, file upload and download speed (bandwidth) are the most important attributes to consider, since users do not want to spend long time to update the image of a local system. Data security is an issue, too, because the files may contain some confidential information.

The second key application is file sharing. As a tool for facilitating information exchange and sharing among different users, an OFSS provides end users with mechanisms for sharing files such as images, documents, music, software, movies, etc. In the file sharing case, the success of file uploading and downloading might be the most important features of concern. Users would, for example, find it frustrating when a movie download fails in between.

Common OFSSs include Amazon S31[1], Box.net[2], Dropbox[3], Mozy[4], SkyDrive[5], Google Drive[6], etc. There is a Wikipedia page[7] that specifies several attributes for OFSSs from a provider's perspective such as storage size, maximal file size, developer API support, etc. The attributes specified in the Wikipedia page are not related to trust

[1] http://aws.amazon.com/s3
[2] https://www.box.com
[3] https://www.dropbox.com
[4] http://mozy.com
[5] https://skydrive.live.com
[6] https://drive.google.com
[7] http://en.wikipedia.org/wiki/File_hosting_service

building since they are specified by a service provider. This has to be distinguished from attributes such as upload failure rate and file transfer bandwidth, which are defined from a client's perspective. The observed value for attributes such as upload failure rate and file transfer bandwidth can vary over time. Trust models are applicable to a service for which attributes are client-oriented and variable.

4.1 Trust Models for OFSS

In this section we introduce the attributes that are relevant for defining trust models for OFSSs. Then we propose trust models for each of the attributes.

4.1.1 Trust Evaluation and Attributes of an OFSS

There are two types of attributes for an OFSS. The first group comprises those stipulated by service providers; they typically have fixed values. They include price, maximum size of file upload, file transfer security, maximum amount of storage space, etc. The value is usually stipulated by the service providers. These types of attribute are of marginal interest from a trust management perspective. The question of whether the service provider actually supports, e.g., the maximum amount of storage space advertised is easy to check. The questions of how trustworthy a provider is regarding the complete, correct and timely delivery of a file is much harder to answer.

So the second group subsumes those that are perceived differently from client to client such as failure rate and bandwidth. Failure rate is defined as the probability with which a file transfer (either upload or download) is not completed. Bandwidth measures the number of bits transferred per time unit between the client and the service provider (or vice versa). The failure rate and bandwidth perceived from an end user in a village in China can be quite different from one in a big city in USA. There are many factors influencing the end user experience, therefore traditional failure and performance prediction models are not adequate for handling this problem. Comprehensive performance and failure models are impossible to design due to the complicated and dynamically-changing configuration parameters of the many instances involved in a file transfer. Instead of viewing the problem of quality prediction from a service provider's perspective, a trust model treats performance and failure modeling from the viewpoint of the client. This is a different paradigm where the boundary between a user and a provider is shifted towards the user's end. Since we do not know (all) the technical parameters on the way from the client to the server, we effectively have to perform a black-box test from the client's end.

It is plausible to state that, the larger a file, the higher the average failure rate, because the probability of a failure increases over time, and transfer time clearly is positively correlated with file size. We performed an experiment to justify the hypothesis. We downloaded three files with different sizes from Amazon S3 for 40 days. The download frequency is uniformly distributed. The small file is a 7MB audio file; the medium file is a 34MB software setup file; the large file is a 1.2GB movie. Since normally the failure rate is too small to plot, we show the counterpart, which is success rate. The value of success rate is equal to 1 minus the value of failure rate. Fig. 4.1 shows the

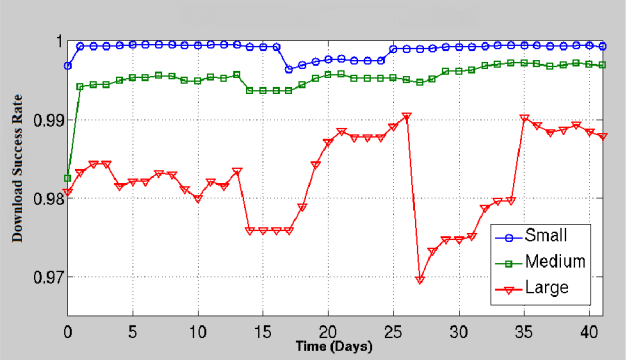

Figure 4.1: Success rates comparison for three types of file

comparison success rates for the three types of file with respect to time. It is shown that the larger a file, the lower the average success rate[1]. The experiment also indicates that, a similar correspondence does not exist between size of a file and the bandwidth of file transfer. As a matter of fact, on some parts of the way between the client and the server load balancing strategies may be in effect that decrease the share of a session in a shared channel the longer it exists, i.e. the more data it tries to transfer. Therefore, we only refine the attribute of failure rate into three sub-attributes: failure rate with respect to small file size, failure rate with respect to medium file size and failure rate with respect to large file size.

Furthermore, we refine the attribute bandwidth into two sub-attributes, one for upload and one for download, respectively. The reason for the refinement is that, there is no experimental result showing that bandwidth for upload is correlated with bandwidth for download. Therefore we need to consider both. There are many factors influencing bandwidth, e.g. a technical bandwidth limitation from providers and ISPs, peering policy, end user connection bandwidth, etc.

In summary, we identify eight attributes of an OFSS regarding trust evaluation: upload failure rate for small file (UFRS), upload failure rate for medium file (UFRM), upload failure rate for large file (UFRL), download failure rate for small file (DFRS), download failure rate for medium file (DFRM), download failure rate for large file (DFRL), upload bandwidth (UB) and download bandwidth (DB).

4.1.2 Trust Models for Failure Rates

Regarding the attribute failure rate, a parameterized beta reputation system can be applied to model the trust in a binary event. A beta reputation system (82) is based on the beta probability density function which is used to represent the probability distribution of binary events. A posteriori probabilities of binary events can be represented as a beta distribution. The beta distribution is a family of continuous probability distribution defined on the interval $[0, 1]$ characterized by two parameters α and β. p

[1]http://en.wikipedia.org/wiki/File_hosting_service

represents the probability of the occurrence of a binary event. The beta distribution $f(p|\alpha, \beta)$ can be expressed using the gamma function Γ as:

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{4.1}$$

where $0 \leq p \leq 1, \alpha > 0, \beta > 0$. The expected value of the beta distribution is given by formula 4.2.

$$E(p) = \frac{\alpha}{\alpha + \beta} \tag{4.2}$$

An observation is represented as either of the two possible outcomes {SUCCESS, FAIL}. Let *pos* be the observed number of outcome SUCCESS, and let *neg* be the observed number of outcome FAIL. The relation between *neg* and α, and that between *pos* and β are listed in formula 4.3.

$$\alpha = neg + 1 \text{ and } \beta = pos + 1 \text{ , where } pos, neg \leq 0 \tag{4.3}$$

Whether a specific file transfer fails or not is a binary event. From Fig. 4.1 we can see, the failure rate of an OFSS is supposed to be very low (less than 0.03). Because of the black-box properties, it is hard to distinguish failure events from the client's end. A failure event occurring during service provisioning is not equivalent to the observations of an end user. For instance, a server is down for a whole day, during the day user A tries to upload a file 100 times and all fail, which doesn't mean the service renders 100 failure events but one. Otherwise it is unfair to assert that A observes failure 10 times as much as B, if user B only uploads file 10 times on that day and all fail. Therefore, we assume that a failure event is a rare event, so the adjacent negative observations in a short interval (e.g. 24 hours) correspond to only one failure event. After the adjustment, both A and B observe only one failure. A failure event is well defined due to the adjustment that the adjacent negative observations in a short interval are considered as one negative observation.

A beta distribution is an ideally suited for normalizing a trust model because it assumes that n experiments are independent and the mean is fixed. Each experiment is independent because the outcomes of n file transfers are independent among each other. The assumption that the mean of the beta distribution is fixed does not really translate into our application because the failure rate of an OFSS is not constant as can be seen from Fig. 4.1. The dynamics of service quality (e.g. failure rate) is one of the most important features of service provisioning. Though a beta distribution belongs to the family of Bayesian probability, combining a weight function regarding time factor with the beta distribution is a better solution than a pure beta system. A trust model is useless when it cannot capture the dynamics of service quality rapidly.

When calculating the trust value for a service, it is reasonable to assume that "old" measurements have less influence on the final result than the most recent observations. Therefore, we divide all the ratings into two classes: old and new ratings.

The parameters used here are a time window tw and weights th_{new} and th_{old}, where $th_{new} + th_{old} = 1$. The weight with respect to a time window is defined in formula 4.4.

$$w_{tw} = \begin{cases} th_{new} & \text{if } t_i \text{ is within the time window } tw \\ th_{old} & \text{otherwise} \end{cases} \tag{4.4}$$

Note that given sufficient computing and communication capacity, two end users having a similar context should observe similar service quality. The context refers to a set of features which may influence the observation of end users, such as IP address, Internet access type, geographical location, etc. We performed an experiment on PlanetLab[1] to find out the most significant indicators which strongly correlated with service quality, for instance bandwidth. The result shows that the network prefix is the best indicator. Since the same network prefix usually implies the two end users belong to the same computer network or share the same ISP, they have similar routing choices from a local machine to the OFSS provider. The similarity measurement for a network prefix is given by formula 4.5, where e represents a trust evaluator and ref represents a referral who provides rating(s).

$$w_{ip}(e, ref) = \begin{cases} 1 & \text{if the network prefix is the same} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

Regarding geographical similarity, there are two ways for representing the feature of an end user. The first way is to consider an administrative division, which is a portion of a country or other political division, established for the purpose of government. For the purpose of our work, the division structure is separated into 4 layers: country layer, province/state layer, city layer and town/village layer. Fig. 4.2 illustrates this hierarchy.

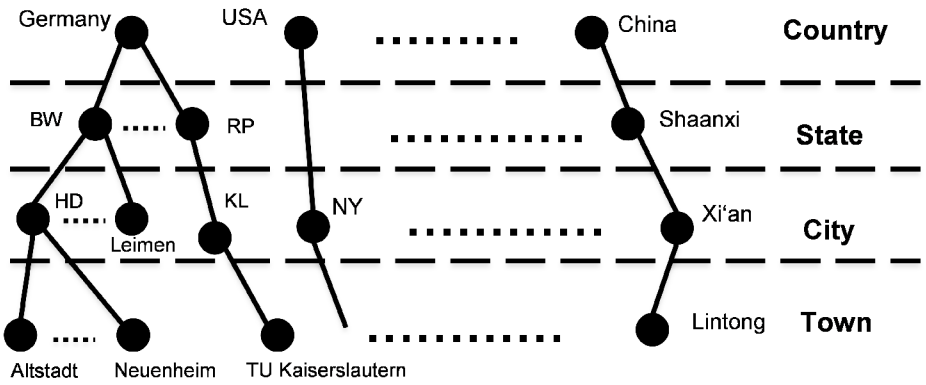

Figure 4.2: Example of a network management hierarchy

[1] www.planet-lab.org

Based on the structure information, we develop a weight representing the geographical similarity of nodes. Note that the structure is equivalent to a forest in graph theory. The measurement of similarity of two nodes is to compare two vertices in a forest. The measurement is implemented by Algorithm 2, where the function parent() returns the parent vertex of two nodes. e represents a trust evaluator and ref represents a referral who provides rating(s). v_e and c_{ref} represent the corresponding vertices. The function $DIS()$ calculates the number of hops between two vertices. The only parameter of this algorithm is μ which is in the interval $[0, 1]$; μ controls how quickly the weight decays exponentially. The reason why the similarity function is implemented by an exponential function is that it follows the intuition that the longer the distance between the vertices and their common parent, the less similar they are.

Algorithm 2: Measure geographical similarity

input : A forest regarding administrative divisions F, α, μ, two vertices v_e and v_{ref}
output: similarity measurement $w_{geo}(e, ref)$
if v_e *and* v_{ref} *overlap* **then**
  $w_{geo}(e, ref) = 1$;
else
  if $parent(v_e, v_{ref}) = \emptyset$ **then**
    $w_{geo}(e, ref) = 0$;
  else
    $w_{geo}(e, ref) = \mu^{MAX(DIS(parent(v_e, v_{ref}), v_e), DIS(parent(v_e, v_{ref}), v_{ref}))}$;
  end
end

An alternative measurement of geographical similarity is to consider the great-circle distance d between two points on the earth. The similarity measurement is given by formula 4.6, where MAX is the largest circle distance between two points on the equator (approximately equal to 20,000 km). The standard formula for calculating the great-circle distance d can be found in (83).

$$w_{geo}(e, ref) = 1 - \frac{d}{MAX} \tag{4.6}$$

The trust model for the failure rate is defined by formula 4.8, where i represents the index of a rating and eva_i represents the evaluator (referral) providing the rating i. $w(i, e)$ represents the weight given to a rating i regarding a trust evaluator e. r_i^s represents a rating i of a service s. The value of r_i^s is equal to one for an unsuccessful file transfer and zero for a successful file transfer.

$$w(i, e) = w_{tw}(i) * w_{ip}(e, eva_i) * w_{geo}(e, eva_i) \tag{4.7}$$

$$TM_{FR}(e, s) = \frac{\sum_{i=1}^{R(s), r_i^s=1} w(i, e)}{\sum_{i=1}^{R(s), r_i^s=1} w(i, e) + \sum_{i=1}^{R(s), r_i^s=0} w(i, e)} \tag{4.8}$$

4.1.3 Trust Models for Network Bandwidth

There is no sound mathematical model for evaluating trust from observations using a continuous scale. However, if we discretize the value range for bandwidth, then a parameterized Dirichlet-multinomial model can be applied. A Dirichlet-multinomial model is a candidate for modelling trust when the observation is discrete (36). Empirically, the range of bandwidths is divided into five intervals which are illustrated in Fig. 4.3, where the outcomes of an observation are transformed into a set {very slow, slow, acceptable, good, fast}.

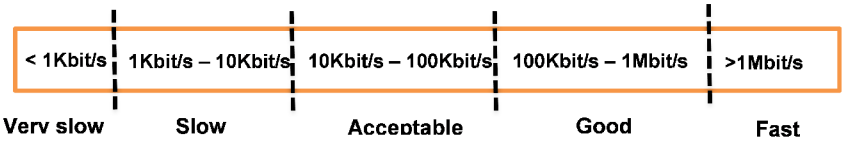

Figure 4.3: Bandwidth discretization

After discretization, a Dirichlet-multinomial model (36) is used for defining a trust model for bandwidth. The Dirichlet distribution is given by formula 4.9, where c_i represents the counts of outcome i after a number of multinomial trials. C represents the vector of observation counts of each outcome. K represents the cardinality of the outcome set and Γ represents the gamma function. P stands for a vector of random variables, with p_i denoting the probability of an outcome i.

$$f(P|C) = \frac{\Gamma(\sum_{i=1}^{K} c_i)}{\prod_{i=1}^{K} \Gamma(c_i)} \prod_{i=1}^{K} x_i^{c_i - 1} \tag{4.9}$$

The expected value of the Dirichlet distribution is given by formula 4.10. We assume a uniform prior distribution by setting $c_i = 1$ for all i.

$$E[p_i|C] = \frac{\alpha_i + c_i}{\sum_{j=1}^{K} (\alpha_i + c_j)} \tag{4.10}$$

The Dirichlet-multinomial model is suitable because it captures the idea that N independent draws from a categorical distribution with K categories are made. Regarding the attribute bandwidth, one draw corresponds to bandwidth experienced for one file transfer; categories correspond to the five outcomes. Using the ideas captured in formulae 4.4 4.5 4.6 and Algorithm 2 we can generate the weight functions for the attribute bandwidth in the same way.

Particularly, Internet access bandwidth should be considered, since the mismatch between an evaluator and the corresponding referrals results in different observations. For instance, even if the server-side bandwidth of an OFSS is extremely fast, a consumer can only experience the client-side bandwidth restricted by e.g. dial-up access. Therefore, it is necessary to measure the similarity regarding Internet access bandwidth. If the Internet Access Bandwidth (IAB) observed by the evaluator e is larger

than the IAB observed by the referral ref, this observation is ignored. The similarity weight function is given by formula 4.11.

$$w_{ia}(e, ref) = \begin{cases} 1 & \text{if } IAB(e) \leq IAB(ref) \\ 0 & \text{otherwise} \end{cases} \tag{4.11}$$

The trust model for bandwidth is defined by formulae 4.12 and 4.13. $w(i, e)$ represents the weight given to a rating i regarding a trust evaluator e. In formula 4.13, the function $argmax()$ calculates the maximal value with respect to an outcome i among all the possibilities of outcomes, i.e. the set very slow, slow, acceptable, good, fast.

$$w(i, e) = w_{tw}(i) * w_{ip}(e, eva_i) * w_{geo}(e, eva_i) * w_{ia}(e, eva_i) \tag{4.12}$$

$$TM_{BW}(e, s) = \arg\max_i \left(\frac{1 + \sum_{j=1}^{R(s), r_j^s = c_i} w(j, e)}{\sum_{k=1}^{K} \left(1 + \sum_{j=1}^{R(s), r_j^s = c_k} w(j, e)\right)} \right) \tag{4.13}$$

4.1.4 Trust Models Considering Social Networks

When information provided by end users regarding the service under consideration is available from social networks, it can be included into a trust model. A simple approach would look like this:

a) A TMS broadcasts an observation of a user to others.

b) Users rate each others' observation by comparing them with their own findings and then explicitly agree (+) or disagree(-).

c) Representing each user as a vertex in a graph, an edge is created if two users agree with each others' observations. The resulting graph represents the social network's "opinion" of the resp. service. It also allows an estimate of a user's "trustworthiness" based on the amount of agreement expressed by the other users; this is captured by w_{so}^i in formula 4.14, where $d_G(i)$ represents the degree of vertex i.

$$w_{so}^i = \frac{d_G(i) + 1}{2|E(G)| + |V(G)|} \tag{4.14}$$

Considering the new weighting function, the original formulae 4.7 and 4.12 can be easily modified to use in the trust models considering social network. We define the corresponding weighting function for failure rate in formula 4.15 by adding the weight w_{so}^i in the original formula 4.7. For bandwidth, the new weighting function is defined in formula 4.16. The original formulas 4.8 and 4.13 remain to build up the corresponding trust models considering social networks.

$$w(i, e) = w_{tw}(i) * w_{ip}(e, eva_i) * w_{geo}(e, eva_i) * w_{so}^i \tag{4.15}$$

$$w(i, e) = w_{tw}(i) * w_{ip}(e, eva_i) * w_{geo}(e, eva_i) * w_{ia}(e, eva_i) * w_{so}^i \tag{4.16}$$

4.1.5 An Example of Trust Evaluation of an OFSS

According to the idea of representing trust for service provisioning introduced in section 3.2, there are two approaches that could be applied: a utility function and a multi-dimensional representation. Here we demonstrate the application of both approaches for an OFSS.

The definition of a utility function is based on the assumption of additive independence (79), where the utility of any given outcome can be broken down into the weighted sum of attributes. There are different ways for eliciting the preference structure of a consumer (84). After all of the parameters of the utility function given by formulae 4.17 and 4.18 have been quantified, the trust value is calculated. In formula 4.17, the trust value TV is represented by a utility function $U(e, s)$, where w_i represents the weighting function, tm_i is the trust model regarding attribute i, and VF is called value function (80). VF refers to a mapping from the domain of trust model regarding an attribute to one-dimension satisfaction space. The valid part is determined by imperative statement, which refers to the preference that cannot be violated. The value function is defined in formula 4.18, where 1 stands for total satisfaction and 0 for total dissatisfaction.

$$TV(e,s) = U(e,s) = \sum_{i=1}^{|A_s|} w_i * VF_i(tm_i(e,s)) \tag{4.17}$$

$$VF(x) = \begin{cases} 1 & \text{if } x \in \text{ valid part} \\ 0 & \text{otherwise} \end{cases} \tag{4.18}$$

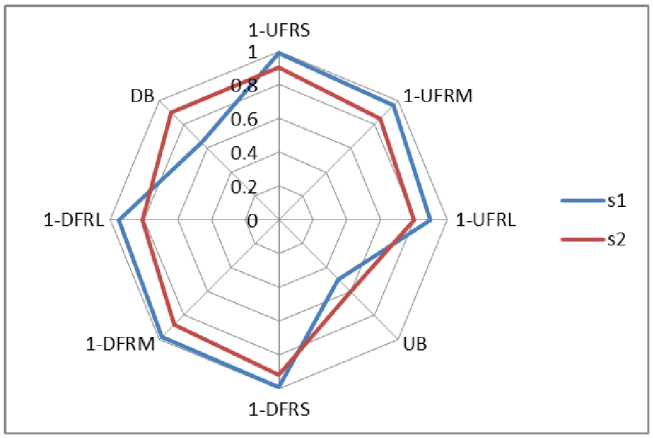

Figure 4.4: Trust evaluation on OFSS

The multi-dimensional representation approach bypasses the integration of attribute dimensions by presenting all the dimensions directly and letting consumers make a decision. Let us look at an example. Let s_1 and s_2 represent two OFSSs. The set of attributes is UFRS, UFRM, UFRL, DFRS, DFRM, DFRL, UB, DB referring back to subsection 4.1.1. There are eight trust models corresponding to eight attributes. The

general trust model for failure rate given by formula 4.11 is refined to six trust models. Similarly, the general trust model for bandwidth given by formula 4.16 is refined to two trust models.

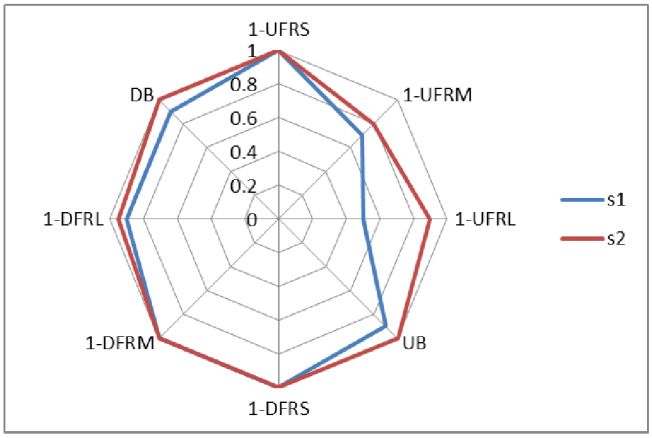

Figure 4.5: Confidence metrics

Figs. 4.4 and 4.5 show an example of trust evaluation in OFSSs and the corresponding confidence evaluation regarding service s_1 and s_2. Instead of presenting trust values for the failure rate directly, we plot the trust value of success rate, which is equal to one minus the trust value of failure rate. According to the regulation of bandwidth discretization, we have separated bandwidth into five intervals. Therefore this scale can be transformed into the real interval in the range [0,1] and we can map all eight dimensions of trust value in one spider diagram. As we can see in Fig. 4.4, s_1 provides better service quality than s_2 in terms of the failure rate. However, s_2 provides higher bandwidth for both upload and download operations. Moreover, s_2 has a higher confidence evaluation than s_1 in all the eight dimensions shown in Fig. 4.5. If a consumer values bandwidth over failure rate, s_2 is the better choice.

4.2 Manipulation Detection Mechanisms for OFSSs

4.2.1 Baseline Sampling (BS)

A TMS collects all the ratings from end users, aggregates them and calculates the trust value. As soon as it gains acceptance, it is likely to become a target for manipulation. The term "manipulation" in the context of TMSs (51, 52) refers to any attempt at influencing or controlling the evaluation of trust. In this work, we mainly focus on two types of manipulation, namely promoting and slandering. The motivation for manipulation can vary: A service provider may want to increase the ranking of his own service; conversely, a competitor may try to decrease the ranking of his rival.

Baseline sampling is the main technique used for detecting and filtering manipulative behavior. Baseline Sampling (BS) aims at generating a reference pattern for normal behavior from a set of sample users. This pattern can then be compared with

actual user activities with statistical methods in order to detect anomalies which may hint at manipulative behavior. Note that, the set of users here is defined differently. In this section, the whole user set represents all the users who share a similar context defined in 4.1.2. A sample user is a user who is trusted by a TMS. Therefore, the reference patterns generated by trusted users define the characteristics of non-manipulative behavior. By creating a baseline sample, a root of trust is built up and all the trust propagations can be made based on this root. The difference between observations from all the users and sample users reflects the precision of prediction regarding trust evaluation. By comparing a baseline with the distribution of observations obtained from the whole population, it is possible to identify manipulative intentions, which results in removing the ratings given by such users from the TMS. The question is when the difference of a reference pattern from the baseline is significant so it can be assumed to reflect manipulation.

Considering observations of all the honest users as a subset of the population, we can extract some samples from the subset. In order for the samples to be representative, a number of "trusted" users are selected based on statistical theory (85). The basic idea is to create a statistic, e.g. sample mean and sample variance, from a small number of "trusted" users, and compare it to the same statistic generated by ratings (observations) from the whole population which might contain dishonest users. There are two approaches for implementing a comparison algorithm. Either a confidence interval is used to indicate the reliability of an estimate of a baseline. Or we compare the shape of a baseline distribution to the whole population of users.

The first approach is uses a confidence-interval-based BS. Under the condition of independence among observations, a sequence of file transfers can be considered as n Bernoulli trials which fulfill the binomial distribution $B(n, p)$. Since the failure rate is supposed to be very low, the probability of success of a file transfer p is very close to 1. Hence the number of failures in a small time interval can be approximated by a Poisson distribution $Pois(\lambda)$. An interval T, e.g. one month, can be partitioned into subintervals t_i of one day. t_i is short enough to capture the possible failure events during the whole interval T. λ is a positive real number denoting the expected number of failures in T. The BS aims to find out the parameter λ by sampling. In the design of BS we define a random sample with size m in which the set of observations is $\{X_1, X_2, X_3, ..., X_m\}$. λ is estimated by formula 4.19.

$$\hat{\lambda} = E(X) = \bar{X} = \frac{\sum_{i=1}^{m} X_i}{m} \tag{4.19}$$

In accordance with the central limit theorem and the rule of thumb, if m is larger than or equal to 30, then the normal approximation Z will be satisfactory to model the distribution of estimated λ regardless of the shape of the population distribution (86). A normal approximation refers to the transformation that a distribution is represented approximately by a normal distribution. The shape of the population distribution is a Poisson distribution in this case. The confidence interval of λ is given by formula 4.20, where t denotes a t-distribution and α stands for the significance level of a test (86).

$$\hat{\lambda} - t_{\frac{\alpha}{2},m-1} * \frac{S}{\sqrt{m}} \leq \lambda \leq \hat{\lambda} + t_{\frac{\alpha}{2},m-1} * \frac{S}{\sqrt{m}} \tag{4.20}$$

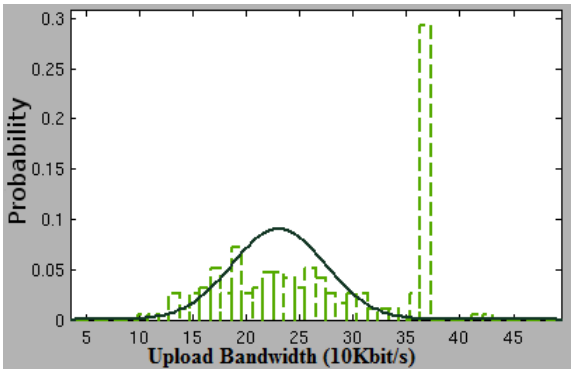

Figure 4.6: Graph based BS with respect to upload bandwidth

Let M ($M >> m$) be the number of all the users who have a similar context, and n_i be the number of trials for one user i in T. It is reasonable to assume that when all the n_i are sufficiently large and q_i sufficiently small, then all the X_i, where i is a natural number between 1 and M, can be considered random variables of the same Poisson distribution. We assume the expected value of number of the failures occurring in T to be a constant. The assumption of a Poisson distribution is reasonable, since an OFSS provider treats all the users who have a similar context in an unbiased way, and the Internet has an oscillating impact on each end users in terms of failure rate and bandwidth.

Regarding bandwidth, context-based trust evaluation implies the distribution of user-observed bandwidth fulfills a normal distribution $N(\mu, \sigma^2)$. Let the number of trusted users be m. Then the interval estimation of mean and its confidence interval regarding the attribute of bandwidth are given by formulae 4.21 and 4.22.

$$E(X) = \bar{X} = \frac{\sum_{i=1}^{m} X_i}{m} \tag{4.21}$$

$$\bar{X} - t_{\frac{\alpha}{2},m-1} * \frac{S}{\sqrt{m}} \leq \mu \leq \bar{X} + t_{\frac{\alpha}{2},m-1} * \frac{S}{\sqrt{m}} \tag{4.22}$$

The second approach is a graph-based BS. First, we plot the graph of probability density function (PDF) or probability mass function (PMF) with respect to number of failures and bandwidth provided by trusted users. Then we compare the shape of the graph with the corresponding graph of PDF or PMF with respect to the whole population of users. Note that the graph regarding the whole user population is difficult to be fitted to a distribution, since it is expected to contain fraudulent observations. Instead, we replace the PDF graph generated from the whole population with a histogram. We compare the shape of the graph generated from observations of trusted users with that of the histogram. In addition, by using confidence-interval-based BS, sometimes the manipulative behavior can be ignored. For instance, assume that a number of dishonest users report the bandwidth is 10Kbit/s for slandering, while other dishonest users report 1Mbit/s for promoting. The confidence interval can be between 10Kbit/s and 1Mbit/s and so does the mean of honest users. Usually it is impossible to detect the

case that both types of manipulation are operating simultaneously using confidence-interval-based BS. Therefore graph-based BS is superior to confidence-interval-based BS.

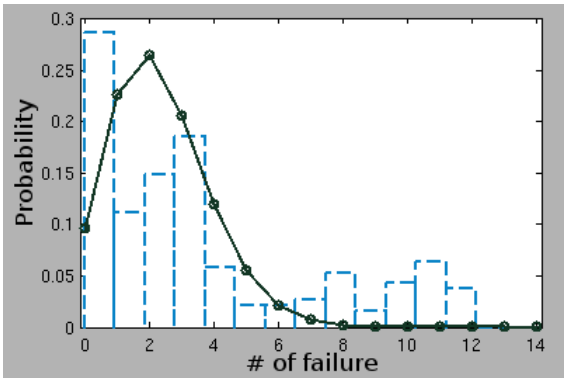

Figure 4.7: Graph based BS with respect to upload failure rates

Figs. 4.6 and 4.7 illustrate to cases using graph-based BS. In Fig. 4.6, the horizontal represents the observed value of upload bandwidth in unit of 10 Kbit/s. The vertical axis shows the corresponding probability. The black curve depicting a normal distribution corresponds to BS. The blue histogram represents the distribution generated by the whole population. The extremely high bar at 370 Kbit/s means there are many users observing a bandwidth is as large as 370 Kbit/s, which is not supported by the base line. This is a typical example of manipulative behavior (promoting). Fig. 4.7 shows both promoting and slandering are attempted with respect to the failure rate. As we can see, the mean value of the Poisson distribution for the base line is 2. However there are two abnormal regions where some users report zero failures and some report more than eight failures.

We propose algorithms 3 and 4 for detecting manipulative behavior with respect to bandwidth and failure rate, respectively. The basic idea for the detection algorithm for manipulation of bandwidth measurement is firstly to find a point rep which has the highest probability within the interval $[mean_{bs} - \sigma_{bs},\ mean_{bs} + \sigma_{bs}]$ in the histogram. rep is a representative point which is supposed to have the highest probability in the whole histogram, because it is inside of the confidence interval of the base line and it has a high probability. $mean_{bs}$ and σ_{bs} are the parameters of a normal distribution with respect to BS. The algorithm then compares rep with the points outside the interval, if there exists a point x in the histogram such that $p(x)/p(rep)$ is larger than the pre-defined threshold th_{normal} which is larger than one, then we consider the point as evidence of manipulation. This threshold follows the intuition that when a point which is outside of the interval $[mean_{bs} - \sigma_{bs},\ mean_{bs} + \sigma_{bs}]$ and has a higher probability, it can be the indicator of manipulative behavior. Because based on the shape of distribution with respect to the base line, the probability is extremely low.

The idea of the representative point is also used for finding attempts at promoting a service provider with respect to failure rate in algorithm 4. Here, the representative point is called rep_{pro} and the interval is $[mean - 1,\ mean + 1]$. Furthermore, in order

to find the evidence of slandering manipulation, we identify the second representative point rep_{sla} which is defined by the distribution for very unlikely events, i.e., $p(rep_{sla}) = poiss(rep_{sla}|\lambda) \approx 0$, for instance $p(rep_{sla}) < 0.01$. If there exists a point x in the histogram, that $p(x)/p(rep_{sla})$ is larger than a pre-defined threshold th^{sla}_{poiss}, then we consider the point as the evidence of slandering manipulation. The similar justification can be given here as above.

Algorithm 3: Graph based BS regarding bandwidth

```
input  : pdf regarding BS p_bs(x) and pdf regarding the whole population p(x)
output: manipulation type: ManiType
Initialization: ManiType := unknown;
Find a point rep which has the highest probability in
[mean_bs − σ_bs, mean_bs + σ_bs] ;
for all the x do
    if (x − mean_bs) > +σ_bs AND p(x)/p(rep) > th_normal then
        ManiType := promoting;
    else
        if (mean_bs − x) < −σ_bs AND p(x)/p(rep) > th_normal then
            if ManiType == unknown then
                ManiType := slandering;
            else
                ManiType := both;
            end
        end
    end
end
```

A key issue of BS is to create and manage a small set of trusted users according to the dynamic properties of a target system. There are three properties to be considered: the size of a group, frequency of file transfer and variation of service quality.

The size of a group refers to the number of end users in a group which is formed considering context such as geographical location, network prefix, etc. Regarding the size, in the case of practical interest (86), if the number of trusted users is larger than 30 the normal approximation for a distribution of sample mean will be satisfactory regardless of the shape of the population. If the number is less than 30, the central limit theorem plays a role if the distribution of the population is not severely non-normal.

The frequency of file transfer with respect to a normal user is changing over time. Due to an assumption when using the Poisson distribution regarding failure rate, one observation per subinterval, e.g. per day, is necessary to capture possible failures during an interval T (e.g. one month). Regarding bandwidth, more frequent observations are required due to the assumption that service quality may vary significantly during an

interval T.

Algorithm 4: Graph based BS regarding failure rate

input : pdf regarding BS $p_{bs}(x)$ and pdf regarding the whole population $p(x)$
output: manipulation type: ManiType
Initialization: $ManiType := unknown$;
Find a point rep_{pro} which has the highest probability in $[mean_{bs} - 1, mean_{bs} + 1]$;
Find a point rep_{sla} which is larger than $mean_{bs} + 1$ and $poiss(rep_{sla}|\lambda)$;
for *all the* x **do**
  if $x < mean_{bs}$ AND $\frac{p(x)}{p_{bs}(x)} > th^{pro}_{poiss}$ AND $\frac{p(x)}{p(rep_{pro})} > 1$ **then**
    $ManiType := promoting$;
  else
    if $x > rep_{sla}$ AND $\frac{p(x)}{p(rep_{sla})} > th^{sla}_{poiss}$ **then**
      if $ManiType == unknown$ **then**
        $ManiType := slandering$;
      else
        $ManiType := both$;
      end
    end
  end
end

The service quality may vary in the long term, but once the assumption about distribution with respect to failure rate and bandwidth is fulfilled and the interval T is well selected, BS is capable of capturing service quality variation and resisting manipulation. Based on the discussion above, the management of trusted users is specified as follows:

a) select qualified trusted users initially;

b) check the quality of trusted users periodically. The quality of a trusted user contains frequency of file transfer, distribution of file transfer time per day, distribution of size of transferred file, etc. If there is at least one unqualified user, the TMS executes an update operation. An update operation is defined as an operation which replaces each unqualified user by a new qualified user.

4.2.2 Clique Identification

When social network information is available for users involved in a TMS, this information can be used to identify manipulative groups. Assume that one user could submit a report about whether he/she trusts the observation of another user. On the one hand, for the sake of increasing trustworthiness of dishonest users, they give positive ratings to their own company. On the other hand, honest users give ratings depending on how similar their observation is with respect to the other users. Following this idea, it is

possible to find cliques of dishonest users which have stronger association than other cliques.

Two notions of social network analysis are used to model the concept of a clique and intensity of association in a clique. N-clan is defined as a set of users such that n is the maximum path length via which members of the clan are connected, and all nodes on the paths are members of the clan (87). In our work, 2-clan is used to define a notion of clique. A distance of 2 can simply be interpreted as the distance between a node and a functional neighbor such as, for example, an intermediary or a broker. Path lengths greater than two, however, are more difficult to interpret (87). Furthermore, the intensity of the association in a clique is defined as the density of the sub-graph corresponding to the clique defined as formula 4.23, where l is the number of edges involved and n is the number of nodes in the clique.

$$density = \frac{l}{\frac{n(n-1)}{2}} \tag{4.23}$$

Fig. 4.8 illustrates the structure of a simulated social network with respect to OFSS. We develop a simulating platform which randomly generates all the users including honest and dishonest users and the corresponding ratings. In the figure, one node represents a user in a TMS for an OFSS. The file download bandwidth experienced by a user is encoded by the color of the corresponding node. The graph is generated using a force-directed graph drawing approach (88). The larger density a clique has, the closer users in the clique stay with each other. Note that the ball-shaped sub graph which is plotted in red. The nodes in the sub graph stay very close with each other. This sub graph is a clique composed of dishonest users.

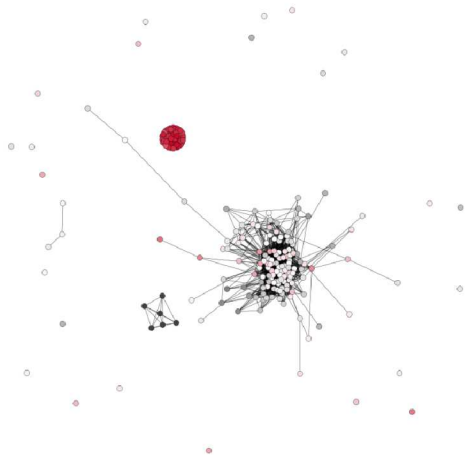

Figure 4.8: Structure of a social network (synthetic example)

4.3 Simulation Results

4.3.1 Experiment Design

We built a simulation platform to evaluate the vulnerability of our trust model to malicious users. It is also used to determine the efficiency of the manipulation detection mechanisms such as graph-based BS, confidence-interval-based BS, and clique identification based on social network analysis (CI). The simulation platform is considered as a flexible and parameterized testbed in which every key feature of the target system is represented by a set of parameters, such as size of user group in which all the users share a similar context, the number of communities for simulation, percentage of dishonest users in a community, the distributions regarding bandwidth and failure rate in a community for honest users, the distributions regarding bandwidth and failure rate observed by a user group for dishonest users, the frequency of file transfer for each user, etc. The framework of the simulation platform is shown in Fig. 4.9. The output "Generated Data" is composed of clean and dirty ratings. Clean ratings refer to the ratings generated by honest users; dirty ratings refer to the ratings generated by dishonest users. Distributions regarding bandwidth and failure rate in a community observed by honest users, are generated from a normal distribution and a Poisson distribution, respectively.

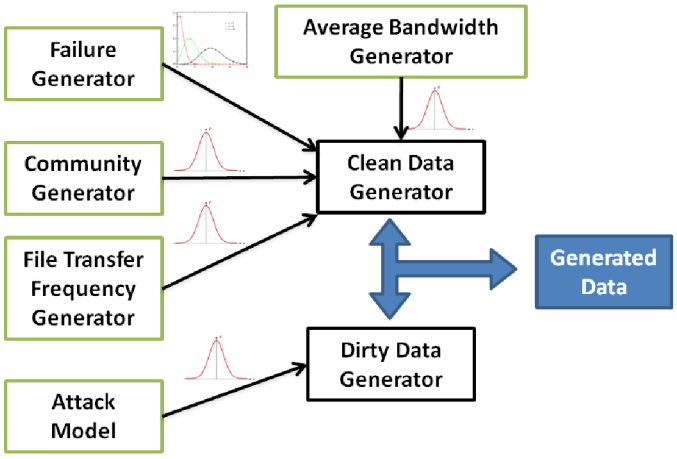

Figure 4.9: The framework of the simulation platform

Moreover, we simulate manipulation in terms of its categorization and intensity. Regarding categorization, three types of manipulation, which are promoting, slandering and a mixture of promoting and slandering, are considered. Regarding intensity, different parameters shown in Table 4.1, are considered depending on the types of manipulation.

For all three types of manipulation, the proportion of dishonest users (DUP) of a community is a key parameter. Regarding promoting, the intensity of manipulation with respect to failure rate is simulated by file transfer frequency (FTF) which stands

Parameter	Value
Promoting	
Dishonest User Proportion (DUP)	10%, 30%, 50%, 70%
File Transfer Frequency (FTF)	2x, 4x, 6x
Bandwidth Offset (BO)	30%, 50%, 70%, 90%
Slandering	
Dishonest User Proportion (DUP)	10%, 30%, 50%, 70%
Number of Failure Offset (NFO)	0, 1, 2, 3
Bandwidth Offset (BO)	30%, 50%, 70%, 90%
Both	
Dishonest User Proportion (DUP)	30%, 50%, 70%
Ratio of Promoter to Slanderer (RPS)	1:9, 3:7, 5:5, 7:3, 9:1
Bandwidth Offset (BO)	90%
Combination of FTF and NFO	(2x, 1), (2x, 2), (4x, 1), (4x, 2)

Table 4.1: Parameters setting for modeling manipulative behavior

for how often files are transferred per day. The intensity with respect to bandwidth is simulated by bandwidth offset (BO) which is calculated by formula 4.24.

$$BO = \frac{|mean_{hon} - mean_{dis}|}{mean_{hon}} \tag{4.24}$$

For instance, if the mean of the bandwidth distribution observed by honest users is equal to 100Kbit/s, then $BO = 0.3$ defines that the mean of the distribution generated by promoters would be 130Kbit/s. Regarding slandering, the intensity in terms of failure rate is characterized by number of failure offset (NFO) per week which represents the difference of the mean values of the Poisson distributions for dishonest users and honest users. For bandwidth the same parameter BO is used. For dealing with both promoting and slandering we reuse all the parameters above. In addition, the Ratio of Promoter to Slanderer (RPS) is specified to characterize the contrast between the two types of manipulation. According to the combination of parameters, 400 test cases without dirty data, 48 for promoting, 64 for slandering and 60 for both are generated.

4.3.2 Simulation Results

For each test case, two time series of trust evaluation are calculated by considering all users on one hand and the honest ones only on the other. Afterwards, the deviation of trust evaluation at time t is calculated by formula 4.25, where $TM(e, s)$ stands for a trust model. The trust model regarding failure rate is implemented by formula 4.8; the trust model regarding bandwidth is implemented by formula 4.13. $TM_t(e, s)$ stands for a trust evaluation at time t. $TM_t^{all}(e, s)$ and $TM_t^{gen}(e, s)$ represent the trust evaluation at time t considering all the users and honest users only, respectively. We specify a representative point in the series as the deviation of trust deviation under manipulation

(Trust Dev). For example take the point which has the highest deviation in the series as the representative point. In this case, each test case is assigned to a value of maximum deviation (MaxDev).

$$Dev_t = \frac{TM_t^{all}(e,s) - TM_t^{gen}(e,s)}{TM_t^{gen}(e,s)} \tag{4.25}$$

In the simulation, we first evaluate how many percentage of manipulation can be detected by using both graph-based BS and confidence-interval-based BS. The results for all types of manipulation on failure rate (FR) are shown in Figs. 4.10, 4.11 and 4.12. In Fig. 4.10 there are 48 points representing 48 test cases for promoting. The points are color coded in terms of MaxDev. The colored scale on the right-hand side relates the colors of the dots to the values of trust deviation. The lighter the color, the larger the trust deviation. The points with dotted circles are the test cases which are not detected as manipulation. As we can see from Fig. 4.10, only points with low trust deviation scores are circled. The fact shows that by using BS the manipulative behavior can influence the trust evaluation within a small range.

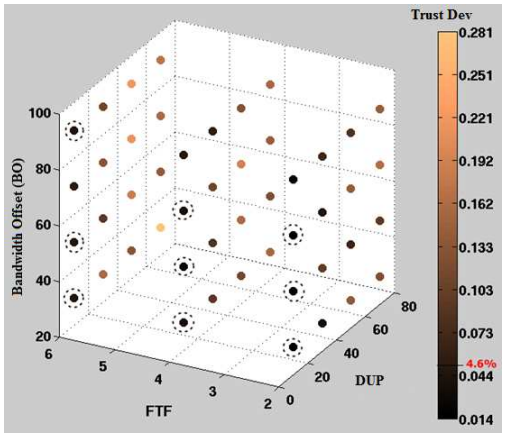

Figure 4.10: Promoting detection using BS regarding failure rate

Figs. 4.10, 4.11 and 4.12 indicate that for the three types of manipulation regarding FR, the MaxDev of a TMS using BS turns out to be much less than that without using it. Because if the strength of manipulation is too large, the manipulative behavior will be detected out by BS. This simulation result shows that using BS the manipulative behavior can be restricted in a small range. In addition, none of the 400 test cases without contamination is detected as manipulation by mistake. We show the result for both FR and bandwidth in table 4.2.

Moreover, when social network information is considered, the advanced trust models introduced in 4.2.4 and clique identification (CI) technique are involved. The differences between the two approaches with respect to advanced trust models are indicated by Figs. 4.13, 4.14, 4.15, 4.16, 4.17 and 4.18.

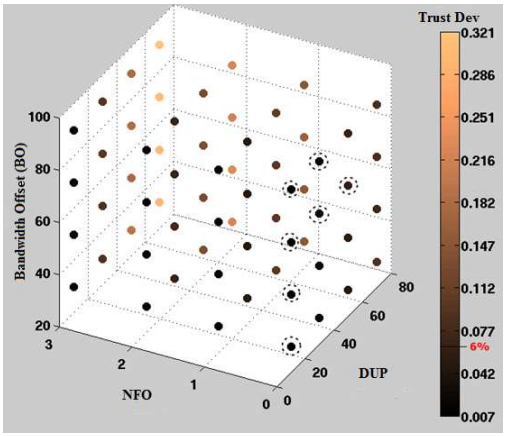

Figure 4.11: Slandering detection using BS regarding failure rate

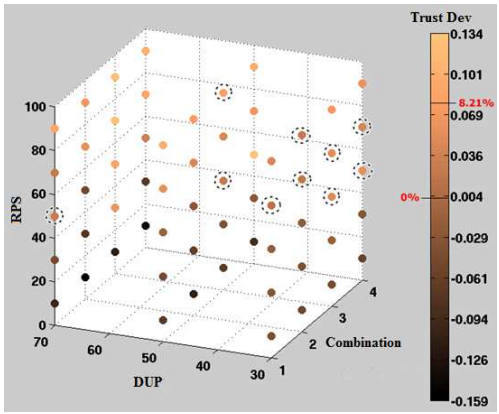

Figure 4.12: Both promoting and slandering detection using BS regarding failure rate

Manipulation Type	MaxDev without Detection	MaxDev with BS
Promoting on FR	28.1%	4.6%
Slandering on FR	-32.1%	-6%
Both on FR	-15.9% $\sim$ 13.4%	0% $\sim$ 8.21%
Promoting on Bandwidth	68.3%	14.5%
Slandering on Bandwidth	-46%	-10%
Both on Bandwidth	-24.8% $\sim$ 59.6%	-10% $\sim$ 15%

Table 4.2: Simulation results for baseline sampling (BS)

In Figs. 4.13 and 4.14 we show the results for detecting of promotion for bandwidth using BS and CI, respectively. There are 48 points representing 48 test cases for promoting like previous experiment. The points are color coded in terms of the maximum deviation. The points with dotted circles are the test cases which are detected as manipulation. As we can see from Fig. 4.13, only two points with very low trust deviation are circled. In Fig. 4.14, it is shown that CI works better than BS by successfully detecting one more test case.

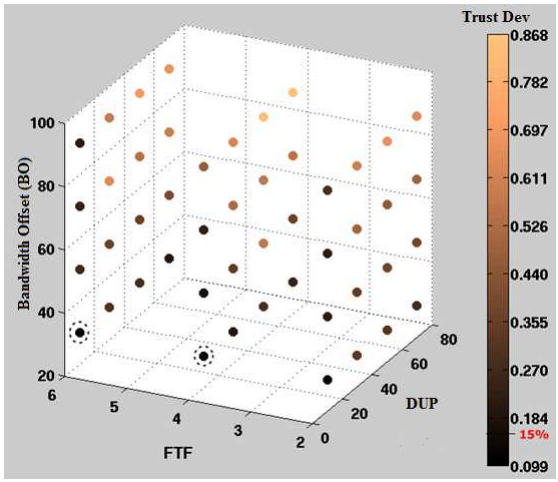

Figure 4.13: Promoting detection using BS regarding bandwidth

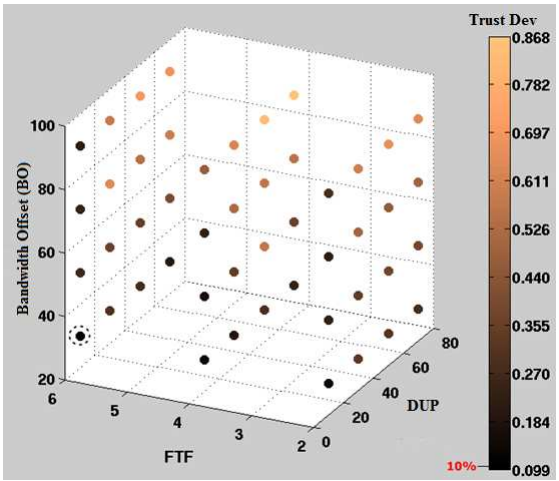

Figure 4.14: Promoting detection using CI regarding bandwidth

In Figs. 4.15 and 4.16, we show the results for slandering detection for bandwidth using BS and CI, respectively. There are 64 points representing 64 test cases for slandering like previous experiment. The points are color coded in terms of the maximum deviation. The points with dotted circles are the test cases which are not detected as

manipulation. As we can see from Fig. 4.15, 11 points with very low trust deviation are circled. Fig. 4.16 shows that CI works better than BS by successfully detecting four more test cases.

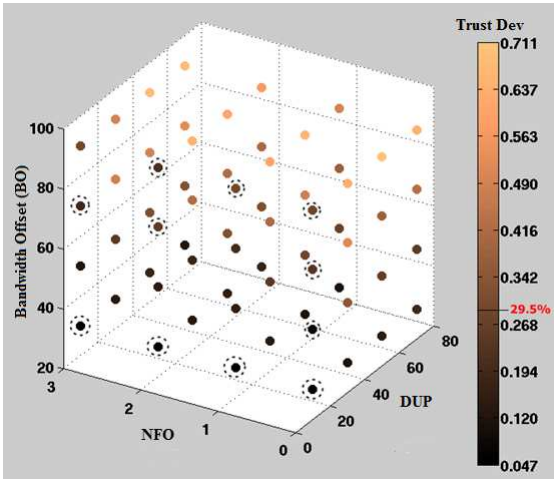

Figure 4.15: Slandering detection using BS regarding bandwidth

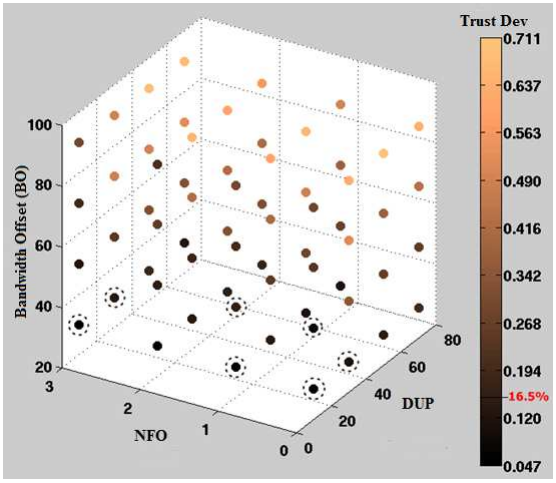

Figure 4.16: Slandering detection using CI regarding bandwidth

In Figs. 4.17 and 4.18, we show the results for mixed promoting/slandering detection for bandwidth using BS and CI, respectively. There are 64 points representing 60 test cases. The points are color coded as before. The points with dotted circles are the test cases which are not able to be detected as manipulation. Fig. 4.17 indicates the case considering only two BS approaches, which are confidence-interval-based BS and graph-based BS. As we can see from Fig. 4.17, four points with very low trust deviation are circled. Fig. 4.18 shows that CI works better than BS by successfully detecting all the test cases for manipulation.

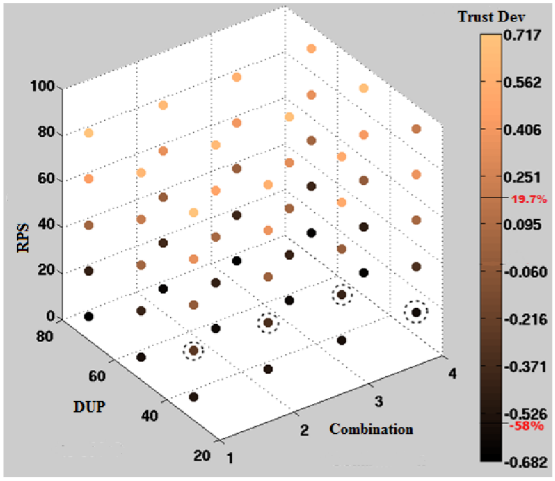

Figure 4.17: Both promoting and slandering detection using BS regarding bandwidth

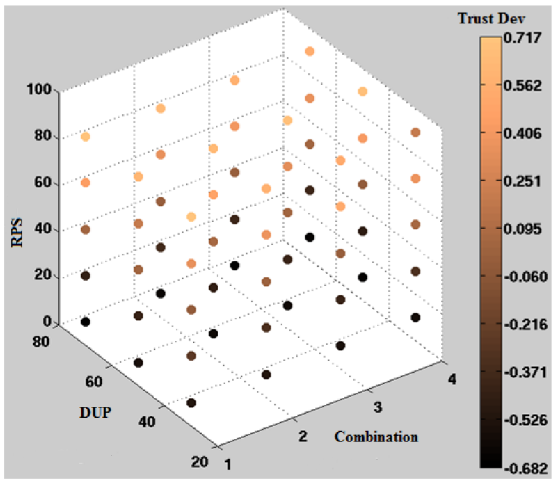

Figure 4.18: Both promoting and slandering detection using CI regarding bandwidth

Manipulation	MaxDev without Detection	MaxDev with BS	MaxDev with CI
Promoting on bandwidth	86.8%	15%	10%
Slandering on bandwidth	-71.1%	-29.5%	-16.5%
Both on bandwidth	-68.2% ~ 71.7%	-58% ~ 19.7%	0%

Table 4.3: Simulation results for detecting manipulative behavior on the attribute "bandwidth"

The range of deviation for all types of attack in terms of bandwidth is shown in Table 4.3. Note the range of MaxDev without considering any detection mechanisms using the trust models considering social network information is much larger than that using the basic trust models only. Since dishonest users rate each other with high value such that the trust models considering social network give more weight to the fraudulent observations than the basic trust models. However, by using CI combined with advanced trust models, the effect of manipulative behavior is largely restricted. For instance, all 60 test cases for both promoting and slandering manipulation are successfully detected. In this particular case, deviation of the TMS is zero. The CI is performed by considering several key parameters such as degree of recognition. Degree of recognition refers to what extent a user agrees with the observation of others. In the simulation, the degree of recognition for dishonest users is fixed to 100%, which means the dishonest users in the same group agree with each other completely. The degree of recognition for honest clients is fixed to 80%.

5

Case Studies on Qualitative Services

5.1 A Case Study for Online Shopping Services

In this section, we choose Taobao.com (the largest Chinese e-commerce website) as a dataset for an online shopping service. Given context of reputation, we can extract characters or assumptions of manipulative behavior from Taobao.com. According to the characters, we present a manipulation detection system, Clustering based Suspect Identification (CSI) that can detect the customers who provide fraudulent ratings, and the vendors who intend to manipulate their trust value. We propose a lightweight trust model, R-Rep, for resisting attempts at manipulation. We suggest two approaches, Ranking Comparison Index (RCI) and Benefit Variation Ratio (BVR), for comparing different trust models with regard to resisting manipulative behavior. Finally the experimental results show that R-Rep outperforms two existing trust models, the trust model employed by Taobao itself and a Beta Reputation System (82). Comparing the statistics of the whole population to the suspicious sub population, we explore characteristics of suspects intensively and discover some patterns and phenomena.

5.1.1 Online Shopping Services and Taobao.com

Online shopping or online retailing (e.g. Amazon.com) is a form of e-commerce service allowing consumers to buy goods or services from a vendor over the Internet. A typical online shopping process consists of the following steps: A customer searches and selects a vendor who provides the product or service the customer wants on an online shopping website. The customer can use a virtual shopping cart to collect multiple items and to adjust quantities. After paying the bill the customer receives an e-mail confirmation after the transaction is complete. The products or services will be delivered to the customers within a promised time interval. The methods of delivery depend on the type of product or service. For digital media products such as software, music, movies

or images, the customer can download them directly; for tickets, codes, or coupons, the customer can print them out; for a product such as clothes and shoes, they are shipped to the customers address.

For our case study we chose Chinese B2C online store Taobao.com[1]. There are several reasons why Taobao was chosen as the target of our research. Taobao is the largest e-commerce platform in China. It reported 250 million registered users and over 700 million products for sale in 2010. In this ecosystem, people can make a large profit by providing fraudulent ratings. Furthermore, there is a community surrounding manipulative behavior exchanging knowledge about a couple of interesting topics, such as the jargon for launching an attack and the possible detection strategies applied by Taobao. There are some special tools[2] assisting manipulation as well. After the investigation of manipulative behavior on Taobao, we conclude three significant assumptions in Taobao.

a) Dishonest customers are apt to provide fraudulent ratings for less expensive product in order to reduce risk.

b) Most of the dishonest customers are not personally known by vendors.

c) The dishonest customers aim at maximizing their profit.

The first assumption is called the "assumption of risk aversion". A statistical study shows that, a vendor usually pays 5-10 cents for an inauthentic review which is associated with a fake transaction (59). A dishonest customer will not take the risk of paying too much and confirming a fake transaction. If a vendor cooperates with the dishonest customer and pays the bill (e.g. 10 Euro) back, the dishonest customer can get 5-10 cents; if not, the customer will lose 10 Euro directly. Regarding the second assumption, due to the huge number of transactions per day, it is impossible for a vendor to maintain such a big friend list. In addition, the second assumption is a prerequisite for the third assumption. Considering a tiny profit, a dishonest customer must provide fraudulent reviews as much as he can in order to earn a fortune. The three assumptions are the root of trust, and our work is built up based on the three assumptions. We believe that an assumption-based solution for manipulation identification is a valid approach due to the complexity of the concept (trust). Given a context of trust, we can capture the basic patterns, which are refined as basic assumptions or characteristics. Based on these assumptions, a concrete trust model can be derived from the trust model framework, which was introduced in chapter 3, and different learning algorithms can be applied to identify different types of manipulation regarding different contexts. In the case of Taobao, we follow the three assumptions above.

The dataset is collected as follows. A set of 1081 target vendors are located by choosing "Nokia Smartphone" as the search keyword. In Fig. 5.1, we can see there are two fitting lines crossing at point two. The steeper line is fitted by the customers who have at most two purchases per day; the other line is fitted by the ones who have more than two purchases per day. They have totally different slopes and based on the third

[1] www.taobao.com

[2] www.schuaxinyong.com, www.tuzi88.com

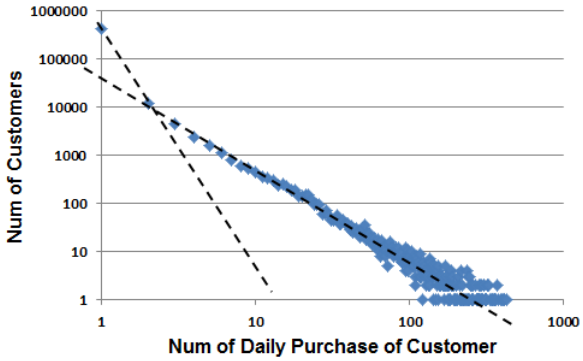

Figure 5.1: Daily volume of purchases vs. number of customers

Data Type	Data Size
Vendor (API)	1,081
Customer (API)	460,095
Customer (Web Crawl)	8,981
Vendor Related Rating (Web Crawl)	1,241,481
Customer Related Rating (Web Crawl)	1,229,352

Table 5.1: Basic statistics of the Taobao dataset

assumption, we believe the customers whose daily purchase number is greater than or equal to three are more suspicious than the rest of the customers. The rating history of a set of 12,657 customers is selected by truncating the number of daily purchase to a minimum of three. Since some customers' rating history is unavailable, we only have rating histories of 8,981 customers. In addition, for both vendors and customers, we record the global attributes, which are gathered via Taobao's APIs, such as reputation value, life span, etc. The general statistics of this dataset are listed in Table 5.1.

5.1.2 Clustering Based Suspect Identification (CSI)

The basic idea of clustering based suspect identification (CSI) is as follows. We assume that the statistical characteristic of a suspicious object (customer or vendor) is distinguishable from that of a normal object. Hence, it is possible to recognize the set of suspicious vendors and customers by analyzing features of their behavior. It is reasonable to assume that there is a strong relationship between suspicious customers and suspicious vendors. Suspicious vendors "hire" suspicious customers to provide fraudulent ratings. We can identify a set of suspicious vendors using this alliance relationship. It is essential to obtain different sets of suspects using anonymous ratings and non-anonymous ratings. An anonymous rating is provided by a customer whose identity is not visible. The union of these sets should identify suspicious activity with higher confidence. The practical contribution of our identification approach is to assist the operator of TMSs in detecting manipulative behavior. It is theoretically possible to verify

the validity of every transaction in a TMS. For instance, Taobao can check whether products or services are genuinely shipped from a vendor to a customer. Practically it is impossible to check every transaction due to the huge number of transactions. Therefore, the detection result delivered by our approach is a suitable method for reducing the number of transactions that need to be checked.

5.1.2.1 Suspicious Customer Identification

We specify five main features for characterizing manipulative behavior of customers.

a) Number of vendors (**NV**). Total number of vendors a customer has transacted with.

b) Average price (**AP**). The mean of the monetary value of transactions.

c) Number of transactions per vendor (**NTV**). The mean number of transactions a customer makes with a vendor.

d) Number of transactions per day (**NTD**). Mean number of transactions that a customer has made in one day.

e) Number of vendors per day (**NVD**). The mean number of vendors, a customer has dealt with per day.

Based on the assumption of maximization of profit, the values of NV, NTV, NTD and NVD for a suspicious customer should be much higher than for a normal customer. In addition, based on the assumption of risk aversion, the value of AP for a suspicious customer should be lower than for a normal customer. We consider suspicious customer identification as a classic unsupervised learning procedure, since nobody can assert that a customer is definitely a "bad" one. Following this idea, we use the k-means algorithm in weka[1] to cluster vendors and customers. We tried different sets of parameters and the final result is given in Table 5.2. The first two clusters are considered to be suspicious. Cluster one (C1) has a very high value for NTV, 3.1708. By contrast, C1 has low value for AP, which is 0.0734. Cluster two (C2) is considered as a suspicious group because of higher values for NV, NVD and NTD and a small value for AP. On the other hand, clusters C3, C4, C5 and C6 do not show similar characters to C1 and C2 in terms of value for NV, NTV, NTD, NVD and AP. In total, 1616 suspicious customers have been identified.

5.1.2.2 Suspicious Vendor Identification

We propose three different methods for identifying suspicious vendors in this section. The first method explores the alliance relationship between a suspicious customer and a suspicious vendor, and derives the suspicious vendors from the suspicious customers. The second method aims at analyzing the statistical features of behavior with respect to vendors by considering only non-anonymous ratings. The third method identifies suspicious vendors by considering only anonymous ratings.

[1] A collection of machine learning algorithms for data mining tasks.

	NTV	NV	AP	NVD	NTD
C1 (1603)	**3.1708**	0.2022	**0.0734**	0.0447	0.1965
C2 (13)	0.8928	**11.4335**	**0.018**	**17.6873**	**18.4477**
C3 (20)	1.1426	0.0083	15.6623	0.0321	0.0338
C4 (4497)	0.7946	0.2004	0.4905	0.2344	0.241
C5 (542)	0.7619	0.0845	2.3808	0.1592	0.1618
C6 (2306)	1.8195	0.1954	0.1823	0.065	0.1601

Table 5.2: Customers clustering results

a) From customers to vendors

Once the set of suspicious customers is identified, it is not difficult to find the corresponding set of suspicious vendors. If a customer account is used to provide fraudulent ratings, the probability that the person behind also uses this account for regular transactions is very low, because usually people do not want to mix their serious business and regular life together. The number of transactions which a suspicious customer has performed with a vendor is a key feature for discovering an alliance relation. If a suspicious customer has traded with a vendor very often e.g. 10 times per day, then the vendor is considered as suspicious as well. By choosing a threshold for the number empirically, i.e. three in our evaluation, 1616 suspicious customers correspond to a set of 63 suspicious vendors (**SV1**).

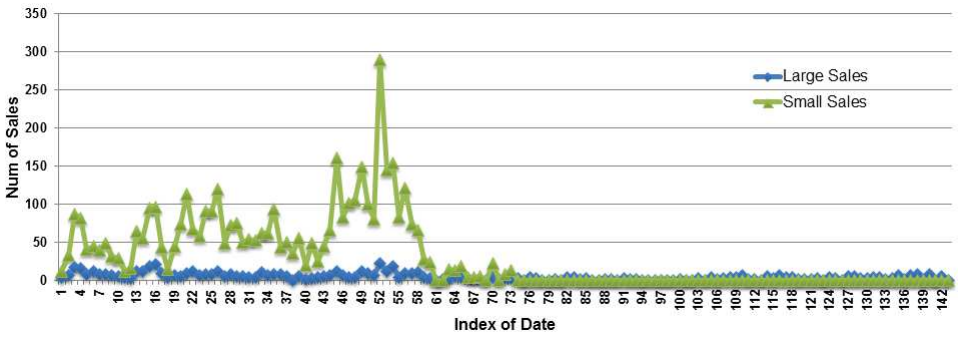

Figure 5.2: Time series of electronic product sales volume

b) Considering non-anonymous ratings

SV1 is derived from analyzing the relationship between suspicious customers and vendors, where the information of non-anonymous ratings is involved. In this case, anonymous ratings cannot be used to relate a vendor to a customer. We can perform an analysis on the same type of information from the perspective of a vendor. Two main features for characterizing the manipulative behavior of a vendor are specified.

- Ratio of small to total sales (**RSTS**). RSTS indicates the proportion of small transactions among all the transactions.

	RSTS	M-Day
C1(295)	**2.2726**	**1.3544**
C2(786)	0.3014	0.121

Table 5.3: Results of vendors clustering

- **M-Day**. M-Day is the short name for the number of days when the manipulative behavior probably occurs. It is necessary to analyze vendor's behavior based on temporal information. Manually we identify some suspicious vendors by analyzing inconsistent purchase behavior such as the one shown in Fig. 5.2. At the beginning, in order to accumulate trust value, the vendor generates a large number of fraudulent ratings on some cheap products, such as some accessories of a cell phone. With the increase of trust, it is not necessary for him to promote the trust value any more. Therefore after around 60 days, the vendor stops manipulation and behaves normally. In order to capture the character of manipulation, we specify two sub features on daily base.
  - Ratio of small to total sales regarding time series (**RSTS_TS**). The definition of RSTS_TS is very similar to that of RSTS, except that the former operates at the daily level.
  - Average number of transactions per customer regarding time series (**NTC_TS**). This feature calculates the average number of transactions that a customer makes with the vendor on a given day. The value of NTC_TS is equal to the volume of sales divided by the number of customers who have traded with the vendor on that day.

We tried different values for parameters regarding k-means (e.g., Euclidean distance function, number of clusters, maximum iteration number, etc.) and the final result is given in Table 5.3. A set of 295 suspicious vendors (**SV2**) is identified.

The union of SV1 and SV2 is not the final result. The basic idea of identifying suspects regarding non-anonymous ratings is to gather the evidence of different aspects, and consider them all. In order to reinforce our conclusion, we combine the two results by an intersection operation. Finally we obtain a set of 57 suspicious vendors (**SV3**).

c) Considering anonymous ratings

Some vendors gain trust by obtaining fraudulent anonymous ratings due to the characteristics of anonymous ratings. An anonymous rating can be created in two manners in Taobao. In the first case, all the ratings on a virtual product, such as software, pre-paid card, phone card, etc., are automatically labeled as anonymous. The relationship between the customer and the vendor of an anonymous rating is not traceable. The second case is that a customer sets the state of a rating as anonymous, maybe because of privacy concerns. This, however, happens very rarely. The greatest advantage of manipulating trust by providing an anonymous rating is the untraceability of the anonymous rating, since there is not a physical

delivery process. However, it is questionable that a vendor who sells mobile phones has achieved a great number of virtual product transactions.

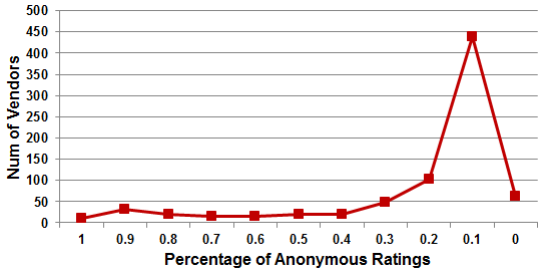

Figure 5.3: The distribution of vendors over the percentage of anonymous ratings

Identification of suspicious vendors considering anonymous ratings is performed by truncating the distribution of vendors over the percentage of anonymous ratings. Fig. 5.3 shows that, most of the vendors have a small percentage of anonymous ratings. We identify a set of 27 vendors (**SV4**) by setting the threshold to 95%. The intersection of SV1 and SV4 is empty, and so is the intersection of SV2 and SV4. The empty result is expected, since there is little overlap between different types of suspicious vendors detected by using different types of information (non-anonymous and anonymous ratings). Suspicious vendors might apply different strategies, for instance non-anonymous or anonymous ratings, to manipulate trust value. In other words, suspicious vendors can be classified into different groups. In our case, there are two groups SV3 and SV4.

5.1.3 The R-Rep Trust Model

A restrictive reputation model, R-Rep, is proposed to increase the robustness of a TMS. According to the assumptions introduced in section 5.1.1, R-Rep assigns different weights to different ratings according to the monetary value of a transaction and relative rating-provision frequency of a customer. Rating-provisioning frequency refers to the frequency a customer provides ratings. R-Rep is formalized in formula 5.1, where r_i^s represents a rating of a service s, $cust(i)$ denotes a function which maps a rating to its corresponding customer, and $prod(i)$ maps a rating to the corresponding product. $RP_{prod(i)}$ stands for the relative price of product $prod(i)$ among all transactions relating to target vendors. $RF_{cust(i)}$ stands for the relative frequency of customer $s(i)$'s rating provisioning. The basic idea of this formula is to give more weight to the rating which is given on more expensive products than cheap ones. Less weight is given to the rating which is provided by the customer who has already provided many ratings. There are many functions for modeling RP and RF. We assume that some of the functions work better than the others. Here we provide some possibilities.

The function for $RP_{prod(i)}$, which is formalized in formula 5.2, is implemented as a variation of sigmoid function. β controls the height of the "S" shape of sigmoid function and λ/β is the upper bound of the function for $RP_{prod(i)}$. The function reaches the lower bound when the variable $p_{prod(i)}$ is equal to 0. The sum divided by $|N_{tran}|$,

where $|N_{tran}|$ represents the number of transactions of interest, denotes the average price for products. $P_{prod(j)}$ is the price of a product $prod(j)$. The "S" shape of sigmoid function represent the idea that, the more expensive the transaction, the more weight the rating of the transaction has. For $RF_{cust(i)}$, which is formalized in formula 5.3, it is implemented as a parabola function, where κ is used to control the shape of the curve. The idea is to give less weight to the rating which is provided by the customer who has already provided many ratings. In a B2C system like Taobao, it is very suspicious for a customer to provide many ratings per day. $f_{cust(i)}$ is the frequency of purchase for a customer $cust(i)$. Following the similar idea as above, formula 5.4 and 5.5 implement $RF_{cust(i)}$ as an exponential and a squared exponential function respectively, where α is the parameter to control the shape of the function.

$$TM(s) = \sum_{i=1}^{R(s)} r_i^s * \frac{RP_{prod(i)}}{RF_{cust(i)}} \tag{5.1}$$

$$RP_{prod(i)} = \frac{\lambda}{\exp\left(\frac{\sum_{j=1}^{|N_{tran}|} p_{prod(j)}}{|N_{tran}|} - p_{prod(i)}\right) + \beta} \tag{5.2}$$

$$RF1_{cust(i)} = \kappa * (f_{cust(i)} - \frac{\sum_{j=1}^{|N_{tran}|} f_{prod(j)}}{|N_{tran}|})^2 + 1 \tag{5.3}$$

$$RF2_{cust(i)} = \exp\left(\alpha * (f_{cust(i)} - \frac{\sum_{j=1}^{|N_{tran}|} f_{prod(j)}}{|N_{tran}|})\right) + 1 \tag{5.4}$$

$$RF3_{cust(i)} = \sqrt{\exp\left(\alpha * (f_{cust(i)} - \frac{\sum_{j=1}^{|N_{tran}|} f_{prod(j)}}{|N_{tran}|})\right) + 1} \tag{5.5}$$

R-Rep is a lightweight model, since the complexity of computation is similar to that of the trust model used by Taobao. Note that, the sums in formulae 5.2, 5.3, 5.4 and 5.5 can be calculated in advance, so they are regarded as constants in formula 5.1. $RP_{prod(i)}$ and $RF_{cust(i)}$ can be computed with complexity of O(1), hence, the time complexity of R-Rep is O(n).

5.1.4 Experimental Results

In this section, two approaches are introduced to compare different trust models with respect to robustness. The two approaches are generally applicable for any trust model without any concrete restriction. Afterwards, the results of the comparison and the corresponding statistics are shown.

5.1.4.1 Comparing Different Models

We propose two approaches Ranking Comparison Index (**RCI**) and Benefit Variation Ratio (**BVR**), to compare trust models regarding robustness given a set of suspicious vendors. Both approaches share a criterion that, if the benefit which a set of "bad" vendors achieve under one trust model, is less than the other, the first model is more robust against manipulation than the second model. The advantage of the both approaches is that the comparison procedure is independent of trust models per se.

For RCI, the ranking of a vendor is positively correlated to its potential benefit. The higher the ranking, the more benefit a suspicious vendor will achieve. Therefore, RCI measures the ranking variation of two trust models and compares them. The Ranking Comparison Index (RCI) is defined in formula 5.6, where $RK_i^{(x)}$ represents the ranking of a vendor i calculated under model x, and SV denotes a set of suspicious vendors. If trust model 1 is better than 2 in terms of resisting manipulation, RCI should be larger than zero and vice versa. Given the assumption that the ranking of a vendor can be modeled as an increasing function of its trust value, the validity of RCI is strongly related to the two trust models. If the impact on resisting manipulative behavior for one model is much larger than the other, the distance of rankings of the same vendor in the two models will be very large. The set of suspicious vendors influences the validity as well. In the extreme case, using a set of innocent vendors can generate the opposite outcome.

$$RCI = \sum_{i=1}^{|SV|} \frac{RK_i^{(1)} - RK_i^{(2)}}{\min(RK_i^{(1)}, RK_i^{(2)})} \tag{5.6}$$

For BVR, the distribution of trust values is taken into consideration, where benefit is proportional to trust value. When plotting the probability density function $f(x)$ of trust, it shows that most suspicious vendors have lower trust value, and very few suspicious vendors have an extremely high value. The value of $f(x)$ is inversely proportional to x. Hence, we consider $1/f(x)$ as the measure of benefit which is proportional to its corresponding trust value x. The probability density function (PDF) $f(x)$ is fitted by EasyFit[1]. Comparison of the fitting results shows two PDFs, the Dagum distribution and Weibull distribution, which are comparable to the PDF generated from the data. Their mathematical expressions are listed as formula 5.7 and 5.8, where α, β, γ and k are the parameters.

$$f_{Dagum}(x) = \frac{\alpha k(\frac{x-\gamma}{\beta})^{\alpha k-1}}{\beta(1+(\frac{x-\gamma}{\beta})^{\alpha})^{k+1}} \tag{5.7}$$

$$f_{Weibull}(x) = \frac{\alpha}{\beta}(\frac{x-\gamma}{\beta})^{\alpha-1} exp(-(\frac{x-\gamma}{\beta})^{\alpha}) \tag{5.8}$$

Formulae 5.9, 5.10, 5.11 and 5.12 define the main calculation steps, where $BV(x)$ stands for the benefit variation of suspicious vendors SV under model x; $TB(x)$ stands for the total quantity of benefit obtained by population under model x, and N denotes

[1]It is a distribution fitting software. www.mathwave.com

the number of all the vendors; BVI stands for benefit variation index, which evaluates the relative benefit variation of trust model 2, in the frame of reference of trust model 1; BVR stands for benefit variation ratio, which evaluates to what degree the benefit obtained by the suspicious vendors is reduced using trust model 2 instead of trust model 1. $TB(1)/TB(2)$ is the key factor, which transforms the benefit variation under trust model 2 into that under trust model 1. This factor implies that, the quantities TB measures under different models are the same, yet the scale is different from model to model. It is possible to transform benefit variation under trust model a to trust model b using the factor $TB(b)/TB(a)$. Similar to the argument for RCI, the validity of BVR is strongly related to the two trust models and the set of suspicious vendors.

$$BV(x) = \sum_{i=1}^{|SV|} \frac{1}{f_x(TM_i^{(x)})} \tag{5.9}$$

$$TB(x) = \sum_{i=1}^{N} \frac{1}{f_x(TM_i^{(x)})} \tag{5.10}$$

$$BVI = BV(2) * \frac{TB(1)}{TB(2)} \tag{5.11}$$

$$BVR = \frac{BV(1) - BVI}{BV(1)} \tag{5.12}$$

5.1.4.2 Results of Model Comparisons

In this section the results of comparing the trust models will be discussed. We will restrict ourselves to two models, namely Taobao's model and Beta Reputation System (82), being compared to our models. Here is a short description of the trust models involved in the experiment.

a) Taobao's trust Model (**TB**). This is the baseline model.

b) Beta Reputation System (**BRS**). The trust value is calculated by formula 5.13, where α stands for the number of positive ratings plus 1, and β denotes the number of negative ratings plus 1. Neutral ratings are ignored.

$$TM_{bs} = \frac{\alpha}{\alpha + \beta} \tag{5.13}$$

c) **RT-FP-P**. RT-FP-P encodes a version of R-Rep, in which both the frequency of rating provisioning and the price of the product are considered. As formula 5.3 shows, the frequency factor is modeled as a Parabola function. In addition, λ is set to 2, β to 1, and κ to 1. All the following models use the same value of parameters.

d) **RT-P-P**. In this model, only the factor price is considered.

e) **RT-F-P**. In this model, only the factor frequency is considered.

Name	SV3	SV3+SV4	SV3'	SV3'+SV4'
BRS	-31	20	-26	-24
RT-FP-P	291	495	269	349
RT-FP-E	293	513	271	350
RT-FP-RE	292	474	271	349
RT-F-P	26	36	20	27
RT-P-P	287	437	266	345

Table 5.4: Robustness analysis by RCI

f) **RT-FP-E**. Formula 5.4 is used for trust calculation, and parameter α is set to 1.

g) **RT-FP-RE**. Formula 5.5 is used for trust calculation, and parameter α is set to 1.

We assume that the vendors who intend to promote their trust value are supposed to have poor status of trust in the first place. We fit the correlation between trust value and daily volume of sales regarding vendors in Fig 5.4 using linear regression. Based on the third assumption (profit maximization), we are able to reduce SV3 and SV4 into two even smaller sets SV3' and SV4' by filtering out the vendors who stay under the regression line in Fig. 5.4. The control experiment compares the robustness against manipulative behavior between Taobao's model and the other six trust models using both RCI and BVR.

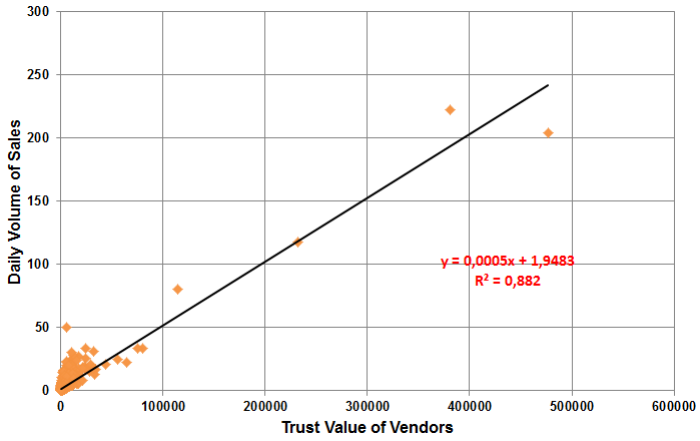

Figure 5.4: Trust value vs. daily volume of sales regarding vendors

The result for RCI is shown in Table 5.4, where an entry represents the RCI of the baseline model (TB) to the corresponding trust model. BS performs even worse than TB for sets SV3, SV3' and SV3'+SV4'. Due to the inferior performance, BS is not involved in further discussion.

RT-FP-P, RT-FP-E and RT-FP-RE perform almost equally well. RT-P-P performs slightly worse than the three models. RT-F-P is the worst model. The results indicate

Name	SV3	SV3+SV4	SV3'	SV3'+SV4'
RT-FP-P	10.625	11.303	4.326	4.722
RT-FP-E	11.635	12.871	4.650	5.125
RT-FP-RE	26.540	72.328	24.222	29.777
RT-F-P	8.832	9.411	4.014	4.388
RT-P-P	9.238	10.010	4.082	4.485

Table 5.5: Robustness analysis result of BVI ($\times 10^5$)

that the exponential function performs best in all the models, but there is little difference among the models which consider both price and frequency factors. Furthermore, price plays a more important role in maintaining robustness than frequency. The main reason that price is a discriminatory fact is that vendors can control the intensity of manipulative behavior in order to avoid being captured by Taobao's detection mechanism. Thirdly, all the values of RCI are positive, and it shows that all the five models perform much better than TB. The RCI value considering the suspicious set SV3+SV4 is larger than that of SV3 among R-Rep and its variations, and so is SV3'+SV4' to SV3'. We can easily calculate the RCI value of SV4 by subtracting that of SV3+SV4 by SV3. Therefore, all the five models work well on both SV3 and SV4. In addition, the RCI considering SV3 is larger than that of SV3' among all the models, and so is SV3+SV4 to SV3'+SV4'. The difference implies the invalidity of the hypothesis that the vendors who intend to promote their trust value initially have poor status.

The results for BVI and BVR are shown in Table 5.5 and 5.6 respectively. In Table 5.5, the number in the entry is the value of BVI divided by 10^5. It is not possible to consider BS due to the characteristic of BS's probability density function. There are two facts shown in the table. The first one is that the BVI considering the suspicious set SV3+SV4 is larger than that of SV3 among all the models, and so is SV3'+SV4' to SV3'. The second one is that BVI considering suspicious set SV3 is larger than that of SV3' among all the models, and so is SV3+SV4 to SV3'+SV4'. The two facts are exactly the same as in RCI, therefore the two comparison approaches RCI and BVI show inherent consistency regarding robustness evaluation. BVR is shown in Table 5.6, in which the BVR indicates that, comparing to the Taobao's trust model, to what degree the benefit has been reduced by using the corresponding model. RT-FP-P, RT-FP-E and RT-FP-RE perform almost equally well. RT-P-P does slightly worse. RT-F-P is the worst model. This observation confirms the previous conclusion that, RCI and BVR share inherent consistency regarding robustness evaluation.

5.1.4.3 Statistical Results

After the set of suspicious vendors and customers have been identified, we explore the statistical characteristics of suspicious vendors and customers separately. In this section, we focus on the information about the distribution of life span and trust grade. Life span refers to how long a vendor or a customer stays in the system after initial registration. Taobao discretizes trust value as trust grades, which is illustrated by Fig.

Name	SV3	SV3+SV4	SV3'	SV3'+SV4'
RT-FP-P	64.50%	97.07%	83.04%	92.62%
RT-FP-E	70.48%	97.56%	84.26%	93.14%
RT-FP-RE	69.13%	97.41%	84.00%	92.99%
RT-F-P	11.31%	81.28%	5.05%	53.45%
RT-P-P	61.12%	96.67%	81.77%	91.99%

Table 5.6: Robustness analysis result of BVR

5.5. The first grade ranges from 4 to 10, the second grade ranges from 11 to 40, and so on. The trust model we use here is RT-FP-E, which has demonstrated to be the best model in the previous section.

Figure 5.5: Trust grade in Taobao

a) Selecting model parameters. We choose the best combination of parameters for RT-FP-E. Since the set of suspicious vendors is given, we are able to select the best combination by calculating RCI. The evaluation results are listed in Table 5.7. The results show that the best result comes from the last row which is written in bold. However, the difference between different combinations is not very large. We argue that, R-Rep is not a parameter-dependent trust model. This is one of the significant properties for a good trust model, since TMS administrators don't need to worry about how to choose the right parameters. For generating the statistical analysis results, only the best parameters are used.

b) Distribution of life span for vendors. In Fig. 5.6, the light color bar stands for the suspicious vendors who are identified using only anonymous ratings (SV4); the dark

λ	β	α	**RCI**
2	1	1	190.23
2	1	0.5	188.46
2	1	2	192.08
1.5	0.5	1	190.47
1.5	0.5	0.5	188.45
1.5	0.5	2	192.00
3	2	1	190.49
3	2	0.5	188.44
3	**2**	**2**	**192.09**

Table 5.7: RCI results with different parameter settings

color bar denotes the ones who are identified using only non-anonymous ratings (SV3); the sum of two bars, therefore, corresponds to the number considering both SV3 and SV4. The distribution of SV3+SV4's life span is close to an exponential distribution; for SV3, without considering year eight, it can be approximated by an exponential distribution. The shorter the life span, the more likely a vendor behaves dishonestly in a non-anonymous manner. In addition, Fig. 5.7 indicates the distribution of the 1081 vendors' life spans. It is obvious that the life-span distribution of suspicious vendors is different from the total vendor population. As the figures demonstrate, most of the suspicious vendors have less than or equal to one year's life span. The trend of misbehavior expansion is obvious. The distribution and the corresponding analysis show evidence for lack of efficient detection mechanisms in the Taobao system.

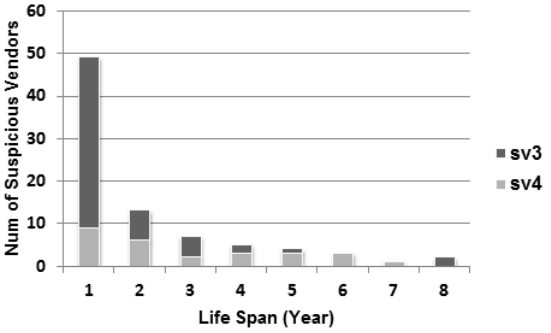

Figure 5.6: Distribution of life span for suspicious vendors

c) Distribution of life span for customers. Following an analogous idea, the statistical characteristics of suspicious customers with respect to life span is plotted in Fig. 5.8, and that of the whole population indicated in Fig. 5.9. The two figures demonstrate that, both distributions can be approximated by exponential distribution. The Fig 5.8 indicates that, the shorter the life span, the more likely a customer behaves

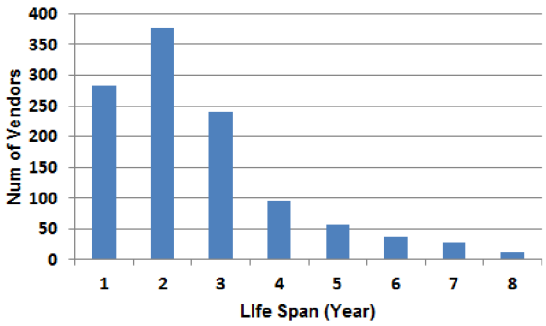

Figure 5.7: Distribution of life span for all vendors

dishonestly. In addition, most of suspicious customers have less than one-year life span.

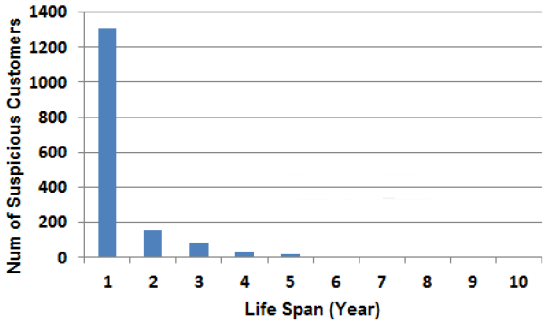

Figure 5.8: Distribution of life span for suspicious customers

d) Distribution of trust grade for vendors. Figs. 5.10 and 5.11 have a similar shape. The peak region covers from grade six to nine. The mode of the distribution for all vendors is eight; the mode of the distribution for suspicious vendors is seven. There are a few vendors outside of the peak area. The similarity of distributions hints at a significant phenomenon that the vendors, no matter which trust grade they have, behave dishonestly. When other people behave dishonestly, in order to occupy an advanced sale position, a vendor has to behave dishonestly.

e) Distribution of trust grade for customers. Figs. 5.12 and 5.13 show the distribution of the trust grades for suspicious customers and for all customers respectively. We observe two phenomena from the two figures. First, most of the suspicious customers have a trust grade between five and twelve; the trust grade of all the customers is mainly between one and six. The suspicious customers intend to maximize their profit such that the amount of transaction history is supposed to be relatively higher than the honest customers. Due to the design of Taobao's trust model for a customer, more positive and less negative ratings provided by vendors imply higher trust value. The second phenomenon is that there is a subset of suspicious customers whose

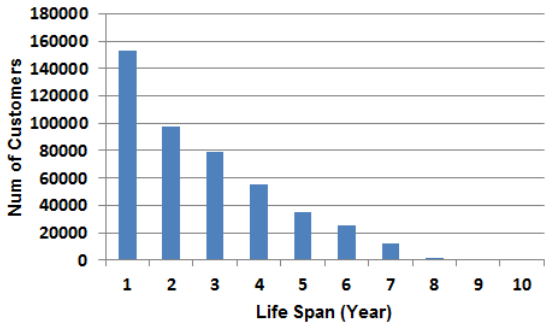

Figure 5.9: Distribution of life span for all customers

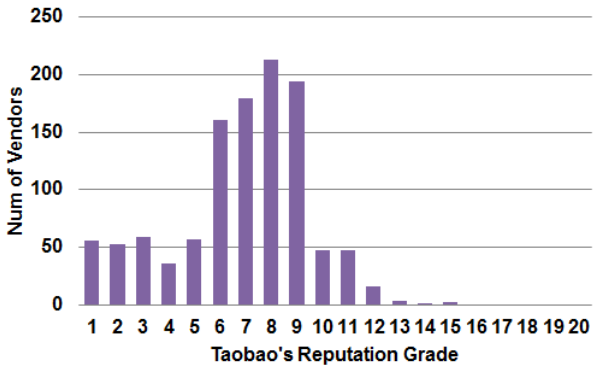

Figure 5.10: Distribution of trust grade for all vendors

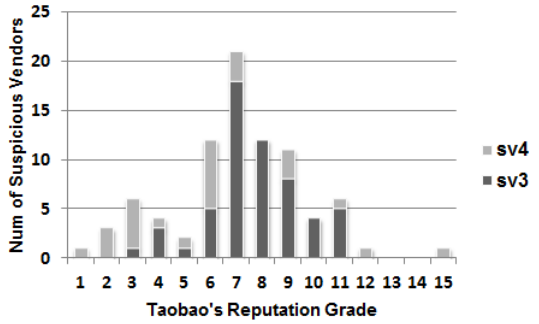

Figure 5.11: Distribution of trust grade for suspicious vendors

trust grade is two that is shown in Fig. 5.12. This phenomenon indicates that these suspicious customers misbehave not so obviously that they don't have higher trust grade. In order to maximize their profit, the attacker probably operates many identities. This type of attack is called Sybil attack (89). A TMS is vulnerable to Sybil attacks since in a computer network it is difficult to find the correspondence between one person and his/her identities. Another explanation could be that there are some dishonest customers who just begin to behave dishonestly, and with the accumulation of transactions, their trust value will increase eventually.

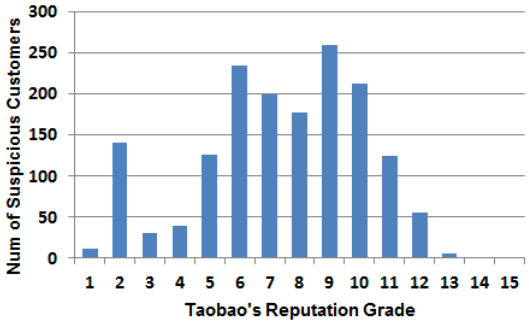

Figure 5.12: Distribution of trust grade for suspicious customers

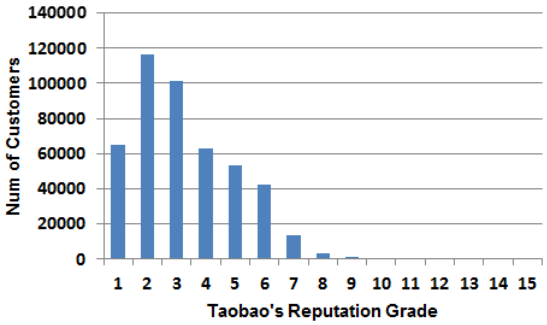

Figure 5.13: Distribution of trust grade for all customers

f) Results of temporal analysis. Due to the dynamics of a TMS, the state of the base system is always changing. In order to capture the temporal impact of R-Rep on the dataset, firstly we record RCI by month; this is shown in Fig 5.14. We have a six-month rating history. The average RCI for 6 months with respect to SV3 is around 200, 350 for SV4, and 550 for SV3+SV4. The average RCI indicates that R-Rep performs better than the model used by Taobao during the whole six months. Hence, R-Rep is able to suppress the rankings of a set of suspicious vendors over time. However, for a single suspicious vendor it is not clear, whether he will also be restricted by R-Rep. It might be the case that for some of the suspicious vendors R-Rep works perfect but does not work for the other suspicious ones and in total the positive effect of R-Rep still dominates. We choose a set of random samples

to investigate the effectiveness of R-Rep at the individual level. In Fig. 5.15 three typical cases are listed. There are three vendors whose trust grades are 3, 9 and 15, respectively. Two lines correspond to one vendor. The line with open points represents the ranking with respect to the Taobao algorithm; the line with filled points represents the ranking with respect to R-Rep. For the vendor with trust grade 15 the resisting effect against the manipulation of a single suspicious vendor is quite distinct. The average ranking of this vendor drops from 3 to around 700. For the vendor with trust grade nine, in the first month, the ranking for Taobao is 1038 and for R-Rep is 1036. In the second month, 555 for Taobao and 426 for R-Rep. This shows that at certain points R-Rep performs worse than the Taobao algorithm. In the first two months, there are not many transactions for the vendor. When there is not enough rating history data, R-Rep might not perform better. For the vendor with trust grade three, the level of data insufficiency is even worse and for the first four months, R-Rep performs almost as same as the Taobao algorithm. The final conclusion is that given sufficient quantity of history, the resisting impact on manipulative behavior can be imposed on the individual level regardless of which grade they are in.

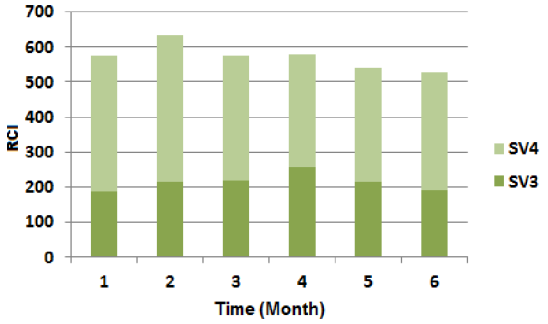

Figure 5.14: Time series of RCI

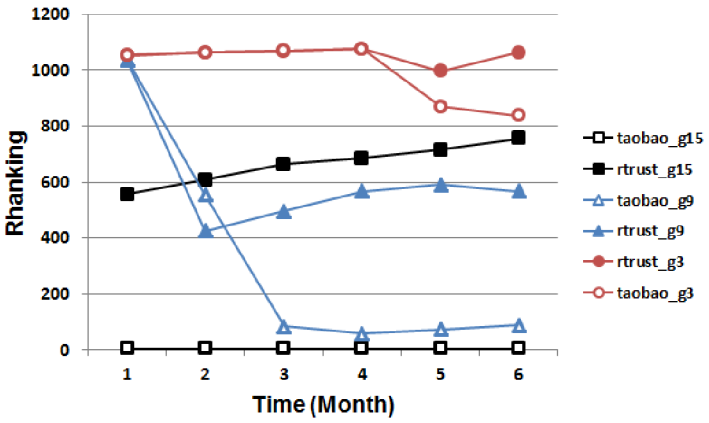

Figure 5.15: Time series analysis

5.2 A Case Study on a Travel-Related Service

In this section, we choose TripAdvisor[1] as an example for a travel-related service. TripAdvisor (TA) contains, among other things, a large number of reviews about hotels. We consider that manipulation can be attempted at three levels: at the rating level, the rating provider level and the service level. A level refers to an aspect of information from which manipulative behavior is observed. Features regarding manipulation identification will be proposed for all the three levels. We propose two different approaches for identifying two types of manipulative behavior, promoting and demoting (slandering). The first manipulation detection approach is an advanced clustering approach, suspicion degree meter (SDM), which is used to rank suspects with respect to their (potential) manipulative behavior. SDM assigns two real numbers, the suspicion index for promoting (SIP) and the suspicion index for demoting (SID), to each object at the three levels. The second approach uses supervised learning for automatic detection of suspicious behavior in a travel-related service. Three trust models are then proposed. The first one is time-window-based, the second one is time-decay-based, and the third one is suspicion-index-based. Finally the two detection approaches and the three proposed trust models are evaluated. Annotations for supervised learning are generated. Considering both unsupervised learning (SDM) and supervised learning, the statistical characteristics of suspicious sub-populations and innocent sub-populations are compared. Using the ranking comparison index (RCI) introduced in section 5.1.4, all three types of proposed trust models are compared in terms of their robustness against manipulative behavior.

5.2.1 Travel-Related Services and TripAdvisor.com

We define a travel-related service as a service which is provided for potential travellers, searching for hotels, restaurants, flights, etc. In recent years many review websites, e.g. TripAdvisor, Yelp[2], and Booking.com[3], have gained success by integrating trust into their websites. A reason for the success of these websites is that, trust acts as a social catalyst which aids consumers to decrease the degree of uncertainty and the risk of decision making. For instance, via a review website, a user can assess the trust in hotels which are in the vicinity of their travel location before making a booking.

TripAdvisor.com is a travel website that assists customers in gathering travel information, posting reviews and opinions of travel-related content and engaging in interactive travel forums. We did not reuse the dataset in the previous studies (66), because we find that the representativeness of a feature for identifying manipulation depends on the characters of a dataset. For instance, a feature called "positive singleton" (66), which refers to a positive review which is the only review posted by a reviewer, is a key feature for detecting manipulation on TA. The dataset used in the previous study (66) covers all 741 hotels and their corresponding reviews throughout Ireland. However, in our dataset, the proportion of positive-singleton reviews is as small as 2.47% compared

[1] www.tripadvisor.com

[2] www.yelp.com

[3] www.booking.com

Statistic	NYC	HNO
Duration	1999.1 - 2011.6	2000.6 - 2011.10
Hotels	420	344
Reviewers	110,128	14,681
Reviews	167,909	17,271
Contributions	770,644	115,160
Singletons	5,446 (3.24%)	426 (2.47%)

Table 5.8: Basic statistics of TA's datasets

to 20% in the dataset which is used in the previous study (66). The reason might be that manipulative behavior observed in different geographical regions is variable. Therefore, one feature which is representative regarding detecting manipulation in one dataset, might not be so useful in our dataset. Instead of using an existing dataset, we collect two segments of data instead of one to act as a control against over-training. New York City (NYC) is considered as one of the most ideal cities for travelling, and there is a large number of hotels. We believe that it is more likely to find manipulative behavior related to NYC than for other regions due to the keen competition among hotels. Hanoi (HNO) is chosen as the second candidate because of a manipulation report[1], where it shows a request of writing fraudulent ratings for some hotel in Hanoi. The basic statistics of the two datasets are listed in Table 5.8. Note that in the case of TA's datasets, the three categories: rating, rating provider and service, are referred to as "review", "reviewer" and "hotel" respectively.

5.2.2 Feature Identification

As was said before, there are three categories that are involved in manipulative behavior: the review, the reviewer and the hotel. A service provider (e.g. a hotelier) may "hire" reviewers either to give positive reviews to their hotel, or to give negative ones for their competitors. We call the former case promoting manipulation and the latter case demoting (slandering) manipulation. In the following, we introduce the features we use for classification and clustering.

a) Advanced positive singleton (**AdvPosSing**), formalized by formula 5.14, is the improved version of positive singleton (66). This feature is defined at the review level for hoteliers to promote manipulation. In Wu's definition of a positive singleton (PS) (66), a positive review is one assigning four or five stars, and a review with fewer than four stars is negative. This definition is arbitrary and could be inaccurate. For instance, a latest four-star rating r_i^t should be considered as negative if the previous ratings are all five-star. Therefore, instead of using fixed values, we estimate the difference between a latest rating and the current trust value of the hotel, i.e. the average score at the moment when the rating is created. If a latest rating is a

[1]http://tripadvisorwatch.wordpress.com/2011/06/10/earn-money-writing-tripadvisor-reviews

PS and the distance is larger than a threshold TH_p, we consider the rating as positive, negative otherwise. In the experiment, we empirically set the threshold to one. Following the same idea, an advanced negative singleton (**AdvNegSing**) is specified for demoting manipulation likewise.

$$AdvPositiveSingleton(r_i^t) = \begin{cases} 1 & \text{if } r_i^t \text{ is PS and } (r_i^t - TV^t) > TH_p \\ 0 & \text{Otherwise} \end{cases} \tag{5.14}$$

b) Time interval between posted date and check-in date (**TimePostedCheckin**) refers to the duration from the date the reviewer stayed in this hotel to the date a review is posted. This feature is defined at the review level for both promoting and demoting manipulation.

c) Time interval between consecutive contributions (**Time_Consec_Contributions**). Contributions of a reviewer are ordered by the time a review is posted. The time interval between two consecutive reviews can be regarded as a random variable. This feature contains two parameters, mean (**TimeConsecContributions_MEAN**) and variance (**TimeConsecContributions_VAR**) of the time interval variable. The two parameters can be considered as two features for machine learning. They are defined at the review level for both promoting and demoting manipulation.

d) Rating preference (**RatingPreference**), formalized by formula 5.15, is an indicator for describing a reviewer's attitude towards review provisioning. In formula 5.15, $SUBR()$ denotes a function whose inputs are the overall review score and the index of the sub score, and the output is the value of the corresponding sub score. In TA a review (rating) is composed of an overall score, a set of sub scores regarding hotel features and a textual content. Both the overall score and the sub scores are encoded as integers from one to five. In this section a review is equivalent to a rating. When writing a review, a reviewer does not only give an overall score r_i^t, but also sub scores $SUBR(r_i^t, k)$ for value, rooms, location, cleanliness, service, etc. This feature is defined at the review level for both promoting and demoting manipulation.

$$RatingPreference(r_i^t) = r_i^t - \frac{\sum_{k=1}^{N} SUBR(r_i^t, k)}{N} \tag{5.15}$$

e) Turning day (**TurningDay**), demonstrated by Fig. 5.16, indicates the maximal trust variation of a hotel. Each point in the figure represents a trust evaluation with a certain time stamp. The cycle of evaluation is one month. We develop a simple algorithm to identify the intervals which have the steepest positive and negative slopes TurningDay_MAX and TurningDay_MIN. These are the times where the trust varies most rapidly. We specify TurningDay_MAX as a feature for promoting manipulation, and TurningDay_MIN as a feature for demoting manipulation. Furthermore, the logical relationships between hotel, reviewer and review are also taken into consideration, since the variation comes from reviews and the reviewers who provide them. TurningDay is defined at all levels and for both promoting and demoting manipulation.

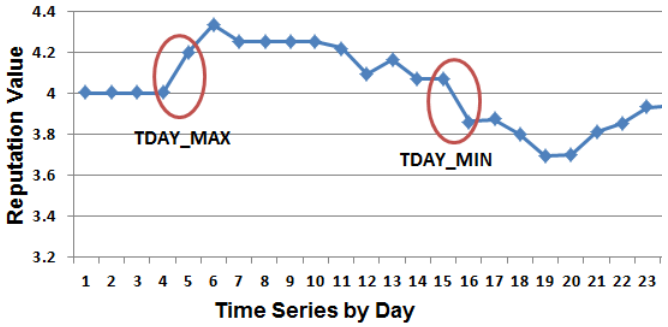

Figure 5.16: An illustration for the feature TurningDay

f) Inactive duration (**InactiveDuration**) refers to the duration between the last post and the time when data is collected. This feature is defined at the reviewer level for both promoting and demoting manipulation.

g) Contribution statistics (**ContributionStatistics**) contains the number of contributions (**ContributionNum**), the arithmetic mean (**Contribution_MEAN**) and the variance (**Contribution_VAR**) of contributions provided by a reviewer. All three sub-features are defined at the reviewer level for both promoting and demoting manipulation.

h) Consistency of ratings (**ConsistencyRating**), contains the variance of mode (**VAR_MODE**) and the variance of mean (**VAR_MEAN**) with respect to different types of reviews for a hotel. The idea behind this feature is to measure to what degree different types of reviews are consistent with each other, and this feature is defined at the hotel level for both promoting and demoting manipulation. First, we categorize reviews of a hotel by the categories of reviewer, such as "business","couples" etc. Then we calculate mode and mean of these variables respectively. Finally, the variance of mode and mean for each of the different categories of reviews are calculated. Formula 5.16 shows the calculation of VAR_MEAN. R_j denotes the set of reviews for a hotel. $SUBS()$ is a function which returns the subset of reviews in terms of type k. $MEAN$ and VAR are defined to evaluate mean and variance respectively.

$$VAR\_MEAN(R_j) = VAR(MEAN(SUBS(R_j, k))) \tag{5.16}$$

i) Average number of reviews per month (**AveNumPerMonth**) refers to the mean number of reviews posted for a specific hotel in one month. This feature is defined at the hotel level for both promoting and demoting cases.

j) Proportion of advanced positive singleton (**PropAdvPosSing**) refers to the proportion of AdvPosSing and is defined at the hotel level for promoting manipulation. The feature is adopted from Wu's definition of "proportion of positive singletons" (66). We only replace positive singleton by AdvPosSing. Parallel to this, proportion

of advanced negative singleton (**PropAdvNegSing**) is defined in an analogous way for demoting manipulation.

k) Reactive advanced positive singletons (**ReactiveAdvPosSing**) is adopted from reactive positive singletons (66). In order to recover from negative reviews, the management may react by posting some positive reviews. The strength of evidence can be quantified as $T - t_i/T$ where T is the length of the entire time period, and t_i is the reaction time associated with a positive review i. This feature is formalized by formula 5.17, where T_h is a normalization factor for hotel h which is the elapsed time between the 1^{st} and the n^{th} reaction positive singletons. This feature is defined at the hotel level for promoting manipulation.

$$ReactiveAdvPositiveSingleton(h) = \frac{1}{|T_h|}(1 - \prod_{i=1}^{n}(1 - \frac{T - t_i}{T})) \tag{5.17}$$

l) Truncated positive rating (**TruncPosRating**) is adapted from the definition of truncated rating (66). The idea is to remove a portion of the most positive reviews for a hotel and recalculate the average to see if it deviates much from the original value. In our evaluation, 20% most positive reviews are removed. TruncPosRating is formalized by formula 5.18, where R_h^{tr} is the truncated rating set. This feature is defined at the hotel level for promoting manipulation. Analogous to this, the Truncated Negative Rating (**TruncNegRating**) is defined for demoting manipulation.

$$TruncPositiveRating(h) = \frac{1}{|R_h|}\sum_{r \in R_h} r - \frac{1}{|R_h^{tr}|}\sum_{r \in R_h^{tr}} r \tag{5.18}$$

m) Rating mean (**Rating_MEAN**) refers to the mean of ratings for a hotel. This feature is defined at the hotel level for both promoting and demoting manipulation.

n) Rating variance (**Rating_VAR**) refers to the variance of ratings for a hotel. This feature is defined at the hotel level for both promoting and demoting manipulation.

o) Ratio of room number to review number (**RatioRoomReview**) refers to the ratio of the number of rooms in a given hotel to the number of reviews for the hotel. The intuition is that it is suspicious for a hotel who owns only few rooms to have a large number of reviews. This feature is defined at the hotel level for promoting behaviour.

p) Hotel reviews contradiction degree (**ContradictionDegree**), formalized by formula 5.19, refers to the maximum variance of sub-ratings for a hotel. There are N sub-ratings for each of the categories such as, value, rooms, location, cleanliness, service, etc. MAX is a function to find the maximum variance. This feature is defined at the hotel level for both promoting and demoting cases.

$$ContradictionDegree(h) = MAX(\{VAR(SUBS(r_i^h, k), \text{where } i = 1...N\}) \tag{5.19}$$

q) Textual features (**UniBigram**) refers to the textual features extracted from the review content. Like in (3, 67, 90), we use unigrams and bigrams representing the review text by the amount of its words and consecutive word pairs.

5.2.3 Suspicion Degree Meter (SDM)

We propose an advanced clustering approach, suspicion degree meter (SDM), to identify manipulative behavior (68). Manipulative behavior here refers to the operation of injecting fraudulent reviews. Either a service provider (e.g., a hotelier or a restaurateur) "hires" reviewers to post positive reviews for their own service, or negative ones for competitors. In the case of promoting behavior, both high- and low-ranking hotels have a potential motivation. High-ranking hotels intend to keep their superior position in the ranking list, and low-ranking hotels simply want to promote their rank. In the case of demoting behavior, usually the victim has a relatively high ranking. For the design of SDM, we consider all three levels, review level, reviewer level and hotel level. For each level, SDM uses two numbers, the suspicion index for promoting/demoting (SIP/SID), for representing suspiciousness of an object with respect to the two types of manipulative behavior. The two numbers are calculated by applying a fuzzy c-means clustering algorithm (91). The basic idea of manipulation identification in each layer is as follows. We assume that the numerical characteristic of a suspicious object is distinguishable from that of a normal object. Hence, given the assumptions listed below, it is possible to identify suspicious clusters by applying a clustering algorithm. There are four basic assumptions for our dataset:

a) (**AS1**) If there is demoting manipulative behavior, usually inconsistencies between reviews of a hotel can be observed in a short time interval.

b) (**AS2**) If there is manipulative behavior, a disturbance in temporal reputation evaluation can be observed.

c) (**AS3**) The more singletons, the more suspicious. A singleton refers to the review provided by a reviewer, who only provides one review.

d) (**AS4**) If a reviewer tries to promote a hotel, their rating attitude tends to be optimistic; if a reviewer tries to demote, their rating attitude tends to be pessimistic.

The four assumptions are the root of trust, and the SDM is built upon their validity. AS1 states that inconsistency between reviews of a hotel appears when demoting manipulation takes effect. The motivation of demoting manipulation is to decline the trust in a hotel which normally has a high trust value. Therefore we observe both positive reviews and negative reviews. AS2 indicates that if there is manipulative behavior, an abnormal variation can be observed in the time series of trust evaluation. AS3 states that the feature singleton is a significant one. AS4 indicates that the polarity of a reviewer's rating attitude is reflected by considering the difference between the overall rating score and the corresponding sub scores provided by a reviewer. When giving a review, a reviewer not only gives an overall score but also scores for the dimensions such as service, value, sleep quality, cleanliness, location, rooms, etc.

Given a context of trust, it is possible to capture basic patterns of manipulation, which are refined as basic assumptions or characteristics. All of the assumptions are considered for manipulation identification, but different ones are assigned different weights in terms of the specific context. Based on the assumptions, a robust reputation model is generated, and different learning algorithms can be applied. In TripAdvisor, we are following the four assumptions above.

When applying the fuzzy c-means algorithm on each level, each object (hotel, reviewer or review) is assigned two numbers in the range 0 to 1, representing SIP and SID respectively. During the unsupervised learning process, the basic assumptions are used to decide which clusters are more suspicious than the others. Moreover, the logical relationship among different levels is utilized to adjust the initial suspicion indices (SIP and SID) which are calculated by fuzzy c-means clustering algorithm. In our case study, the suspicion indices for reviews are used to adjust the suspicion indices for reviewers and hotels. There are two logical relations being taken into consideration. First, a hotel's suspicion index depends on the corresponding reviews. Second, a reviewer's suspicion index depends on the corresponding reviews. Yet, one should not assume that the suspicion index of a hotel depends on the reviewers who provide reviews to the hotel, because a suspicious reviewer may only write fraudulent reviews for some target hotel, but behave "normally" in other cases. Following those lines of thought, we develop two operations to adjust the suspicion index (SI) for reviewers and hotels by considering the SI of reviews. Here the SI represents either PID or SID. Formula 5.20 is used to adjust SI at the reviewer level. $SI_{1\times n}^{RWR}$ represents the old SI vector regarding reviewers, in which there are n reviewers in system; $SI_{1\times n}^{'RWR}$ represents the new SI vector regarding reviewers. $SI_{1\times n}^{RE}$ represents the SI vector regarding reviews. $RR_{m\times n}$ represents a relationship matrix between reviews and reviewers. An entry rr_{ij} is equal to one, if a reviewer j posts a review i. α is the weight given to the old SI vector. A suspicion degree vector derived from reviews is given weight $1-\alpha$. $\otimes$ is an aggregation factor, which maps the SIs of corresponding reviews of a reviewer into one value. For the dataset in TA, it is implemented as a MAX function, which returns the maximal value. We use a MAX function instead of a AVG function which calculates arithmetic average, since a reviewer can only take action on the target and behave normally otherwise. Formula 5.21 is used to adjust SI at the hotel level. $SI_{1\times n}^{HT}$ represents the old SI vector regarding hotels, in which there are p hotels in system. $RH_{m\times p}$ represents a relationship matrix between reviews and hotels. An entry rh_{ij} is equal to one, if there is a review i of a hotel j. In formula 5.21, the aggregation factor is implemented as an AVG function. Since the suspicion degree of a hotel can be measured by the average of suspicion degree of the reviews which are related to the hotel.

$$SI_{1\times n}^{'RWR} = \alpha * SI_{1\times n}^{RWR} + (1-\alpha) * SI_{1\times m}^{RE} \otimes RR_{m\times n} \tag{5.20}$$

$$SI_{1\times p}^{'HT} = \beta * SI_{1\times p}^{HT} + (1-\beta) * SI_{1\times m}^{RE} \otimes RH_{m\times p} \tag{5.21}$$

One of the main tasks is to find suspicious candidates at review, reviewer and hotel levels. Once SIP and SID are calculated via the SDM, we can rank objects by their SIs, and make a final decision based on that. The SID of reviews in New York City

is plotted in Fig. 5.17. The plot looks like the character "L", which means, there is a small number of reviews having much higher SI than the rest of the reviews. We choose the points on the left-hand side as the candidates of suspicious reviews, because these reviews have higher SID value than the others. For the reviewer level, which is illustrated in Fig. 5.18, the data can be fitted by an exponential function. In this case, we simply choose a threshold for SI (e.g. 0.7) or top ranking (e.g. 10%) to identify the most suspicious group. The choice of threshold depends on various factors, such as the characteristic of the dataset, the shape of the plot, expert's opinion, etc. It is up to the analyzer to make the decision which threshold to choose. In the experiment different thresholds are chosen, depending on the level and the type of manipulation.

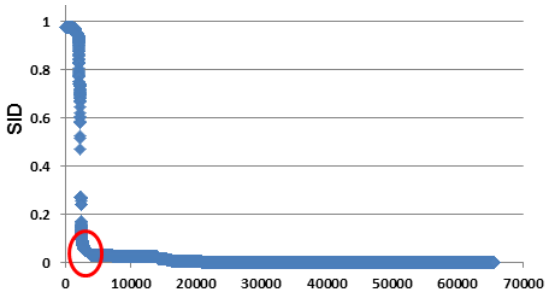

Figure 5.17: SID of reviews in New York City

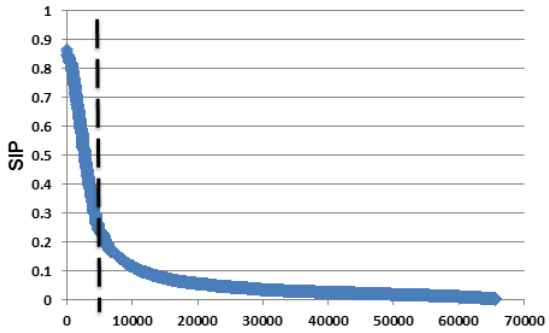

Figure 5.18: SIP of reviewers in New York City

5.2.4 Proposed Trust Models

The basic idea of evaluating trust is to calculate the trust value via aggregating pieces of evidence (reviews) in terms of giving different weights to different types of evidence. In order to build a good trust model for resisting manipulative behavior in TripAdvisor, two factors are considered. The first factor is time. On the one hand, since the trust of a travel-related service varies over time, it is unnecessary to consider reviews several years ago to be as important as more recent reviews. On the other hand, if there are fraudulent reviews in the past, it is a good idea to weaken them by a time window

or time decay factor. We apply both time-window-based and exponential-decay-based approaches. The second factor is the suspicion index (SI), which was introduced in section 5.2.3. Formula 5.22 represents a time-window-based trust model, where ts denotes the time when trust is evaluated; tw stands for a time interval and only reviews within this interval are considered. $TW()$ is a function which returns 1 if the time stamp t of a review is within the interval tw; otherwise it returns 0. The exponential-decay-based trust model is implemented by formula 5.23, where λ is a negative real number.

$$Rep\_TW(ts, tw) = \frac{\sum_{i=1,t<ts}^{N_{tw}} r_i^t * TW(tw, t)}{N_{tw}} \tag{5.22}$$

$$Rep\_EXP(ts) = \frac{\sum_{i=1,t<ts}^{N} r_i^t * e^{(\lambda t)}}{N} \tag{5.23}$$

Formulae 5.24, 5.25 and 5.26 formalize a SI-based trust model designed to resist manipulation, where w_i^{re} represents the weight assigned to a review and w_i^{rwr} to the corresponding reviewer. For resisting promoting manipulation, the weight is implemented by an exponential function, which has one special point (SIP_{min}^{rwr}, 1) at the reviewer level or (SIP_{min}^{re}, 1) at the review level. SIP_{min}^{rwr} denotes the smallest SIP among reviewers. In order to discount the effect of suspicious reviews and reviewers regarding demoting, a linear function is used.

$$Rep\_SIP(ts) = \frac{\sum_{i=1,t<ts}^{N} r_i^t * w_i^{rwr} * w_i^{re}}{N} \tag{5.24}$$

$$w_i^{rwr} = \begin{cases} e^{(\gamma * (SIP_i^{rwr} - SIP_{min}^{rwr}))} & \text{Promoting} \\ \alpha * SIS_i^{rwr} + \beta & \text{Demoting} \end{cases} \tag{5.25}$$

$$w_i^{re} = \begin{cases} e^{(\gamma * (SIP_i^{re} - SIP_{min}^{re}))} & \text{Promoting} \\ \alpha * SIS_i^{re} + \beta & \text{Demoting} \end{cases} \tag{5.26}$$

5.2.5 Experimental Results for Unsupervised Learning

5.2.5.1 Statistical Characteristics of Suspects

Given sets of suspects at the levels of review, reviews and hotel, respectively, we investigate the statistical characteristics of suspects and of the rest of population. As we have mentioned in section 5.2.3, the validity of labeling a review largely depends on the trustworthiness of feedback or review helpfulness (75). Review helpfulness refers to the number of positive feedback from other users to a review. For instance if there are 10 out of some people giving "thumbs-up" in TripAdvisor, then the review helpfulness is equal to 10. Here we investigate the (uncertain) assumption that review helpfulness indicates the trustworthiness of a review. Fig. 5.19 shows the distribution of review helpfulness in New York City. A promoting group is generated by choosing the most suspicious reviews with respect to promoting manipulation, and a demoting group is generated in an analogous fashion. The rest of the reviews are considered the innocent

group. The horizontal axis represents review helpfulness of a review plus one, since in a log-log plot there is no zero on the horizontal axis. More than 99.9% of reviews for each group have zero review helpfulness. However, innocent reviews obtain much more review helpfulness than suspicious reviews in NYC. The reviews which obtain the highest review helpfulness are all considered innocent. It seems that, using review helpfulness to measure trustworthiness of reviews is a valid idea. Yet, the result in Fig. 5.20 gives us a counterexample, which shows the distribution of review helpfulness in Hanoi. Every group shows a power law distribution. More important, for both promoting and demoting groups, there are many reviews with a large review helpfulness score. For instance, there is one review which is suspicious in terms of demoting manipulation. This review has a helpfulness score of 68, which is extremely large. Since there are so many suspicious reviews which have relatively large number of review helpfulness, it is not reasonable to measure trustworthiness of a review using review helpfulness.

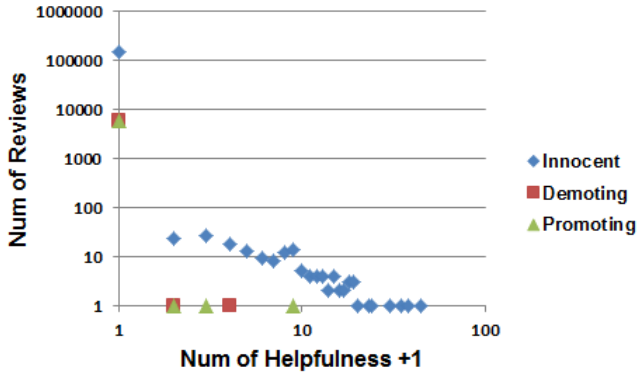

Figure 5.19: Distribution for review helpfulness in New York City

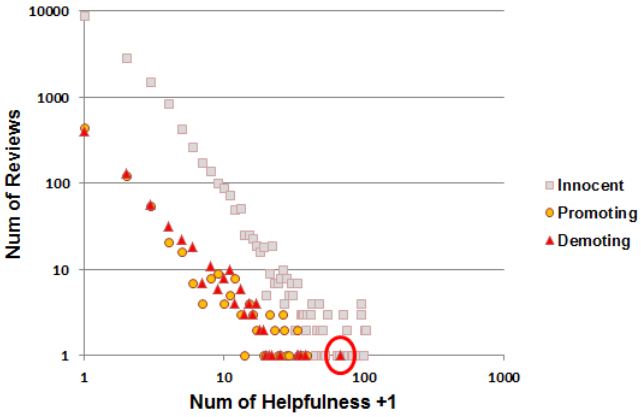

Figure 5.20: Distribution for review helpfulness in Hanoi

Table 5.9 shows the mean and standard deviation of reviewer helpfulness from different subpopulations. Reviewer helpfulness is defined as sum of all the review

Statistic	Promoting	Demoting	Rest	Dataset
Mean	2.35	1.39	0.72	NYC
Standard Deviation	3.90	3.28	2.07	NYC
Mean	1.90	2.26	0.84	HNO
Standard Deviation	8.54	5.21	3.14	HNO

Table 5.9: Statistics for reviewer helpfulness

helpfulness of reviews, which are posted by this reviewer. The table shows that, for both New York City and Hanoi, the mean of reviewer helpfulness for suspicious reviewers is higher than the normal reviewers. The reason may be that most of "bad" reviewers post fraudulent reviews which can mislead a normal user judgment, or most of the reviewer helpfulness is provided by conspirators. It is possible to investigate the exact reason by considering information such as the IP address of every positive and negative feedback to a review.

The distribution of review scores, grouped by travel types is shown in Figs. 5.21 and 5.22. The horizontal axis represents the five travel types, business, couples, family, solo and friends. The vertical axis represents the proportion of the population. Different groups are color coded, red for the promoting group; green for the demoting group; blue represents the innocent group. Figs. 5.21 and 5.22 show that the distributions are almost the same regardless of different groups. The distributions with similar shape imply that, manipulative behavior is not spontaneous but well organized to maximize profit.

Moreover, the rating distribution for different groups is given in Figs. 5.23 and 5.24. The horizontal axis represents the rating scale from one to five, and the vertical axis stands for proportion of the population. Blue is the promoting group, which denotes the subpopulation of suspicious reviewers for promoting; green is the demoting group; red represents the innocent group. Fig. 5.23 shows in New York City suspicious reviewers for promoting give ratings of four or five stars in most cases. However, suspicious reviewers for demoting do not rate one or two star(s) only, but also give a large number of five stars ratings. This is probably due to suspicious reviewers attempting to avoid the TripAdvisor's detection mechanism by injecting some random positive reviews. Another explanation is that, one user account is used for both promoting and demoting operation in order to maximize profit. In Fig. 5.24 which indicates the case in Hanoi, the suspects for promoting behave so extremely that almost all the ratings are five stars. This phenomenon shows that manipulative behavior is handled differently in different regions. The manipulation detection strength imposed on Hanoi is so weak that the suspicious reviewers for promoting do not need to post any review less than five in order to avoid from be detected out by TripAdvisor.

5.2.5.2 Trust Model Comparison

Given a set of suspicious hotels, different trust models are compared using RCI which was introduced in section 5.1.4. We have two options to choose a baseline trust model.

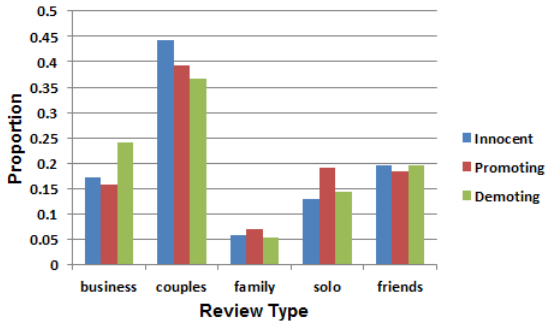

Figure 5.21: Review distribution for types of travel in New York City

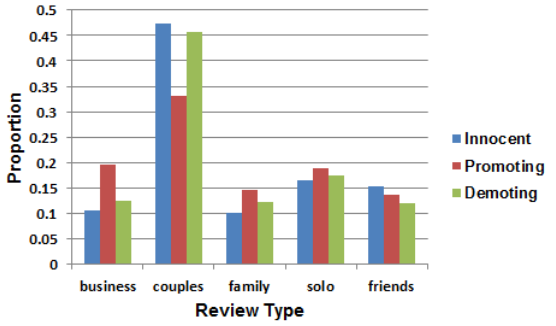

Figure 5.22: Review distribution for types of travel in Hanoi

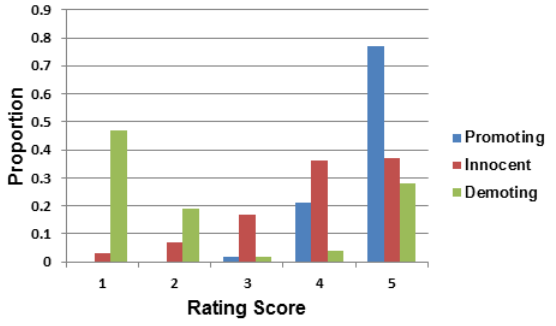

Figure 5.23: The distribution of ratings in New York City

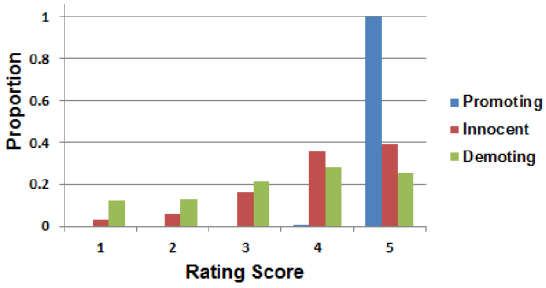

Figure 5.24: The distribution of ratings in Hanoi

The first option is the mean, which is considered as the key evaluation component in TA's algorithm (92). Since we have review history spanning years, a series of RCIs can be evaluated over time. Fig. 5.25 and 5.26 demonstrate the efficiency of different trust models against manipulation considering the mean as the base. RCI is evaluated monthly over 24 months. TW_1Y and TW_2Y are one-year and two-year time-window-based trust models respectively. TD_EXP is the trust model considering exponential time decay. SIP and SID denote the models considering SI of reviews and reviewers, corresponding to a hotel, for promoting and demoting respectively.

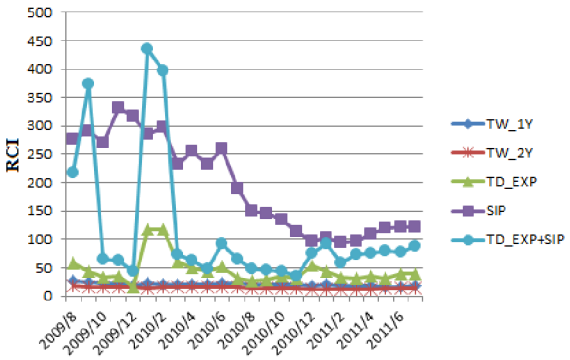

Figure 5.25: RCI index for different trust models designed to resist manipulation by promoting in New York City

As Fig. 5.25 shows, SIP is the best model, except for the three points before April 2010, where the hybrid trust model TD_EXP+SIP surpasses SIP. The explanation for this is that the three points in time suspicious reviews and reviewers are largely discounted by TD_EXP. The hybrid model is the second best choice in this case. Time-window-based trust models are almost as good as the mean.

In Fig. 5.26, the ranking of the trust models for Hanoi is different. The hybrid model TD_EXP+SIP is the best, with SIP performing second best. Note that, the same parameters for the trust models are used for both New York City and Hanoi. The result justifies that, according to diverse characters of different datasets, there is not a universal trust model which fits the best in every context. However, via SDM and RCI,

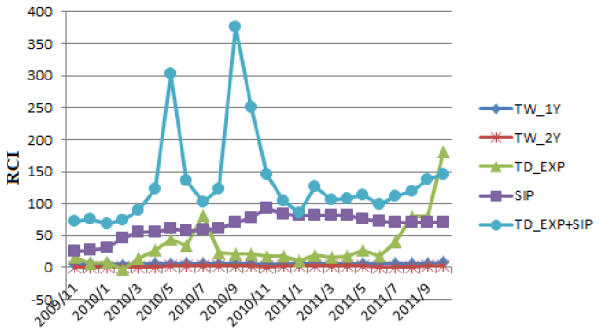

Figure 5.26: RCI index for different trust models designed to resist manipulation by promoting in Hanoi

it is possible for us to find the best solution for each case or a compromising candidate for most of the cases.

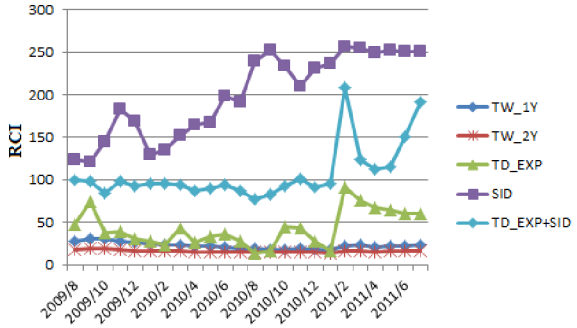

Figure 5.27: RCI index for different trust models designed to resist manipulation by demoting in New York City

For the case of demoting, which is demonstrated in Fig. 5.27 and 5.28, the SI-based model (SID) is the best choice in both regions. In Fig. 5.28 the hybrid model TD_EXP+SID is just as good as TD_EXP. At a few points in time, TD_EXP and the hybrid model are worse than the base model.

The second option is TA's algorithm, in which the popularity index[1] is considered to be the baseline trust model. Unfortunately, we don't have the formula for TripAdvisor algorithm, but we have collected the ranking of hotels in TripAdvisor. We require only the ranking of hotels to execute RCI. Since we have only the latest ranking information, only the current point in time can be evaluated by considering TripAdvisor algorithm as the base. The results are shown in Table 5.10. The numbers in Table 5.10, also show a similar result where the SI-based model and the hybrid model retain top rankings.

[1]http://www.tripadvisor.com/help/how_does_the_popularity_index_work

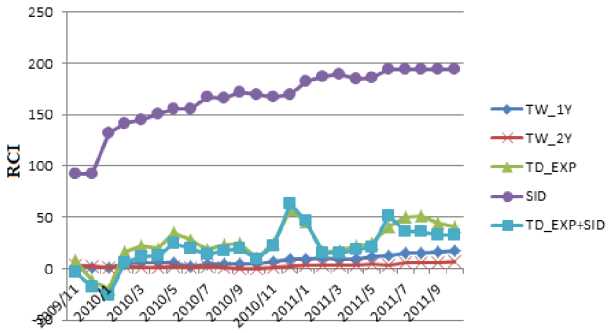

Figure 5.28: RCI index for different trust models designed to resist manipulation by demoting in Hanoi

	Type	TW_1Y	TW_2Y	TD_EXP	SI_based	TD_EXP+ SI_based
NYC	Promoting	16.62	13.42	35.64	119.48	81.33
HNO	Promoting	1.93	-4.60	204.85	61.10	136.54
NYC	Demoting	19.92	13.42	55.79	245.79	170.83
HNO	Demoting	6.98	-2.93	24.50	185.16	17.36

Table 5.10: RCI index for different models considering the TripAdvisor algorithm as the baseline performance indicator

Annotated Object	Number
Number of Genuine Reviews	180
Number of Promoting Reviews	139
Number of Demoting Reviews	24
Number of Genuine Reviewers	390
Number of Promoting Reviewers	131
Number of Demoting Reviewers	20
Number of Genuine Hotels	43
Number of Promoting Hotels	26
Number of Demoting Hotels	2

Table 5.11: Annotations statistics

5.2.6 Results for Supervised Learning of Manipulative Behavior

As we mentioned in subsection 5.2.3, supervised learning is difficult to perform due to the problem of data annotation. It is difficult to generate a gold-standard training data. In this section, we enhance the data annotation and generate training data. Based on the training data, a classifier is trained and the statistical characteristics of manipulative behavior is analyzed and discussed.

5.2.6.1 Annotations

Since we apply classic supervised learning approaches, having properly labeled data is the most significant part in our work. Before describing the annotation process, we have some comments on the previous work (67). Ott et al. used Amazon Mechanical Turk (AMT) to generate fraudulent reviews (67) and they mix fraudulent reviews with real reviews which are considered to be written by honest reviewers in TripAdvisor. Then human annotators are required to identify the fraudulent reviews from the mixture. One of the main findings of the experiment is that humans are inefficient at identifying fraudulent reviews. We agree with the finding that humans are inefficient, yet we argue that the idea of generating fraudulent reviews using AMT has its own limit. It is unclear whether the characteristic of fraudulent reviews written by virtual workers on AMT is similar to that in TripAdvisor[1]. Furthermore, the annotators make a decision based on the review text only (67). A better solution is to identify fraudulent reviews which are extracted from a dataset using all available complimentary information given, for example, checking various reviews from the same reviewer or the date they were posted. We assume that if the annotation process is carefully handled, an appropriate gold standard can be manually generated.

We select three well-trained and independent annotators. Well-trained means every annotator has at least a basic notion of manipulative behavior. They are encouraged to evaluate each review by identifying logical inconsistency within the information which

[1]http://tripadvisorwatch.wordpress.com/2010/10/10/tripadvisor-pay-review-fake

is related to a review. The information does not only refer to the numerical and textual value of a review per se, but all types of information about the corresponding reviewer, such as uploaded pictures, reviewer profile etc. The annotators make their decisions on facts like one uploaded picture was the only one looking quite different from the pictures uploaded by other reviewers. We randomly pick 1000 reviews from the dataset in New York City only, whose overall score is either one or five star(s), and let all of the annotators evaluate the same 1000 reviews separately. We believe that a review with the lowest or highest score is most likely to be suspicious. The annotation process is a very time-consuming procedure, since an annotator has to check a lot of information in order to make a decision. Moreover, we calculate the inter-annotation agreement using the Fleiss' Kappa, which is $\kappa = 0.18$. This indicates only slight agreement, which is consistent with the findings (67). To provide reliable labels, we chose only those reviews for our final gold standard that were unanimously labeled by all three annotators. Thus having a complete agreement level and considering the fact that our annotators made use of all information provided about the review, the reviewer and the hotel, we assume the labels in the gold standard to constitute the truth.

So far, only reviews are labeled, but we still need to label reviewers and hotels. Considering logical relations among different object levels, a set of labeled suspicious reviewers and hotels can be generated from labeled reviews. There are two logical arguments that we use. If a review is suspicious, the corresponding reviewer is also suspicious; if a number of reviews posted about a hotel are all suspicious, the hotel is also suspicious. Following this idea, the sets of suspicious reviewers and hotels are generated. In addition, in our previous work, we succeeded in assigning a Suspicion Index (SI) to the objects at review, reviewer and hotel levels (68). Fig. 5.17 demonstrates the distribution of the SI at the reviewer level with respect to promoting manipulation. The data can be fitted by an exponential function. In this case, we simply set a threshold for SI (e.g. 0.01) to choose a set of genuine reviewers with respect to promoting manipulation. The statistics of annotated objects is listed in Table 5.11.

5.2.6.2 Feature Evaluation

To illustrate the effectiveness of the features introduced in section 5.2.2, we plot the distribution of feature values with respect to genuine and suspicious objects considering the gold standard annotations. In this section, we sample the two most representative features.

The average number of reviews per month (AveNumPerMonth) is one of the most representative features specified at the hotel level. The value distribution of AveNumPerMonth is plotted in Fig. 5.29. All hotels are ordered by their AveNumPerMonth value, which is represented on the y-axis. The x-axis corresponds to the indices of the hotels. There are three groups of hotels: those with genuine reviews (innocent group), those with promoting reviews (promoting group) and those with demoting reviews (demoting group). The values of the demoting group clearly differ from those of the genuine group. Comparing the promoting group to the genuine group, all of the hotels whose AveNumPerMonth is greater than 15 are suspicious. This numerical characteristic can be captured by supervised learning.

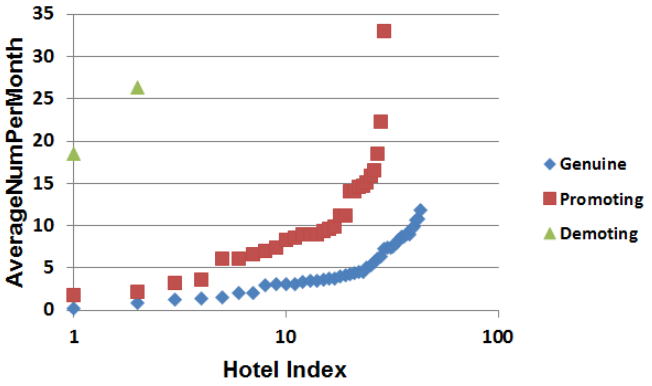

Figure 5.29: Average number of reviews per month

Contribution mean is a feature specified at the reviewer level. Its value distribution is plotted in Fig. 5.30. On the x-axis, there are five intervals and each represents a range of values. The y-axis denotes the percentage of reviewers whose feature value falls into this range. Fig. 5.30 shows the fact that the range of contribution mean of the genuine group is in the range four to five (four is exclusive), which is determined by the way we generate the labels for genuine reviewers in the annotation process. Contribution mean of the promoting group is mostly distributed in the range four to five (four is exclusive), whereas that of the demoting group is distributed in the range zero to four (zero is exclusive). Again, boundaries among the different groups can be learned.

5.2.6.3 Learning Results

The main learning results are shown in Table 5.12. For machine learning, we use the toolkit Weka[1]. Due to the experience of previous work (62, 67), several classic supervised learning approaches are applied, such as a linear logistic regression model, SVMs and a Naive Bayes classifier. Since SVMs clearly outperform other classifiers in our evaluation, we only show the classification results regarding SVMs. Achieving accuracies above 90%, identifying manipulative behavior at hotel and reviewer level seems to work well. Especially demoting manipulation could be detected correctly in all cases.

However, the classification results at the review level are not as good as we expected. All the scores, accuracy, precision, recall and F-score, are much lower than those at the reviewer and hotel level. Although the accuracies ranging between 65% and 84% do not seem to be that low, the actual performance for detecting fraudulent reviews has an f-measure as low as 13% for demoting behavior. Comparing non-textual features and textual features, the latter ones clearly outperform non-textual features especially regarding demoting manipulation classification. We draw the conclusion that it is extremely difficult to identify fraudulent reviews. More representative features for

[1] www.cs.waikato.ac.nz/ml/weka/

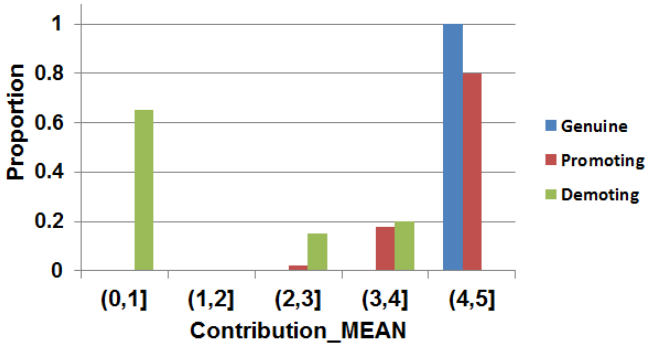

Figure 5.30: Contribution mean

			Genuine			Fraudulent		
Types	**Features**	**A**	**P**	**R**	**F**	**P**	**R**	**F**
$\text{Hotel}_{\text{PMB}}$	Non-Textual	**91.3%**	100%	87.8%	93.5%	76.9%	100%	87%
$\text{Hotel}_{\text{DMB}}$	Non-Textual	**100%**	100%	100%	100%	100%	100%	100%
$\text{Reviewer}_{\text{PMB}}$	Non-Textual	**96.4%**	100%	95.4%	97.6%	85.5%	100%	92.2%
$\text{Reviewer}_{\text{DMB}}$	Non-Textual	**100%**	100%	100%	100%	100%	100%	100%
$\text{Review}_{\text{PMB}}$	Non-Textual	65.2%	71.1%	68.4%	69.8%	57.6%	60.6%	59%
$\text{Review}_{\text{PMB}}$	UniBigram	68.3%	76.7%	70.1%	73.2%	57.6%	65.6%	61.3%
$\text{Review}_{\text{DMB}}$	Non-Textual	80.4%	89.4%	88.5%	89%	14.3%	13.6%	13%
$\text{Review}_{\text{DMB}}$	UniBigram	84.3%	90%	92%	91%	41.7%	35.7%	38.5%

Table 5.12: Classification Results, where PMB for Promoting Manipulative Behavior, DMB for Demoting Manipulative Behavior, A for Accuracy, P for Precision, R for Recall and F for F-Score (3). UniBigram denotes both Unigram and Bigram are considered during learning process. Non-textual denotes all the corresponding features described in section 3.

identifying suspicious reviews need to be developed. In the following section, we will focus on the results at the reviewer and hotel level only.

5.2.6.4 Feature Selection

In this section, we explore the performance of the features. Given human annotations, features are ranked by the weight assigned by the SVMs (93). Table 5.13 shows the top five features for suspicious hotel classification. As we expected, average number of reviews per month (AveNumPerMonth) is the best feature for identifying promoting manipulation, and the second best for identifying demoting manipulation. A hotel suffering from demoting manipulation usually has a large value for AveNumPerMonth, since in order to recover from slandering, the hotels "hire" reviewers to give fraudulent positive reviews. The singleton related features such as PropAdvPosSing and

Ranking	$\text{Features}_{\text{promoting}}$	$\text{Features}_{\text{demoting}}$
1	AveNumPerMonth	Rating_VAR
2	Rating_VAR	AveNumPerMonth
3	RatioRoomReview	PropAdvNegSing
4	TurningDay	Rating_MEAN
5	PropAdvPosSing	VAR_MODE

Table 5.13: Top 5 features at the hotel level

Ranking	$\text{Features}_{\text{promoting}}$	$\text{Features}_{\text{demoting}}$
1	Contribution_MEAN	Contribution_MEAN
2	InactiveDuration	ContributionVAR
3	ContributionVAR	ContributionNum
4	TurningDay	InactiveDuration
5	ContributionNum	TimeConsecContributions_MEAN

Table 5.14: Top 5 features at the reviewer level

PropAdvNegSing are shown in the list as well.

Table 5.14 shows the top five features for suspicious reviewer classification. As we expected, Contribution mean (Contribution_MEAN) is the top rank for both promoting and demoting manipulation. Inactive duration (InactiveDuration) is ranked second for promoting manipulation detection, since providing a singleton review usually implies a large value for InactiveDuration. Contribution variation (ContributionVAR) is ranked third for promotion manipulation and second for demoting manipulation.

5.2.6.5 Statistical Characteristics of Suspects

In this section, we investigate uncertain assumptions and explore statistical characteristics of suspects by considering the predictions made by our trained classifiers.

In section 5.2.2, we specify hotel reviews contradiction degree with the expectation that the larger the Hotel reviews contradiction degree a hotel has, the more suspicious the hotel is. Applying the same approach for feature evaluation, we plot the hotel reviews contradiction degree distribution for both promoting and demoting manipulation in Fig. 5.31 and 5.32. Hotels are ranked by their Hotel reviews contradiction degree value. The two figures show that the Hotel reviews contradiction degree the suspicious and the genuine groups completely overlap. This result rejects the validity of Hotel reviews contradiction degree as a representative feature. Hotel reviews contradiction degree is not very useful feature for detection of manipulation.

O'Mahony considers the helpfulness of a review as a representative feature for evaluating the trustworthiness of a review (75). We cannot evaluate this hypothesis at the review level where we do not have a good classifier. However, we can still learn some similar notion at the reviewer level where we have qualified classifiers. The helpfulness

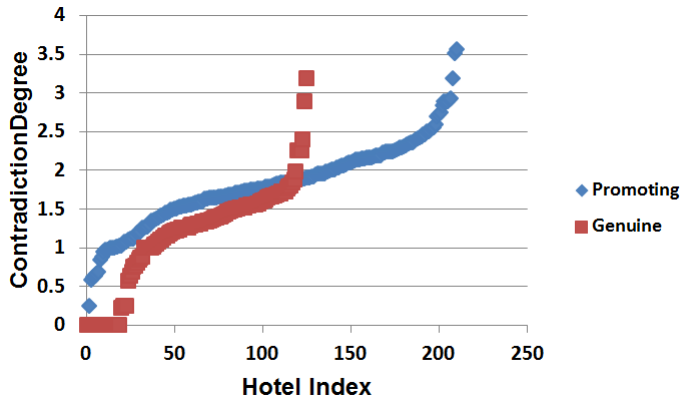

Figure 5.31: Hotel reviews contradiction degree evaluation result for promoting behavior

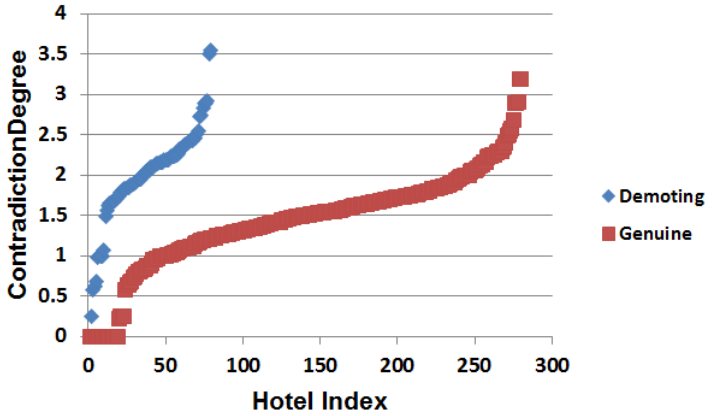

Figure 5.32: Hotel reviews contradiction degree evaluation result for demoting behavior

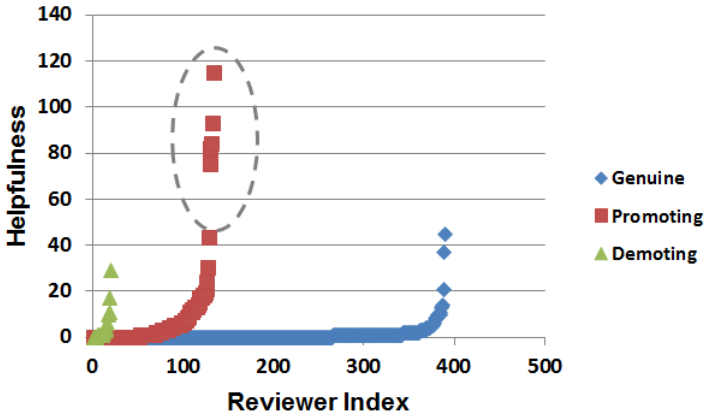

Figure 5.33: Reviewer helpfulness distribution

of a reviewer is equal to the sum of helpfulness of all the reviews which are provided by the reviewer. Fig. 5.33 shows the distribution of helpfulness of reviewers with respect to the different groups. In the dotted circle area, the helpfulness of the promoting group is much larger than that of the genuine group. This is an indirect evidence to reject the hypothesis that the more reviewer helpfulness, the less suspicious regarding manipulation.

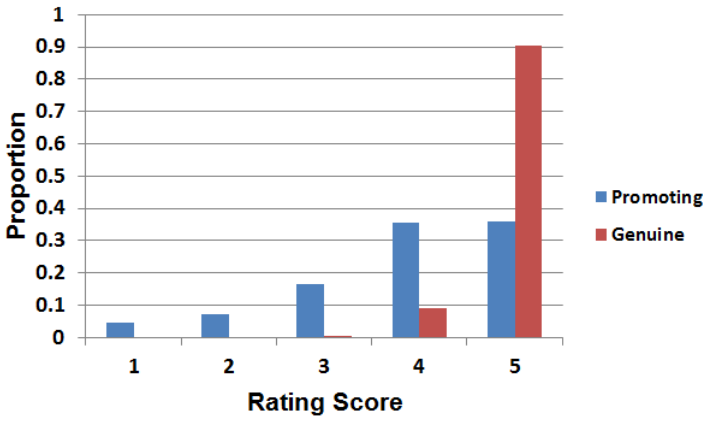

Figure 5.34: Rating distribution for promoting behavior

We also explore the statistical characteristics of the whole population. After having learned good classifiers at both reviewer and hotel levels, we generate the predictions at the two levels considering all the reviewers and hotels. One of the most important questions is what the rating distribution looks like with respect to different groups of reviewers. Do reviewers who try to promote a hotel always give five stars? Similarly, do reviewers who try to demote a hotel always give the lowest rating? The results are shown in Fig. 5.34 and 5.35. From Fig. 5.34 we can see that, genuine reviewers provide mostly four or five stars, whereas promoting reviewers provide all from one to five star(s). The proportion of one or two points given by promoting reviewers is much larger than that given by genuine reviewers. This phenomenon implies that reviewers who intend to promote a hotel provide more negative ratings than honest reviewers, which is a very counterintuitive result. A reasonable explanation is that diversity of review provisioning is a strategy for promoters to avoid being identified by TA's manipulation detection algorithm. An alternative explanation is that in order to maximize profit, a reviewer provides both positive and negative fraudulent reviews. In the case of demoting, which is plotted in Fig. 5.35, reviewers do not only provide negative fraudulent reviews but positive reviews as well. The explanation is similar as before. Another important result that we can derive from the two figures is that, most of the negative reviews are fraudulent. As you can see, few genuine reviewers give one star or two stars rating.

Regarding the ranking of hotels in terms of trust value, the ranking distribution of different groups is shown in Fig. 5.36. Three groups are extracted from the prediction which is generated by the trained classifiers. The promotion group refers to the set of suspicious hotels which are predicted to be related to promoting manipulative behavior;

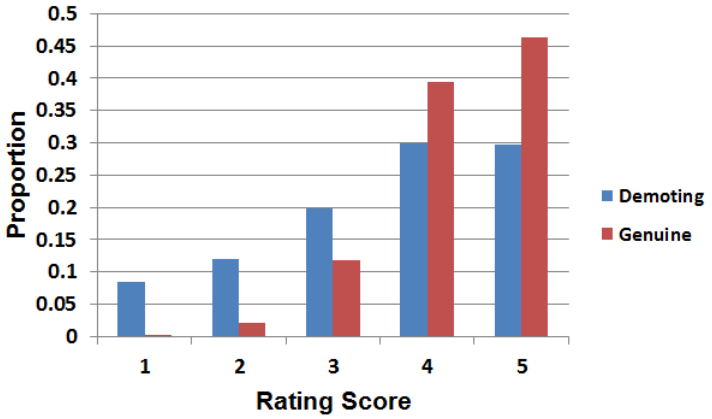

Figure 5.35: Rating distribution for demoting behavior

the demotion group refers to the set of suspicious hotels which are predicted to be related to demoting manipulative behavior; the both group is the intersection of the first two groups. The x-axis represents 10 intervals in which hotels fall in terms of their ranking. For instance, the top 10% ranked hotels fall into the first interval and so on. The y-axis denotes the number of suspicious hotels which fall into an interval. Fig. 5.36 shows that manipulation appears in all intervals and promoting manipulation is much more popular than demoting manipulation. Note that this result is derived from TA, which probably already applied manipulation detection mechanisms. Even considering TA applying manipulation detection, there are still many suspects existing in the system. Another fact is that most suspicious hotels suffering from demoting manipulation are also related to promoting manipulation. It seems that promoting behavior is triggered by demoting behavior, since in order to recover from demoting behavior, hotels "hire" reviewers to provide fraudulent positive reviews.

In this section, datasets from New York City and Hanoi are collected. Both the unsupervised learning approach namely suspicion degree meter (SDM) and the supervised learning namely classification approach are applied for identifying promoting and demoting manipulation.

Following the basic assumptions AS1 to AS4, SDM can detect sets of suspects at different levels: the review, the reviewer and the hotel level. Given sets of suspects, statistical characteristics of suspicious group and innocent group are compared for each level. The assumption that review helpfulness can be used to calculate the trustworthiness of reviews is not valid in the dataset HNO. The unbalanced manipulation detection force imposed by TripAdvisor in different regions (e.g. NYC and HNO) is shown. Furthermore, given a set of suspicious hotels, different reputation models are compared via RCI. The results show that, superiority of reputation model is dependent on the choice of model parameters and feature of dataset per se. Although there is not a universal reputation model which fits best for every circumstance, given a set of suspects in all the layers identified by SDM, local optimization can be achieved.

For supervised learning, annotations regarding review, reviewer and hotel levels are generated by unanimous voting and the results from SDM. Using the annotations and

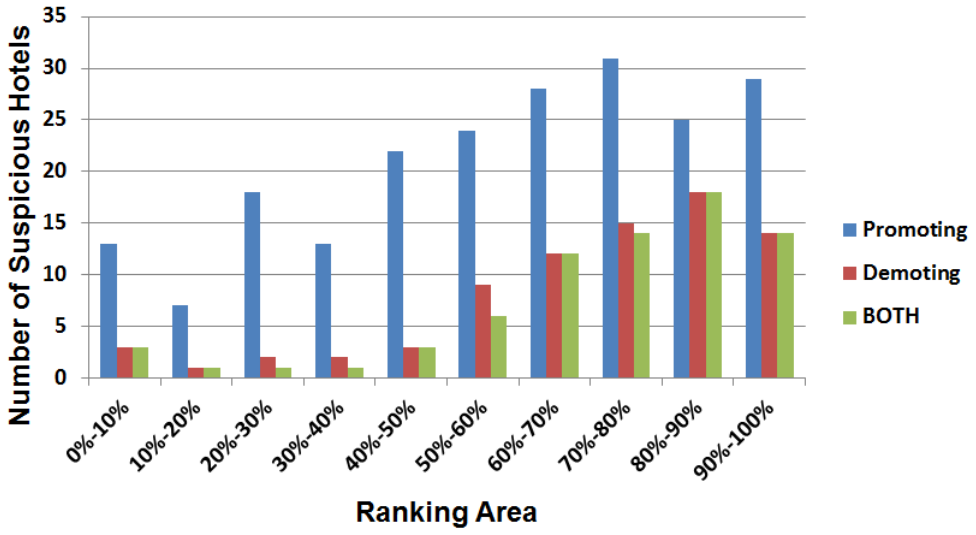

Figure 5.36: Trust ranking distribution for hotels

SVM, classifiers are learned using Weka. Results show that we are able to learn good classifiers for layers of reviewer and hotel, but not for layer of review even considering both non-textual and textual features. The rating distribution with respect to different groups, genuine, promoting and demoting groups, shows that suspicious reviewers provide reviews with a large variation. The reason could be either a suspicious reviewer provides both fraudulent positive and negative reviews in order to maximize profit, or he/her does this for avoiding from being detected by TripAdvisor's manipulation detection mechanism.

5.3 A Case Study on Lifestyle-Related Services

In this section we choose Dianping.com[1] as an example of a platform where lifestyle-related services are widely reviewed and discussed. Dianping is a Chinese rating platform, where people can post their reviews on any lifestyle-related service. Unfortunately it has been pointed out that there are a lot of suspicious reviewers who post fraudulent reviews[2]. In order to identify suspicious reviewers and the corresponding fraudulent reviews on reviewing websites like Dianping.com, the features with respect to reviewers are specified. We propose an advanced clustering approach, Annotation-Auxiliary Clustering (AAClust), to identify reviewers suspected of manipulation. In order to extend our knowledge of manipulation identification, we analyze social network information for innocent and suspicious reviewers. The knowledge could be used to enhance our machine learning process.

[1] www.dianping.com
[2] www.sootoo.com/content/398411.shtml

5.3.1 Lifestyle-Related Services and Dianping.com

A lifestyle-related service is an offline service which could be used in daily life, such as a restaurant, caf, bar, shopping, doctor, lawyer, etc. Lifestyle -related services cover a large spectrum of services and the evaluation of a lifestyle-related service is highly subjective. A restaurant is a typical example, since different people prefer different cuisines. In Dianping, there is a variety of categories of lifestyle-related service such as restaurant, shopping, doctor, etc. In every category, there are hundreds of instances of service on which a reviewer can post reviews. For example, a reviewer can post a review on a restaurant in Beijing named "See You Next Time". A review is composed of a total score, some sub-scores and a comment. A score is a number ranged from one to five. The specification of categories of sub-scores depends on the type of service. For a restaurant, the categories of sub-scores are the taste, the environment and the service. For another service, the specification of categories can be different. In addition, Dianping provides some social networking features. A reviewer can make friends and keep a friend list in Dianping. A reviewer can send a flower to another reviewer in order to present a sense of complement to the reviewer who posts a nice review.

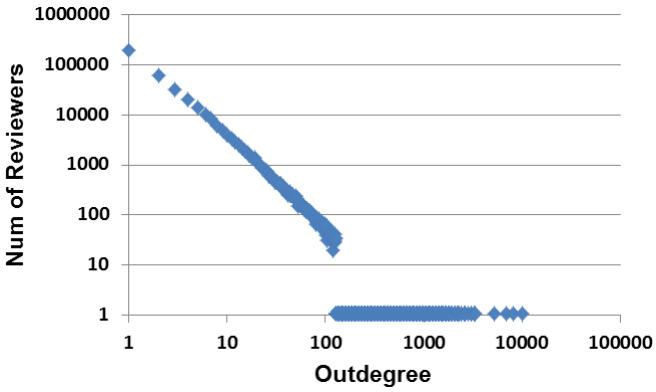

Figure 5.37: Number of reviewers vs. out-degree

Reviewer information is collected as follows. Since the whole reviewer population is large, the first question is which subset of reviewers to select. Due to the social networking feature in Dianping, the relationships among reviewers in Dianping can be regarded as a graph. Crawling large, complex graphs like Dianping presents a challenge. In order to explore the social network information among reviewers, we should collect a subset of reviewers who keep the original social network structure such as weakly connected component(WCC)[1]. Breadth-first search (BFS) can help us to collect such a subset (94). The collection of the subset started from a power node which is a reviewer who posted over 1000 reviews and has more than 500 friends. In each step, we retrieve the list of friends for a reviewer is retrieved we had not yet visited and add the retrieved reviewers to the list of reviewers. Finally we collect 380,489 reviewers and 1,471,610

[1]A weakly connected component in a directed graph is a set of nodes where each node in the set has a path to every other node in the set if all links are viewed as undirected.

Object	Size
COLLECTED	
Reviewers	380,489
Friendship relation	1,471,610
SELECTED (20%)	
Reviewers	78,378
Friendship relation	104,924
Flower relation	1,053,759

Table 5.15: Statistics of the Dianping.com dataset

friendship relations. The distribution of the out-degree of reviewers is plotted in Fig. 5.37. The out-degree of a reviewer refers to the number of their friends. As one can see, the distribution follows a power law.

We put all the 380,489 reviewers into bins which correspond to different out-degrees. For each bin, we randomly choose a subset of 20% (78,378) of reviewers from all the reviewers. We collect information such as the posted reviews, number of received flowers, number of uploaded pictures, gender, city, etc. for each reviewer. The basic statistics of our dataset are given in Table 5.15.

5.3.2 Feature Identification

We specify the following features of a reviewer as input data for the subsequent phase of machine learning:

- Gender: There are three options: male, female and N/A. N/A is the default when the reviewer does not provide the information.
- Number of reviews. This is the total number of reviews posted by the reviewer.
- Wish list: This is the list of service providers preferred by the reviewer. For instance, a reviewer might keep a list of favorite restaurants as a wish list.
- Number of uploaded pictures: This is the total number of pictures uploaded by the reviewer. The pictures could be personal photos, photos of a restaurant, or whatever.
- Number of recommendations: A recommendation refers to a list of recommended service providers posted by the reviewer. Here we count the total number of such recommendations.
- Number of friends: This is the reviewer's out-degree.
- Forum activity degree: This is the total number of posts the reviewer has made in the Dianping.com forum.

- Contribution: This is a value indicating the contribution a reviewer delivers in Dianping. The main factors are posted reviews, uploaded pictures, recommendations, manipulative behavior, and personal profile completeness. Contribution is calculated by Dianping, note that contribution could decrease due to identification of manipulative behavior. Contribution can be regarded as an internal trust value in Dianping.

- Number of badges: This is the number of badges which a reviewer receives. For instance if a reviewer posts a certain amount of reviews for a service such as hotpot restaurants, the reviewer will receive a special badge for that. Here we count the number of badges held by the reviewer, thus measuring his level of activity in Dianping.

- Number of received flowers: This is the total number of flowers received by the reviewer. Number of updated activities: It refers to the number of recent actions of the reviewer such as a new posted review, a new recommendation, etc. This number measures his current level of activity.

- Inactive duration: It is the elapsed time between the reviewer's initial registration and the last login time.

- Life span: It is the time interval between registration and the last login time.

- Number of replies: Total number of replies made by other users to any of the reviews posted by the reviewer.

- Average number of replies: Average number of replies made by other users to any of the reviews posted by the reviewer.

- Number of flowers received per review: Total number of flowers received per review posted by the reviewer.

- Average number of flowers received per review: Average number of flowers received per review posted by the reviewer.

- Average words per review: Average number of words per review posted by the reviewer.

- Maximum number of reviews: Maximum number of reviews posted by the reviewer within different time intervals. We consider the interval to be daily, monthly and annual, respectively. Thus we have three parameters: Maximum number of reviews per day, maximum number of reviews per month, and maximum number of reviews per year.

- Minimum number of reviews: Minimum number of reviews posted by the reviewer within different time intervals. Following the pattern described above, we get three parameters: Minimum number of reviews per day, minimum number of reviews per month, and minimum number of reviews per year.

- Average number of reviews: Average number of reviews posted by the reviewer within different time intervals. Following the pattern described above, we get three parameters: Average number of reviews per day, average number of reviews per month, and average number of reviews per year.

5.3.3 Annotation-Auxiliary Clustering (AAClust)

Here we propose an advanced clustering approach, Annotation-Auxiliary Clustering (AAClust), for identifying manipulative behavior. The basic idea is to use annotations to evaluate the quality of clustering. In this case study, we need annotations to help us to make a decision that which cluster is more suspicious than the others. For Dianping.com, we cannot make the basic assumptions as for Taobao.com and TripAdvisor.com which help us to find suspicious groups.

Compared to a travel-related service, it is more difficult to trace whether a transaction for a lifestyle-related service takes place. In booking.com[1], the service (hotel) consumption and the given rating are inherently bound. A reviewer usually books and pays for a hotel on the website before posting a review of a hotel. In this case, it is expensive for a dishonest hotelier to promote his property; it is even more expensive for a competitor of the hotelier to slander the property. Normally, this inherent binding does not exist for a lifestyle-related service. Therefore, manipulation identification for lifestyle-related service is more challenging. Anyone with a valid account in Dianping.com can post reviews of any service. There are few patterns we could identify in advance for identifying manipulative behavior. Meanwhile, it is extremely time-consuming to annotate a large number of reviewers, since the annotators need to look into all kinds of information regarding the behavior of a reviewer such as posted reviews, uploaded pictures, recommendations, friends list, etc. If the annotated set of reviewers is too small, a supervised learning approach, such as SVMs, does not work.

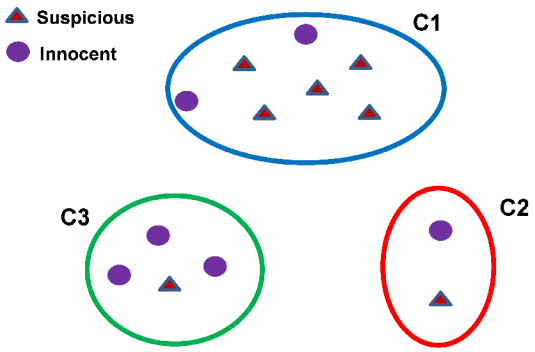

Figure 5.38: An illustration for Annotation-Auxiliary Clustering, where circles represent innocent objects and triangles represent suspicious reviewers. After clustering, every annotated reviewer is assigned to one of the three clusters, C1, C2 or C3.

[1] www.booking.com

We illustrate the process of AAClust in Fig. 5.38. First, we apply a classic clustering algorithm such as K-means[1]. We assume that all the objects, including annotated objects, are clustered into three clusters, C1, C2 and C3. For Dianping, an object refers to a reviewer. The reviewers that are annotated as suspicious or innocent fall into clusters as shown in Fig. 5.38. We define a ratio rat_i, which is given by formula 5.27, as an extra label for each cluster. N_i^{sus} stands for the number of suspicious reviewers in cluster i; N_i^{inn} stands for the number of innocent reviewers in cluster i. Therefore, $rat_{C1} = 5/2$, $rat_{C2} = 1$ and $rat_{C3} = 3$. We set a threshold th_{ratio} for identifying the labelled clusters with high confidence. The manipulative behavior status of the reviewers in the rest of the clusters is unknown.

$$rat_i = \begin{cases} 0 & \text{if } N_i^{sus} = N_i^{inn} = 0 \\ \frac{N_i^{sus}}{N_i^{inn}} & \text{if } N_i^{sus} \geq N_i^{inn} \text{ and } N_i^{inn} > 0 \\ +\infty & \text{if } N_i^{sus} \geq N_i^{inn} \text{ and } N_i^{inn} = 0 \\ \frac{N_i^{inn}}{N_i^{sus}} & \text{if } N_i^{sus} < N_i^{inn} \text{ and } N_i^{sus} > 0 \\ +\infty & \text{if } N_i^{sus} < N_i^{inn} \text{ and } N_i^{sus} = 0 \end{cases} \tag{5.27}$$

Four metrics for measuring the quality of clustering have previously been introduced (74). They are purity, normalized mutual information, rand index and F measure. Purity is used in our work due to its simplicity and easy-understandability. To compute the purity metric, each cluster is assigned a label according to the occurrence of the most frequently observed annotated reviewers within the cluster. In Fig. 5.38, there are three clusters (C1, C2 and C3) and two labels (innocent and suspicious). The accuracy of the assignment is measured by counting the number of correctly assigned reviewers and dividing by the number of annotations. Purity is formalized in formula 5.28, where $\Omega = \{\omega_1, \omega_2, ..., \omega_k\}$ is the set of clusters, $L = \{l_1, l_2, ..., l_j\}$ is the set of labels and N is the number of annotations. For the example in Fig. 5.38, purity is $(5 + 3 + 1)/13 \approx 0.62$.

$$purity(\Omega, L) = \frac{1}{N} \sum_k \max_j |\omega_k \cap l_j| \tag{5.28}$$

5.3.4 Experimental Results

In this section, we show the candidates of innocent and suspicious reviewers. Different feature types for innocent and suspicious reviewers are compared. Particularly we introduce two metrics, degree correlation k_{nn} (94) and degree distribution $P(k)$ (95) to show the social network information of innocent and suspicious reviewers.

[1] http://en.wikipedia.org/wiki/K-means_clustering

5.3.4.1 AAClust Learning Results

In order to select candidate suspicious reviewers, we manually annotated 168 suspicious and 56 innocent reviewers. Then we performed supervised learning (SVMs) using the 224 annotations. As we expected, the supervised learning results have as low as 68% accuracy. The reason why the supervised learning results are poor is that, the quantity of the annotated set is too small.

Using AAClust, we cluster all the 78,378 reviewers using K-means considering the parameters such as seed and number of clusters with different values. th_{ratio} is set to 10, which means the number of reviewers with one type of label (either suspicious or innocent) are 10 times larger than that with the other type. After the fixing of th_{ratio}, the candidates of suspicious and innocent clusters are selected. Meanwhile, we accept only cluster where purity is higher than 75%. We identify 9045 suspicious and 42351 innocent reviewers in the end. There are over 11.5% suspicious reviewers in our dataset. This number indicates that there are many suspicious reviewers in Dianping.com.

5.3.4.2 Feature Comparison

We compare the distribution of ratings for suspicious reviewers to that of innocent reviewers. In Dianping.com, a review is composed of a total rating score, a set of rating scores for features of a service and a textual comment. Features of a service refer to a refinement of service quality. For instance, a lifestyle-related service like a restaurant has three features: taste, environment and service. For the distribution of ratings, we consider only the total rating score of a review.

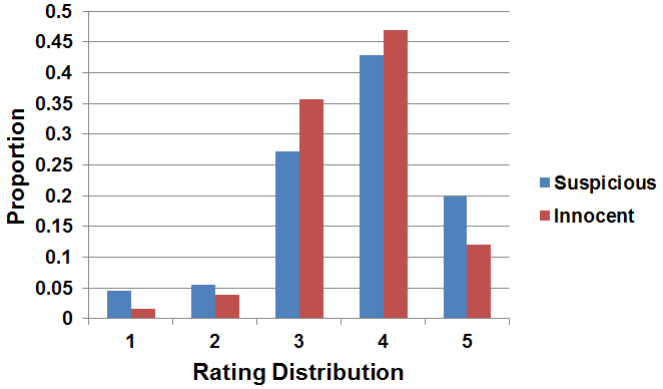

Figure 5.39: Rating distribution

Comparison of the distribution of ratings is shown in Fig. 5.39. The horizontal axis represents rating distribution, which is an integer between one and five. The vertical axis represents the proportion of reviews providing a certain rating score. The two different groups are color coded. The suspicious group which is composed of all the identified suspicious reviewers is in blue; the innocent group is in red. The distributions of two groups look very similar; the suspicious group has slightly more extreme ratings (e.g. rating one and five) than the innocent group. The rating in a review does not

play a role in manipulation because Dianping does not have a strong trust management system. In Dianping, a fraudulent review concentrates more on constructing the textual comment.

In Dianping.com, reviewers are strongly related via a social network. There are two types of social relationships. The first type is a friendship. A reviewer can request or accept a friendship with other reviewers. The second type is a flower relation. A reviewer can send a flower to another reviewer in order to present a sense of complement to the reviewer who posts a nice review. Here we explore the character of the two types of social network regarding the suspicious and innocent reviewers identified using AAClust.

We introduce degree distribution $P(k)$ (95) and degree correlation k_{nn} (94) to analyze the topology of the social network in Dianping.com. A power-law degree distribution, $P(k) \sim k^{\gamma}$, where k is the node degree, attest to the existence of a relatively small number of nodes with a very large number of relations. The degree distribution $P(k)$ is plotted as a complementary cumulative probability function (CCDF), which is given by formula 5.29.

$$\bar{F}(x) = P(X > x) = 1 - F(x) \tag{5.29}$$

The degree correlation, k_{nn}, is a mapping between node out-degree k and the average in-degree of neighbors of nodes with out-degree k. An increasing k_{nn} indicates a tendency that higher-degree nodes connect to other high-degree nodes; a decreasing k_{nn} represents the opposite trend.

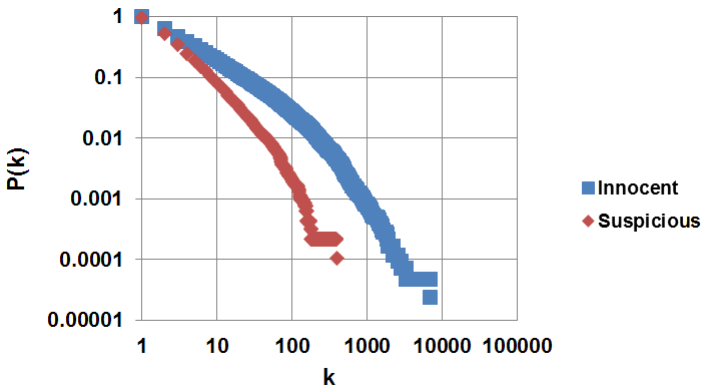

Figure 5.40: Degree distribution for friendship

Fig 5.40, 5.41 and 5.42 are log-log plots of the degree distribution for friendship degree, in-degree and out-degree respectively. In a directed graph, the degree of a node is equal to the sum of in-degree and out-degree of the node. The horizontal axis represents the degree k; the vertical axis represents $P(k)$. The innocent reviewers group is shown in blue; the suspicious group is shown in red. The three figures show three common phenomena. The first phenomenon is that both two groups show general power-law character. The second phenomenon is that, the innocent group has more reviewers whose degree is larger than the suspicious group. This phenomenon is shown

again in Fig. 5.41, where the in-degree of reviewers in the suspicious group is less than 100, while that in the innocent group is approaching 10000. This phenomenon implies that there are fewer reviewers connected to suspicious reviewers than to innocent reviewers. The third phenomenon is that the line of CCDF for the suspicious reviewers decreases faster than that of the innocent reviewers. This phenomenon implies that innocent reviewers have larger degree than the suspicious reviewers, which is a positive sign for system robustness, because the suspicious reviewers do not have as large a social influence as the innocent reviewers.

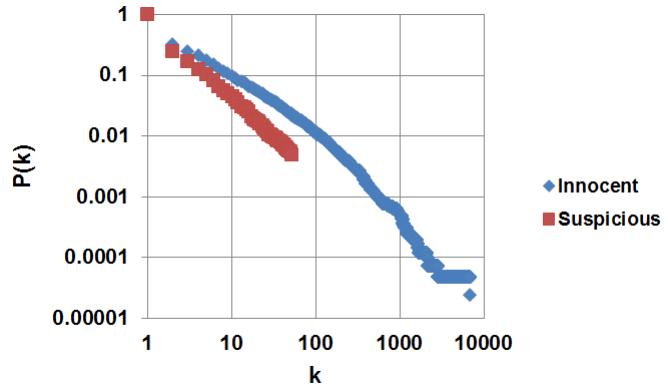

Figure 5.41: In-degree distribution for friendship

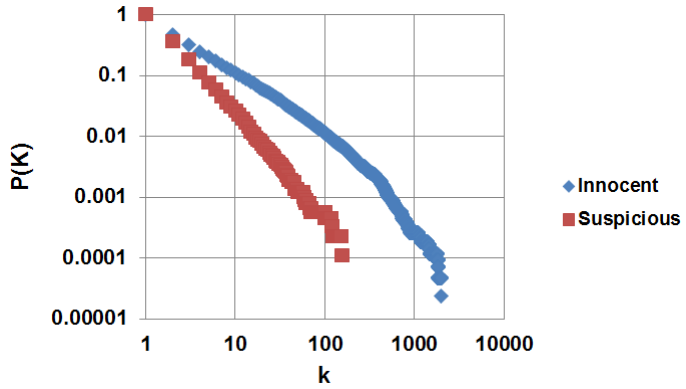

Figure 5.42: Out-degree distribution for friendship

Fig 5.43, 5.44 and 5.45 plot the distributions for the flower relationships: degree, in-degree and out-degree, respectively. In all the three figures, we can see that the flower relationship doesn't follow a power-law distribution, because the number of reviewers whose degree is equal to one is extremely large. Most of the reviewers in Dianping give or receive only one flower to or from other reviewers. In addition, from Fig. 5.44 we can see that, when in-degree is smaller than 40, both groups have the similar distributions regarding the in-degree. Similarly to friendship, the line of CCDF for the suspicious

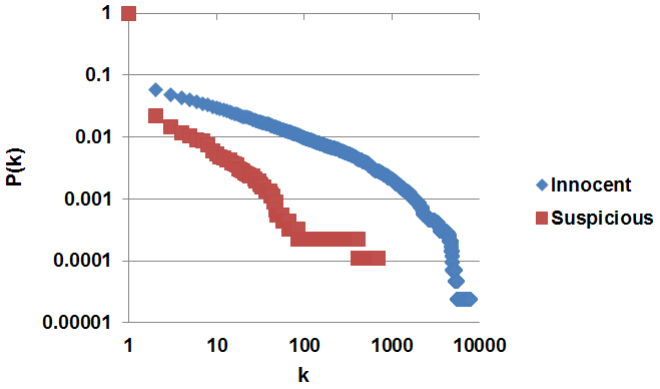

Figure 5.43: Degree distribution for flower relationships

reviewers decreases faster than that of the innocent reviewers. In other words: the innocent reviewers have larger degree than the suspicious reviewers.

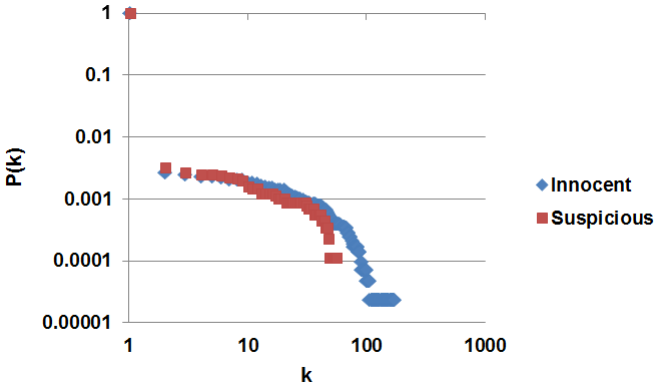

Figure 5.44: In-degree distribution for flower relationships

Fig. 5.46 shows the degree correlation k_{nn} for friendship. In the figure, the degree correlation for the two reviewer groups, innocent and suspicious, decreases when degree k increases. The decrease implies that there are a few extremely popular reviewers in Dianping.com to whom many unpopular reviewers connect. In addition, we can see that the innocent reviewers have extremely high k_{nn} when degree k is smaller than ten. The suspicious reviewers generally have smaller k_{nn}. This tells us that the suspicious group contains much less unpopular reviewers than the innocent group. The k_{nn} regarding innocent and suspicious groups have a similar shape.

Fig. 5.47 shows the degree correlation k_{nn} for flower relationships. k_{nn} of the innocent reviewers stays almost still. On the other hand, k_{nn} of the suspicious reviewers decreases rapidly when degree k increases. Particularly when degree k is larger than ten, k_{nn} is approaching one. It implies that the flower which a popular suspicious reviewer sends to another reviewer is mainly the only flower the reviewer receives.

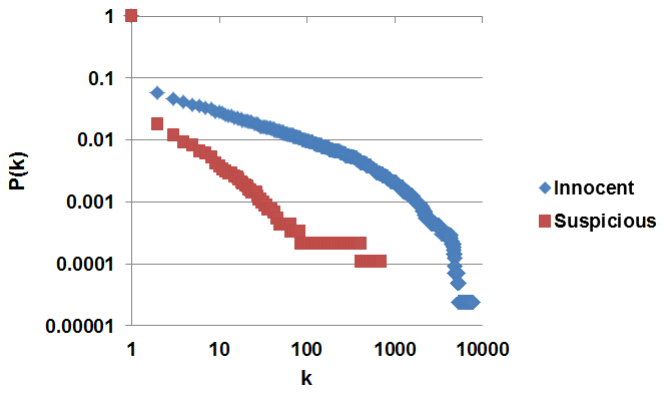

Figure 5.45: Out-degree distribution for flower relationships

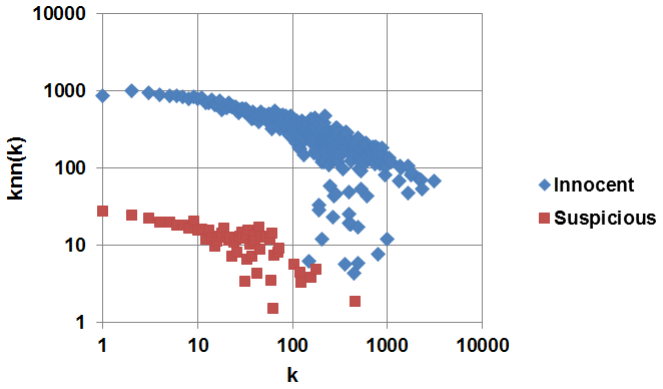

Figure 5.46: Degree correlation for friendship

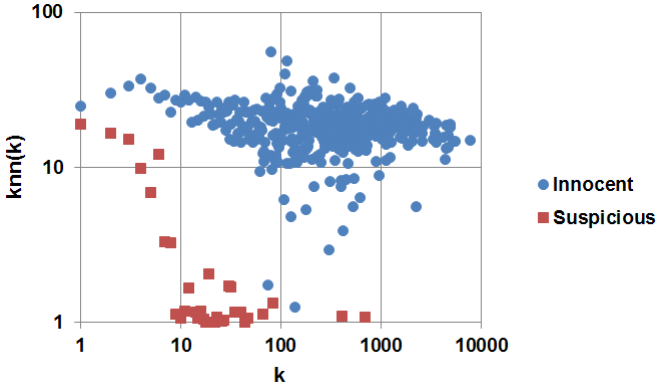

Figure 5.47: Degree correlation for flower relationships

In this section we investigate the manipulative behavior of reviewers in Dianping.com. A subset of reviewers in Dianping is collected using Breadth-first search (BFS). Features regarding clustering are specified in section 5.3.2 and a small number of reviewers are manually annotated. Annotation-Auxiliary Clustering (AAClust) is proposed for identifying suspicious reviewers regarding manipulation. The comparison result between distributions of rating provided by suspicious reviewers and innocent reviewers shows that the distributions look very similar. This similarity tells us that in Dianping suspicious reviewers do not concentrate mainly on the biased ratings but textual comments. In addition, two social network metrics degree distribution $P(k)$ (95) and degree correlation k_{nn} (94) are introduced to analyze the topology of the social network in Dianping.com. The analysis results show that the innocent reviewers have larger degree than the suspicious reviewers and the flower which a popular suspicious reviewer sends to another reviewer is mainly the only flower the reviewer receives.

6

Conclusion and Discussion

6.1 Summary

Given a large amount of user-generated information (ratings, observations, experiences or evidences) for a service, this thesis investigates the idea of trust measures to solve the problem of service selection by mapping the multi-dimension user-generated information into a single value or a vector. 40 trust models (5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38) proposed by other researchers are compared from both theoretical and practical perspectives using criteria such as application context, information representation, properties of trust evaluation, and robustness of system. A trust model framework is proposed that generalizes the core of a trust model. In addition, we propose six metrics for measuring the confidence of trust evaluation at system and query level. A combination of trust value and the corresponding confidence in the trust value helps users to select services more reliable.

We study trust building and management for both quantitative and qualitative services. We choose an Online File Storage Service (OFSS) as a case study for a quantitative service. Trust models for an OFSS consider the attributes of failure rate and bandwidth. Two manipulation detection approaches, Baseline Sampling (BS) and Clique Identification (CI) are proposed to mitigate the negative influence of manipulation on the Trust Management System (TMS) for the OFSS. A simulation platform is designed and simulation results show that both BS and CI suppress the influence of manipulation largely, where CI outperforms BS for each of three types of manipulation, promoting manipulation, slandering manipulation, and mixed promoting and slandering manipulation. For qualitative services, we study three instances, an e-commerce website Taobao.com, a travel-related-service-review website TripAdvisor.com, and a lifestyle-related-service-review website Dianping.com. Considering different characters of the three instances, different criteria such as price of a transaction, frequency of trading, suspicious degree of a reviewer are used to design weighting functions. We propose two universal approaches, Ranking Comparison Index (RCI) and Benefit Variation Ratio (BVR), for comparing the robustness of trust models given a set of suspicious objects

such as vendors and hotels. Trust models are compared using RCI or BVR, and the results show that our proposed trust models outperform the baseline trust model. In addition, for qualitative services, we propose three types of machine learning approach, clustering, classification, and semi-unsupervised learning, for identifying suspicious objects with respect to manipulation. Afterwards, the statistical character of features in the suspicious group, and innocent group, such as the rating distribution, rating helpfulness, service ranking, are compared.

6.2 Remaining Challenges

6.2.1 Formalization of a TMS

The purpose of TMS formalization is to analyze the properties of a TMS such as accuracy. TMS formalization refers to model a TMS as a formal system based on mathematics. The accuracy of a TMS is a critical issue, since a TMS becomes useless when the system delivers inaccurate results. The first step of formalization is to identify the key variables in a TMS such as a rating of a service, the consumer set, the service set, etc. Then a metric or metrics should be defined to measure the quantity of a property. For the property of TMS accuracy, we define mismatch of a TMS at time t as the metric in Section 3.1.3. Next, mathematical statements are proposed and proved based on the definitions above. In Section 3.1.3, we proved two lemmas for accuracy of a TMS given the restriction that all the rating images and labels never change over time. However a statement becomes extremely difficult to prove when we try to relax the restriction. The difficulty lies in how to model manipulation and evaluation of a TMS per se. For manipulation modeling, the key factors are the types of manipulation and the strength of manipulation. For evaluation of a TMS, we shall model how the rating image varies over time and this is largely dependent on the design of a trust model. Given different weight functions, TMSs evaluate trust differently over time. We argue that, the proof on a property of a universal TMS such as accuracy is impossible. However it is possible to prove a property of a concrete TMS where the modeling of manipulation and weight functions are fixed.

6.2.2 The Identity Problem

There is not a correspondence between one person and their identity on the Internet. Users typically create multiple accounts for different websites and currently there is few mechanisms relating the accounts together. Furthermore, users can generate multiple accounts for review websites such as TripAdvisor.com and Dianping.com and theoretically there is no way to group the accounts accurately. The ambiguity of identity on the Internet causes difficulty for identifying manipulation. One dishonest person can create multiple accounts and post positive reviews for the same hotel property. Additionally, in order to promote one service, a group of people could take action on the same target where each person has multiple accounts.

Instead of identifying the correspondence between suspicious accounts and one person, we should focus on the behavioral characteristics of a group of accounts. For

instance, in the same day there are five accounts which post five reviews for the same service from the same IP address. In order to capture the behavioral character of a group of accounts, the detection method should collect the history of behavior for an account and analyze them. For instance, currently Google only allows registered users to post reviews in the Google Play online store[1]. Anonymous users whose behavior history is untraceable are restricted from posting reviews.

Looking into the future, with the development of the Internet, the identity of one person in the physical world could eventually be mapped into the virtual space. Until then, the strength of the manipulation will be largely restricted, since one cannot create multiple virtual identities online arbitrarily. TMSs will be playing a more significant role for service provisioning than currently, since the trust in the real world can be completely imported into the virtual world.

6.2.3 Evaluating a Trust Model

The main approach of evaluating a trust model introduced in the research work (15, 16, 17, 33, 75) is based on a user-service matrix where each entry represents the rating a user gives to a service. The matrix is not completely filled, since a user will not typically use all the services. The user-service matrix is therefore a sparse matrix. For instance, in the case of recommender systems (33, 75), the matrix is instantiated as a user-movie or user-product matrix. For instance, a user-product matrix in Amazon.com is a typical sparse matrix. In order to evaluate a trust model, the values of some entries are selected and hidden. Afterwards, the prediction of these values is made by applying the trust model on the remaining entries. The goal is to check whether the trust model can make an accurate prediction on the selected entries where the data is hidden. The main weakness of using a user-service matrix for trust model evaluation is that time is not taken into account during the process of trust model evaluation. Evaluation based on a user-service matrix might work if the service quality does not vary over time for example, in the case of movies and products. However, due to the dynamics of service quality, evaluation based on a user-service matrix might not perform well. For instance, for trust evaluation of a hotel (service), it is illogical to predict an entry that is one year old by considering the entries generated one week ago. Therefore we should select entries very carefully to avoid logical inconsistencies.

A good way of evaluating a trust model is to compare the prediction and the rating given by the evaluator afterwards. There is a strong correspondence between the trust model applied, the ratings used for evaluating trust and the rating given by the evaluator afterwards. Since the decision of choosing one service largely depends on the trust evaluation result provided by the trust model and the result is generated from the available ratings at that moment. After the evaluator (user) used the service, they generate a rating for the service. This feedback-based evaluation approach is a practical one and it is not labor-saving to apply this approach for research.

[1] http://www.spiegel.de/netzwelt/web/app-bewertungen-google-vermaehlt-plus-und-play-a-871065.html

6.2.4 Application of Trust Models in Industry

Once a new trust model has been developed, how can the model be integrated into the current TMS? This is one of the key problems which hinder the upgrade of a TMS running in industry. For instance, previous work (4, 96, 97) has mentioned a defect in the trust model used by eBay. The trust value is calculated by summing up all the ratings. A positive rating is equal to +1; a negative rating is equal to -1; a neutral rating is equal to zero. We all see the defect that eBay simply treats all the ratings equally. The current problem is that how can we improve the original trust model used by eBay in practice?

The trust model framework introduced in Chapter 3 can solve the problem. The basic idea of the framework is to give different ratings different weights and calculate the average. The basic operation of a trust model is to compute the average like the trust model used by eBay does. We can improve the trust model by giving different weights to different observations. Therefore all trust models which implement the trust model framework can easily replace the original trust model employed by eBay. The only work one should do is to design new weight functions.

6.2.5 Low Incentive to Provide a Rating

Five years ago, Audun Jøsangs survey (4) pointed out that users lack motivation to provide ratings. In the paper, the authors mentioned that Epinions[1] and BizRate[2] provided financial incentives such as discounts or cash incentives. In order to solve the data sparsity, TripAdvisor.com asks user to rate the hotels using a pop-up web page when someone opens its homepage. TripAdvisor.com's approach is not a good way to solve the problem since people usually do not like to be disturbed in this way. A combination of social network and rating provisioning could be one way to solve the problem. The motivation of writing or providing a review a rating shows friends where they have travelled and how they feel about services such as Google+[3]. A social-network-based TMS can extract trust evaluation related information using Opinion Mining, Text Mining, Web Ontology Language, etc. The basic idea is to explicitly or implicitly obtain the evaluation of a service from an end user and build a TMS afterwards.

6.2.6 Purposeless Attack identification

Purposeless attack refers to an attack the intention of which is not explicit or hidden. For instance, one might post a review of a hotel with random words to the system. Or in order to destroy one TMS, one could inject random ratings to the TMS without any explicit pattern. The real motivation behind the destructive behavior could be that, there are two TMSs and one tries to destroy the other one. This type of

[1] www.epinions.com

[2] www.bizrate.com

[3] https://plus.google.com

attack is extremely difficult to identify due to the unclear intention behind it. To identify purposeless attack, the semi-unsupervised learning approach, Annotation-Auxiliary Clustering (AAClust), should outperform the clustering and classification approaches. The reason is, on the one hand, we cannot easily draw some convincing assumption like in the case studies on Taobao.com and TripAdvisor.com. On the other hand, it is extremely time-consuming to annotate suspicious objects like reviews or reviewers. AAclust fits well in the machine learning problem with few assumptions and a small number of annotations.

6.3 Possible Applications of TMSs for Trust in Service-s

6.3.1 Integration of TMSs

With the increasing of the number of TMSs, the integration of TMSs becomes an issue for facilitating peoples service selection. For one service, you can find its entry on more than one TMS. For instance, for a hotel in Berlin, you can find the hotel on the TMSs of HRS.de, TripAdvisor.com, Booking.com, Holidaycheck.com, etc.; for a restaurant in Heidelberg, you can find this restaurant on the TMSs of TripAdvisor.com, restaurant-kritik.de, qype.com, etc. It is convenient for users to look up services from one ultimate TMS instead of several TMSs.

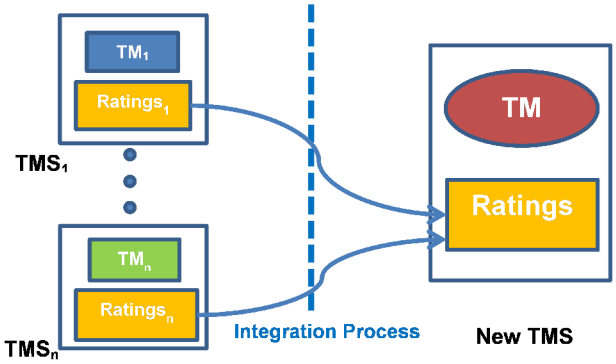

Figure 6.1: Rating-level integration

Here we discuss theoretical and technical issues surrounding TMS integration. The practical issues such as negotiation with the TMSs and data extraction process are outside the scope of this thesis. There are two types of integration, rating-level integration and model-level integration. Rating-level integration refers to the integration where only the inputs of TMSs, ratings, are collected. A new trust model is used to evaluate trust in the new TMS which is the product of the integration. Fig. 6.1 shows a diagram of rating-level integration. The task of integration is to derive a new TMS from n TMSs. Each old TMS is modeled as a collection of ratings and a trust model TMi. For the new TMS, a new trust model TM is created since none of the trust

models in the n TMSs is able to evaluate the ratings which are generated by combining all old ratings in the n TMSs. For instance, there are two TMSs. One mainly contains ratings about the hotels in Germany, while the other one mainly contains ratings about the hotels in China. Each original trust model might be specific to the corresponding regional features such as culture differences. For the integration of the two TMSs, a new trust model is constructed by compromising on the original trust models.

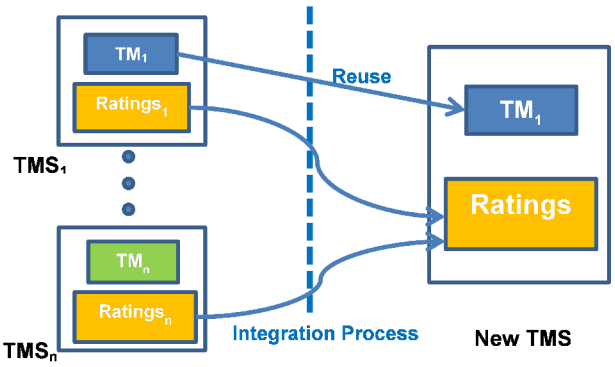

Figure 6.2: Model-level integration

Model-level integration refers to an integration that the ratings in the old TMSs are combined and a trust model is reused. Instead of creating a new trust model, an original trust model is reused in the new TMS. The model-level integration is demonstrated in Fig. 6.2.

6.3.2 Information Service Quality Prediction

With the prevalence of social networking and microblogging services, sources of information such as news, advertisements and reports are extended to normal people. Nowadays not only the mainstream media but also individual citizens can generate and propagate information online. Meanwhile, the speed of information propagating via social networks on the Internet becomes extremely fast. During the process of information generation and propagation, information service quality becomes one uncertain factor which should be considered. Information service quality contains two types of quality, information generation quality and information propagation quality. Considering the two types of quality as services, Trust can be attached to the two types of service.

Fig. 6.3 illustrates the trust evaluation for an information service. The information is generated from a source, propagated through a social network and received by an evaluator. The trust in the information service is calculated by combining both the trust of the source and the trust of the nodes through which the information is propagated, namely the trust in information generation and information propagation.

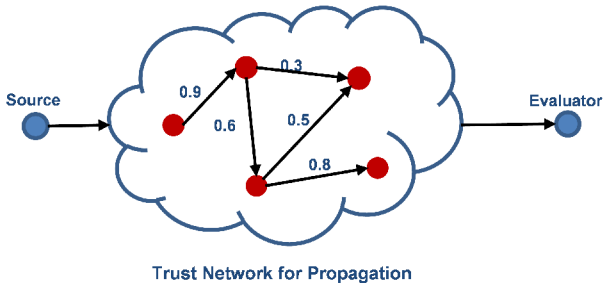

Figure 6.3: Trust evaluation for information service

Attribute	Observation	TM	Criteria for Weight Function Design
Prediction Accuracy	Binary	Beta System (8)	Time, Transitivity
Response Time	Discrete	Dirichlet System (36)	Time, Rating preference

Table 6.1: TMS service specification

6.3.3 A TMS of TMSs

Considering a TMS as a special service, one can imagine that there could be a TMS for evaluating the quality of a TMS service. Following the same idea of trust evaluation in services, we could specify a number of attributes for a TMS service. Table 6.1 shows the attributes considered for trust evaluation on a TMS service.

At least, two attributes, prediction accuracy, and response time, should be considered. Prediction accuracy refers to the observation of whether a TMS makes a good prediction of the service quality. For instance, one user chooses one service having the highest trust value in a TMS. After using the service, a user is satisfied with it and the observation is positive. If the user is not satisfied with it, then the observation should be negative. The attribute of response is equivalent to the concept for system performance. As a special Information System (IS), a TMS should respond to the query in a short time. One could empirically divide the response time into several intervals like what we have done for the attribute of an OFSS bandwidth. Therefore the observation of response time is a discrete variable. For instance, we can divide the domain of response time into three intervals, which correspond to the set slow, acceptable, fast.

In addition, different criteria are considered for each attribute in order to design weight functions. Time and transitivity of trust are considered for predicting accuracy. Time is chosen as one of the criteria since as a service, the quality of a TMS service might change over time. Transitivity of trust is considered as well since a peer can provide and consume a TMS service in a P2P environment. It is possible to calculate the trustworthiness of a peer via applying transitive closure operation on the trust network derived from the target system. Time and Rating preference are considered for response time. As we have analyzed in Chapter 4, the perception of response time

from the user end might be different from region to region. We should give more weight to the referrals who are similar to the evaluator with respect to the network contexts such as IP prefix and geographical location.

References

[1] D. Harrison Mcknight and Norman L. Chervany. **The Meanings of Trust**. Technical report, 1996. ix, 2, 7, 8, 15, 16, 18

[2] Stephen Paul Marsh. *Formalising trust as a computational concept*. PhD thesis, University of Stirling, 1994. ix, 2, 7, 9, 11, 15, 28

[3] Myle Ott, Claire Cardie, and Jeff Hancock. **Estimating the Prevalence of Deception in Online Review Communities**. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 201–210, New York, NY, USA, 2012. ACM. xiv, 86, 99

[4] Audun Jøsang, Roslan Ismail, and Colin Boyd. **A Survey of Trust and Reputation Systems for Online Service Provision**. *Decis. Support Syst.*, **43**(2):618–644, March 2007. 2, 3, 7, 8, 10, 15, 16, 28, 120

[5] Matthew Richardson, Rakesh Agrawal, and Pedro Domingos. **Trust Management for the Semantic Web**. In *IN PROCEEDINGS OF THE SECOND INTERNATIONAL SEMANTIC WEB CONFERENCE*, pages 351–368, 2003. 3, 4, 9, 11, 13, 14, 17, 20, 21, 25, 28, 29, 30, 117

[6] Bin Yu and Munindar P. Singh. **A Social Mechanism of Reputation Management in Electronic Communities**. In *In Proceedings of Fourth International Workshop on Cooperative Information Agents*, pages 154–165, 2000. 3, 4, 9, 11, 13, 14, 17, 18, 20, 21, 25, 117

[7] D.W. Manchala. **E-commerce trust metrics and models**. *Internet Computing, IEEE*, **4**(2):36–44, Mar 2000. 3, 4, 9, 11, 13, 14, 15, 17, 18, 20, 21, 25, 117

[8] Audun Jøsang. **A Logic for Uncertain Probabilities**. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, **9**(3):279–311, June 2001. 3, 4, 9, 10, 11, 13, 14, 17, 20, 21, 25, 28, 29, 117, 123

[9] Bin Yu and Munindar P. Singh. **Detecting Deception in Reputation Management**. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 73–80, New York, NY, USA, 2003. ACM. 3, 4, 9, 11, 13, 14, 15, 17, 18, 20, 21, 25, 117

[10] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. **The Eigentrust Algorithm for Reputation Management in P2P Networks**. In *Proceedings of the 12th International Conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM. 3, 4, 9, 11, 12, 13, 14, 15, 17, 18, 20, 21, 25, 28, 30, 117

[11] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. **Propagation of Trust and Distrust**. In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pages 403–412, New York, NY, USA, 2004. ACM. 3, 4, 9, 11, 13, 14, 17, 20, 21, 25, 117

[12] Yonghong Wang and Munindar P. Singh. **Trust Representation and Aggregation in a Distributed Agent System.** In *AAAI*, pages 1425–1430. AAAI Press, 2006. 3, 4, 9, 11, 13, 14, 17, 20, 21, 25, 117

[13] Yonghong Wang and Munindar P. Singh. **Formal Trust Model for Multiagent Systems**. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1551–1556, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc. 3, 4, 9, 11, 13, 14, 17, 20, 21, 25, 31, 117

[14] Runfang Zhou and Kai Hwang. **PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing**. *Parallel and Distributed Systems, IEEE Transactions on*, **18**(4):460–473, April 2007. 3, 4, 9, 11, 13, 14, 15, 17, 18, 20, 21, 25, 28, 117

[15] D. Quercia, S. Hailes, and L. Capra. **Lightweight Distributed Trust Propagation**. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 282–291, Oct 2007. 3, 4, 9, 10, 11, 13, 14, 15, 17, 18, 20, 21, 25, 117, 119

[16] Haifeng Liu, Ee-Peng Lim, Hady W. Lauw, Minh-Tam Le, Aixin Sun, Jaideep Srivastava, and Young Ae Kim. **Predicting Trusts Among Users of Online Communities: An Epinions Case Study**. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, EC '08, pages 310–319, New York, NY, USA, 2008. ACM. 3, 4, 9, 11, 13, 14, 15, 17, 20, 21, 25, 117, 119

[17] Audun Jøsang and Simon Pope. **Semantic Constraints for Trust Transitivity**. In *Proceedings of the 2Nd Asia-Pacific Conference on Conceptual Modelling - Volume 43*, APCCM '05, pages 59–68, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc. 3, 4, 9, 11, 15, 25, 28, 117, 119

[18] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. **Item-based Collaborative Filtering Recommendation Algorithms**. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM. 3, 4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 25, 28, 117

[19] Jordi Sabater and Carles Sierra. **REGRET: Reputation in Gregarious Societies**. In *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, pages 194–195, New York, NY, USA, 2001. ACM. 3, 4, 10, 11, 12, 13, 14, 15, 17, 18, 20, 21, 25, 30, 31, 32, 117

[20] Bin Yu and Munindar P. Singh. *Computational Intelligence*, **18**(4):535–549, 2002. 3, 4, 10, 11, 13, 14, 17, 18, 20, 21, 25, 117

[21] Sonja Buchegger and Jean Y. Le Boudec. **A Robust Reputation System for P2P and Mobile Ad-hoc Networks**. In *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems*, 2004. 3, 4, 10, 11, 13, 14, 15, 17, 18, 19, 20, 21, 25, 27, 30, 117

REFERENCES

[22] Li Xiong and Ling Liu. **PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities**. *IEEE Transactions on Knowledge and Data Engineering*, **16**(7):843–857, 2004. 3, 4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 25, 30, 117

[23] Mudhakar Srivatsa, Li Xiong, and Ling Liu. **TrustGuard: Countering Vulnerabilities in Reputation Management for Decentralized Overlay Networks**. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 422–431, New York, NY, USA, 2005. ACM. 3, 4, 10, 11, 13, 14, 15, 17, 18, 19, 20, 21, 25, 30, 117

[24] W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. **TRAVOS: Trust and reputation in the context of inaccurate information sources**. *Journal of Autonomous Agents and Multi-Agent Systems*, **12**(2):183–198, 2006. 3, 4, 10, 11, 13, 14, 15, 17, 19, 20, 21, 25, 30, 117

[25] Jie Zhang and Robin Cohen. **Evaluating the Trustworthiness of Advice About Seller Agents in e-Marketplaces: A Personalized Approach**. *Electron. Commer. Rec. Appl.*, **7**(3):330–340, November 2008. 3, 4, 10, 11, 13, 14, 15, 17, 18, 20, 21, 25, 30, 117

[26] Zeinab Noorian, Stephen Marsh, and Michael Fleming. **Multi-layer Cognitive Filtering by Behavioral Modeling**. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 871–878, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems. 3, 4, 10, 11, 13, 14, 15, 17, 19, 20, 21, 25, 30, 117

[27] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. **An Algorithmic Framework for Performing Collaborative Filtering**. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA, 1999. ACM. 3, 4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 25, 26, 30, 117

[28] Shanshan Song, Kai Hwang, Runfang Zhou, and Yu-Kwong Kwok. **Trusted P2P Transactions with Fuzzy Reputation Aggregation**. *IEEE Internet Computing*, **9**(6):24–34, November 2005. 3, 4, 10, 11, 13, 14, 15, 17, 18, 20, 21, 25, 117

[29] E. Michael Maximilien and Munindar P. Singh. **Agent-based Trust Model Involving Multiple Qualities**. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 519–526, New York, NY, USA, 2005. ACM. 3, 4, 11, 12, 13, 14, 15, 17, 20, 21, 25, 117

[30] Yao Wang and J. Vassileva. **Bayesian network-based trust model**. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 372–378, Oct 2003. 3, 4, 10, 11, 12, 13, 14, 17, 20, 21, 25, 117

[31] Mao Chen and Jaswinder Pal Singh. **Computing and Using Reputations for Internet Ratings**. In *Proceedings of the 3rd ACM Conference on Electronic Commerce*, EC '01, pages 154–162, New York, NY, USA, 2001. ACM. 3, 4, 10, 11, 12, 13, 14, 15, 17, 20, 21, 25, 30, 31, 32, 117

[32] Paul Resnick and Rahul Sami. **The Influence Limiter: Provably Manipulation-resistant Recommender Systems**. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pages 25–32, New York, NY, USA, 2007. ACM. 3, 4, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 25, 117

[33] Marko Balabanović and Yoav Shoham. **Fab: Content-based, Collaborative Recommendation**. *Commun. ACM*, **40**(3):66–72, March 1997. 3, 4, 11, 13, 14, 17, 18, 20, 21, 25, 117, 119

[34] Karl Aberer and Zoran Despotovic. **Managing Trust in a Peer-2-peer Information System**. In *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, pages 310–317, New York, NY, USA, 2001. ACM. 3, 4, 11, 13, 14, 17, 19, 20, 21, 25, 117

[35] Bin Yu and Munindar P. Singh. **Detecting Deception in Reputation Management**. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 73–80, New York, NY, USA, 2003. ACM. 3, 4, 11, 25, 117

[36] Yanchao Zhang and Yuguang Fang. **A Fine-Grained Reputation System for Reliable Service Selection in Peer-to-Peer Networks**. *IEEE Transactions on Parallel and Distributed Systems*, **18**(8):1134–1145, 2007. 3, 4, 11, 13, 14, 15, 16, 17, 18, 20, 21, 25, 30, 45, 117, 123

[37] N. Limam and R. Boutaba. **Assessing Software Service Quality and Trustworthiness at Selection Time**. *Software Engineering, IEEE Transactions on*, **36**(4):559–574, July 2010. 3, 4, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 25, 117

[38] Alfarez Abdul-Rahman and Stephen Hailes. **Supporting Trust in Virtual Communities**. In *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6 - Volume 6*, HICSS '00, pages 6007–, Washington, DC, USA, 2000. IEEE Computer Society. 3, 4, 11, 13, 14, 17, 20, 21, 25, 117

[39] M. Boldt, A Borg, and B. Carlsson. **On the Simulation of a Software Reputation System**. In *Availability, Reliability, and Security, 2010. ARES '10 International Conference on*, pages 333–340, Feb 2010. 3, 4

[40] Yao Wang, Jie Zhang, and Julita Vassileva. **Effective Web Service Selection via Communities Formed by Super-Agents**. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, **1**:549–556, 2010. 3, 4

[41] Jianming He. *A Social Network-based Recommender System*. PhD thesis, Los Angeles, CA, USA, 2010. AAI3437557. 3, 4

[42] Le-Hung Vu, Manfred Hauswirth, and Karl Aberer. **QoS-Based Service Selection and Ranking with Trust and Reputation Management**. In *Proceedings of the 2005 Confederated International Conference on On the Move to Meaningful Internet Systems - Volume Part I*, OTM'05, pages 466–483, Berlin, Heidelberg, 2005. Springer-Verlag. 3, 4

[43] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. **Collaborative reputation mechanisms for electronic marketplaces**. *Decision Support Systems*, **29**(4):371 – 388, 2000. 3, 4

[44] Jason Sonnek, Abhishek Chandra, and Jon Weissman. **Adaptive Reputation-Based Scheduling on Unreliable Distributed Infrastructures**. *IEEE Transactions on Parallel and Distributed Systems*, **18**(11):1551–1564, 2007. 3, 4

[45] Elena Zheleva, Aleksander Kolcz, and Lise Getoor. **Trusting Spam Reporters: A Reporter-based Reputation System for Email Filtering**. *ACM Trans. Inf. Syst.*, **27**(1):3:1–3:27, December 2008. 3, 4

[46] John O'Donovan and Barry Smyth. **Trust in Recommender Systems**. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, pages 167–174, New York, NY, USA, 2005. ACM. 3, 4

[47] W. T. Luke Teacy, Jigar Patel, Nicholas R. Jennings, and Michael Luck. **Coping with Inaccurate Reputation Sources: Experimental Analysis of a Probabilistic Trust Model**. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '05, pages 997–1004, New York, NY, USA, 2005. ACM. 3, 4

[48] Zeinab Noorian and Mihaela Ulieru. **The State of the Art in Trust and Reputation Systems: A Framework for Comparison**. *J. Theor. Appl. Electron. Commer. Res.*, **5**(2):97–117, August 2010. 3, 4

[49] Trung Dong Huynh, Nicholas R. Jennings, and Nigel R. Shadbolt. **An Integrated Trust and Reputation Model for Open Multi-agent Systems**. *Autonomous Agents and Multi-Agent Systems*, **13**(2):119–154, September 2006. 3, 4

[50] Tyrone Grandison and Morris Sloman. **A Survey of Trust in Internet Applications**. *Commun. Surveys Tuts.*, **3**(4):2–16, October 2000. 3, 4

[51] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. **A Survey of Attack and Defense Techniques for Reputation Systems**. *ACM Comput. Surv.*, **42**(1):1:1–1:31, December 2009. 3, 10, 16, 48

[52] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. **Toward Trustworthy Recommender Systems: An Analysis of Attack Models and Algorithm Robustness**. *ACM Trans. Internet Technol.*, **7**(4), October 2007. 3, 16, 48

[53] R. Golembiewski and M. McConkie. **The Centrality of Interpersonal Trust in Group Processes**. *Theories of Group Processes*, pages 131–185, 1975. 8

[54] M. Deutsch. **The resolution of conict: Constructive and destructive processes.** *New Haven: Yale University Press.*, 1973. 8

[55] Nebraska Symposium on Motivation. *Cooperation and Trust: Some Theoretical Notes.* Nebraska University Press, 1962. 8

[56] Niklas Luhmann. *Trust and Power.* John Wiley and Sons Ltd., 1979. 8

[57] A. H. Harcourt. **Help, cooperation and trust in animals**. *Cooperation and Prosocial Behaviour*, page 1526, 1991. 8

[58] Diego Gambetta. **Can We Trust Trust?** In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988. 8

[59] Huiying Duan and Feifei Liu. **Building and Managing Reputation in the Environment of Chinese e-Commerce: A Case Study on Taobao**. In *Proceedings of the 2Nd International Conference on Web Intelligence, Mining and Semantics*, WIMS '12, pages 43:1–43:10, New York, NY, USA, 2012. ACM. 13, 14, 15, 17, 19, 22, 23, 27, 64

[60] Reid Kerr and Robin Cohen. **Smart Cheaters Do Prosper: Defeating Trust and Reputation Systems**. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 993–1000, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems. 19

[61] Audun Jsang. **Robustness of Trust and Reputation Systems: Does It Matter?** In Theo Dimitrakos, Rajat Moona, Dhiren Patel, and D.Harrison McKnight, editors, *Trust Management VI*, **374** of *IFIP Advances in Information and Communication Technology*, pages 253–262. Springer Berlin Heidelberg, 2012. 19

[62] Nitin Jindal and Bing Liu. **Opinion Spam and Analysis**. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, WSDM '08, pages 219–230, New York, NY, USA, 2008. ACM. 19, 22, 27, 98

[63] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. **Detecting Product Review Spammers Using Rating Behaviors**. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 939–948, New York, NY, USA, 2010. ACM. 19, 22, 27

[64] Arjun Mukherjee, Bing Liu, and Natalie Glance. **Spotting Fake Reviewer Groups in Consumer Reviews**. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 191–200, New York, NY, USA, 2012. ACM. 19, 22, 27

[65] Huiying Duan and Feifei Liu. **Building Robust Reputation Systems in the E-commerce Environment**. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, pages 326–333, June 2012. 19, 22, 27

[66] Guangyu Wu, Derek Greene, and Pádraig Cunningham. **Merging Multiple Criteria to Identify Suspicious Reviews**. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pages 241–244, New York, NY, USA, 2010. ACM. 19, 22, 27, 81, 82, 84, 85

[67] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. **Finding Deceptive Opinion Spam by Any Stretch of the Imagination**. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 309–319, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. 19, 22, 27, 86, 96, 97, 98

REFERENCES

[68] Huiying Duan and Peng Yang. **Building robust Reputation Systems for travel-related services**. In *Privacy, Security and Trust (PST), 2012 Tenth Annual International Conference on*, pages 168–175, July 2012. 13, 14, 17, 19, 22, 23, 27, 86, 97

[69] Huiying Duan and Cäcilia Zirn. **Can We Identify Manipulative Behavior and the Corresponding Suspects on Review Websites Using Supervised Learning?** In *Proceedings of the 17th Nordic Conference on Secure IT Systems*, NordSec'12, pages 215–230, Berlin, Heidelberg, 2012. Springer-Verlag. 19, 22, 27

[70] Chrysanthos Dellarocas. **Immunizing Online Reputation Reporting Systems Against Unfair Ratings and Discriminatory Behavior**. In *Proceedings of the 2Nd ACM Conference on Electronic Commerce*, EC '00, pages 150–157, New York, NY, USA, 2000. ACM. 19, 22, 26, 27

[71] Andrew Whitby, Audun Jsang, and Jadwiga Indulska. **Filtering Out Unfair Ratings in Bayesian Reputation Systems**. In *The Icfain Journal of Management Research*, 2005. 19, 22, 27

[72] Huiying Duan. **Trust Building and Management for Online File Storage Service**. In Changhoon Lee, Jean-Marc Seigneur, JamesJ. Park, and RolandR. Wagner, editors, *Secure and Trust Computing, Data Management, and Applications*, **187** of *Communications in Computer and Information Science*, pages 31–40. Springer Berlin Heidelberg, 2011. 19, 22, 27

[73] Xiaojin Zhu. **Semi-Supervised Learning Literature Survey**, 2006. 22

[74] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. 22, 109

[75] Michael P. O'Mahony and Barry Smyth. **A Classification-based Review Recommender**. In Max Bramer, Richard Ellis, and Miltos Petridis, editors, *Research and Development in Intelligent Systems XXVI*, chapter 4, pages 49–62. Springer London, London, 2010. 22, 27, 89, 100, 119

[76] Huiying Duan. **Trust Building and Management for Rated-Online Services**. In *Workshop Proc. Of the Fifth IFIP WG 11.11 International Conference on Trust Management (IFIPTM 2011)*, Communications in Computer and Information Science. 2011. 27

[77] L. Mui, M. Mohtashemi, and A. Halberstadt. **A Computational Model of Trust and Reputation for E-businesses**. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 7 - Volume 7*, HICSS '02, pages 188–, Washington, DC, USA, 2002. IEEE Computer Society. 31

[78] Cronin JR J.J Brady, M.K. **Some new thoughts on conceptualizing perceived service quality: A hierarchical approach**. *Journal of Marketing*, **63**(3):34–49, 2001. 36

[79] R.L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Trade-Offs*. Wiley series in probability and mathematical statistics. Applied probability and statistics. Cambridge University Press, 1993. 36, 47

[80] S. Lamparter, D. Oberle, and A Eberhart. **Approximating service utility from policies and value function patterns**. In *Policies for Distributed Systems and Networks, 2005. Sixth IEEE International Workshop on*, pages 159–168, June 2005. 36, 47

[81] Eugene W. Anderson and Mary W. Sullivan. **The Antecedents and Consequences of Customer Satisfaction for Firms**. *Marketing Science*, **12**(2):125–143, 1993. 37

[82] Audun Jøsang and Roslan Ismail. **The Beta Reputation System**. In *In Proceedings of the 15th Bled Electronic Commerce Conference*, 2002. 41, 63, 72

[83] R. W. Sinnott. **Virtues of the Haversine**. *Sky and Telescope*, **68**(2):159+, 1984. 44

[84] Li Chen and Pearl Pu. **Survey of Preference Elicitation Methods**, 2004. 47

[85] Mary Natrella. *NIST/SEMATECH e-Handbook of Statistical Methods*. NIST/SEMATECH, 2010. 49

[86] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley and Sons, 2003. 49, 52

[87] J. Scott and P.J. Carrington. *The SAGE Handbook of Social Network Analysis*. SAGE Publications, 2011. 54

[88] Thomas M. J. Fruchterman and Edward M. Reingold. **Graph Drawing by Force-directed Placement**. *Softw. Pract. Exper.*, **21**(11):1129–1164, November 1991. 54

[89] Brian Neil Levine, Clay Shields, and N. Boris Margolin. **A Survey of Solutions to the Sybil Attack**. (2006-052), 10/2006 2006. 79

[90] Raymond Y. K. Lau, S. Y. Liao, Ron Chi-Wai Kwok, Kaiquan Xu, Yunqing Xia, and Yuefeng Li. **Text Mining and Probabilistic Language Modeling for Online Review Spam Detection**. *ACM Trans. Manage. Inf. Syst.*, **2**(4):25:1–25:30, January 2012. 86

[91] J. Bezdek, R. Ehrlich, and W. Full. **FCM: The fuzzy c-means clustering algorithm**. *Computers & Geosciences*, **10**(2-3):191–203, 1984. 86

[92] Guangyu Wu, Derek Greene, Barry Smyth, and Pádraig Cunningham. **Distortion As a Validation Criterion in the Identification of Suspicious Reviews**. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 10–13, New York, NY, USA, 2010. ACM. 93

[93] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. **Gene Selection for Cancer Classification using Support Vector Machines**. *Machine Learning*, **46**(1-3):389–422, 2002. 99

[94] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. **Measurement and Analysis of Online Social Networks**. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 29–42, New York, NY, USA, 2007. ACM. 105, 109, 111, 115

[95] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. **Analysis of Topological Characteristics of Huge Online Social Networking Services**. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 835–844, New York, NY, USA, 2007. ACM. 109, 111, 115

[96] Paul Resnick, Richard Zeckhauser, John Swanson, and Kate Lockwood. **The value of reputation on eBay: A controlled experiment**. *Experimental Economics*, **9**(2):79–101, 2006. 120

[97] Paul Resnick and Richard Zeckhauser. *The Economics of the Internet and E-Commerce*, chapter Trust among strangers in internet transactions: Empirical analysis of eBay's reputation system. Number 11 in Advances in Applied Microeconomics. Elsevier Science, 2002. 120

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified; notions taken over directly or indirectly from other sources have been identified as such. This thesis has not previously been presented in identical or similar form to any other German or foreign examination board.

The thesis work was conducted from 2008.3.1 to 2013.4.31 under the supervision of Prof. Dr. -Ing. Dr. h.c. Andreas Reuter at University of Heidelberg.

Karlsruhe,